



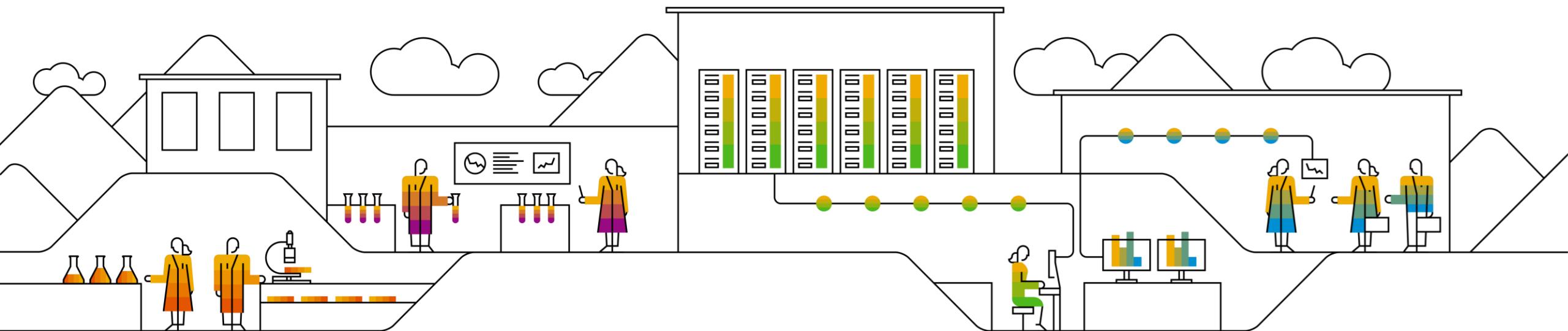
SAP Logistics Business Network, Global Track and Trace Option SAP ERP Integration

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.

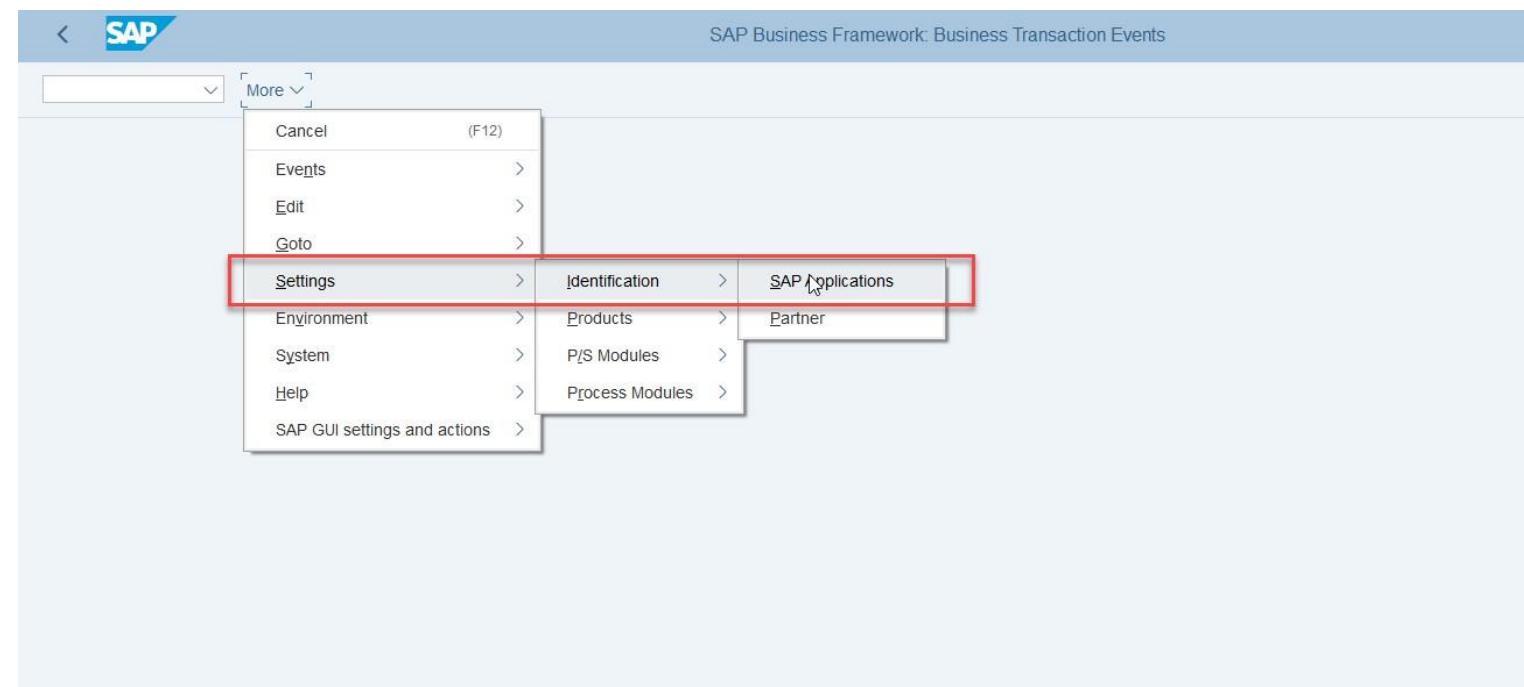
TIPs:

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
QBEXT	<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

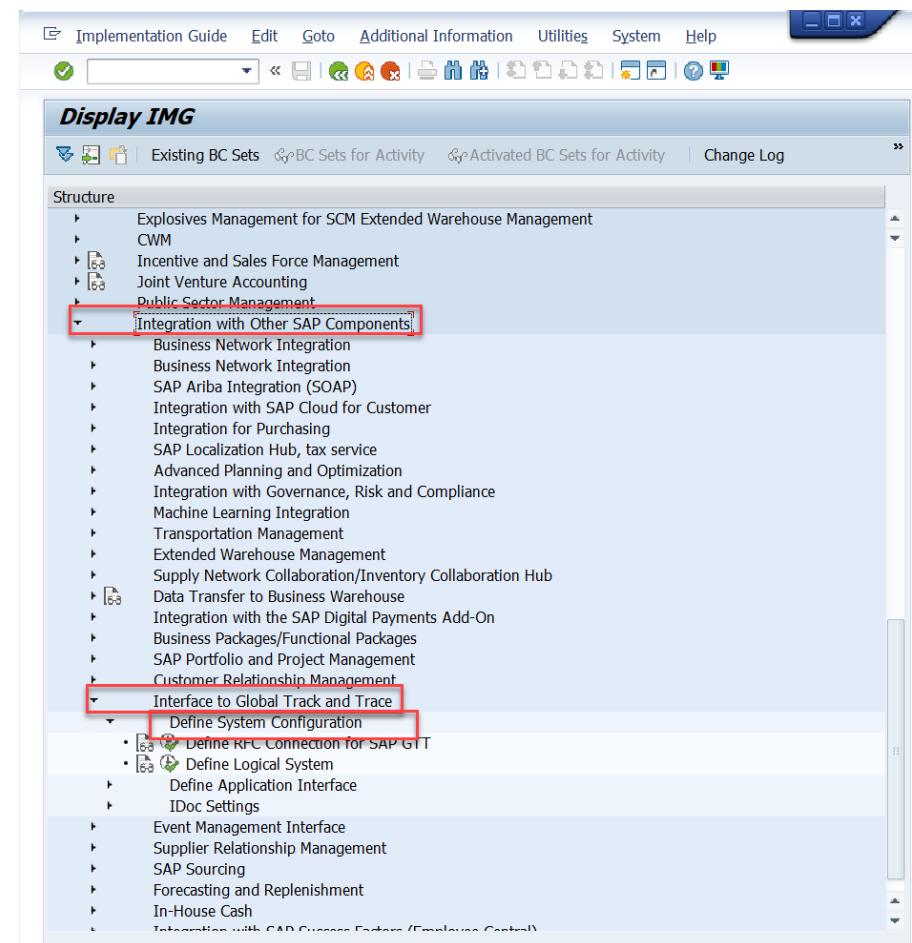
- Basic

B1. IDOC Configuration



STEP 1: Define RFC Connection for GTT

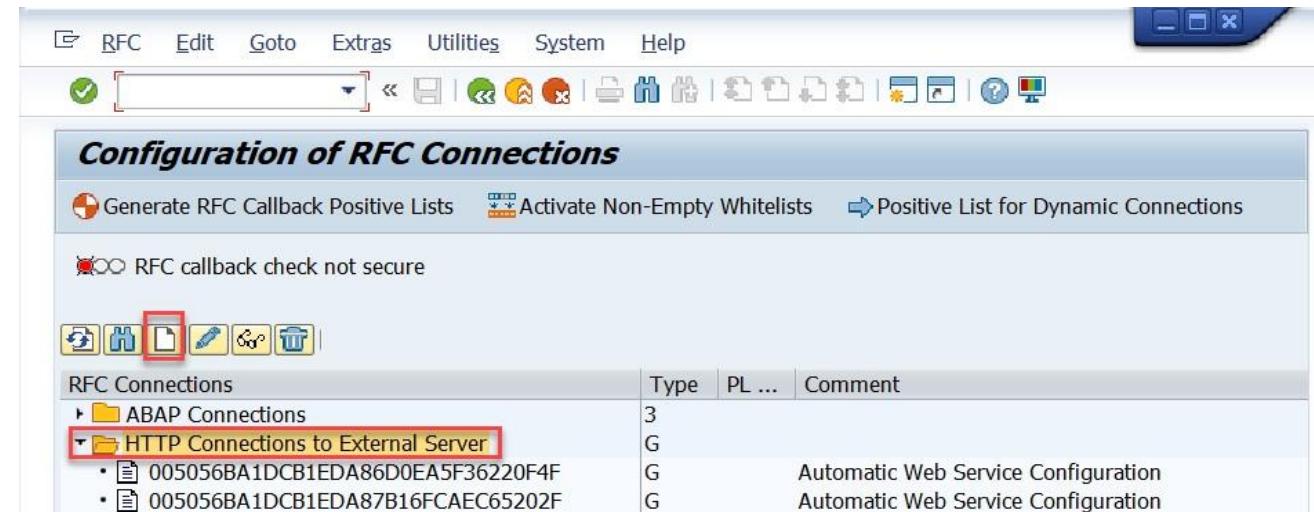
1. Log on to the business client
2. Enter T-code **SPRO** and then click **SAP Reference IMG** to open **Display IMG** page
3. Click **Integration with Other SAP Components**
-> **Interface to Global Track and Trace**
-> **Define System Configuration**
4. Choose activity:
Define RFC Connection for SAP 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, it displays the RFC Destination **ZGTT_SST_FO_EVENT_ACC** and the Connection Type **HTTP Connection to External Server**. Below this, the **Technical Settings** tab is selected. Under the **Target System Settings**, the host is set to **[redacted]** and the port is **443**. The path prefix is set to **/api/idoc/em/v1/Event**. In the **HTTP Proxy Options** section, the **Global Configuration** tab is active, showing fields for **Proxy Host**, **Proxy Service**, **Proxy User**, and **Proxy PW Status** (**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*.

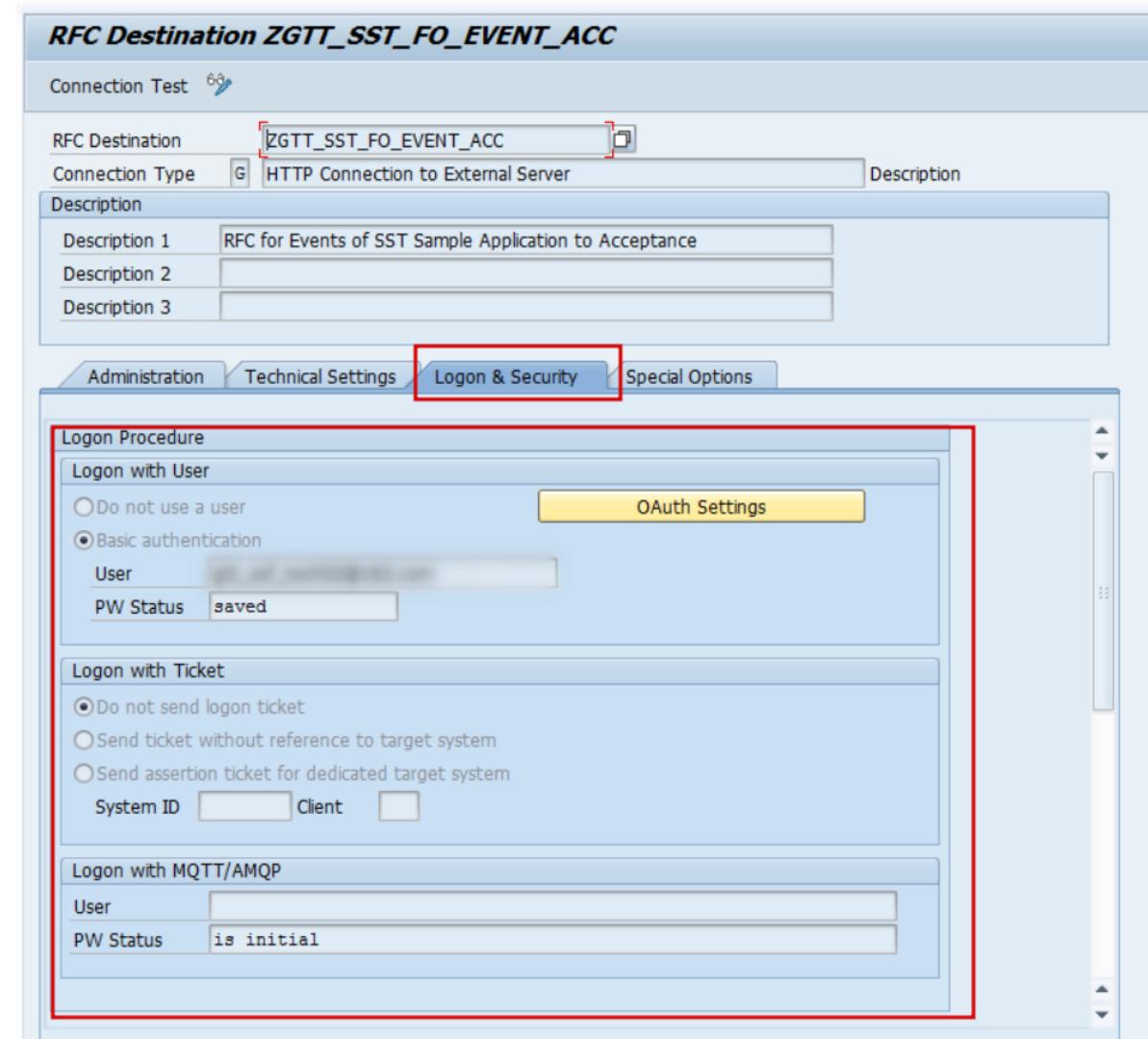
The recommended SSL Certificate is: *DFAULT SSL Client (Standard)*.

10. Save the configuration

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

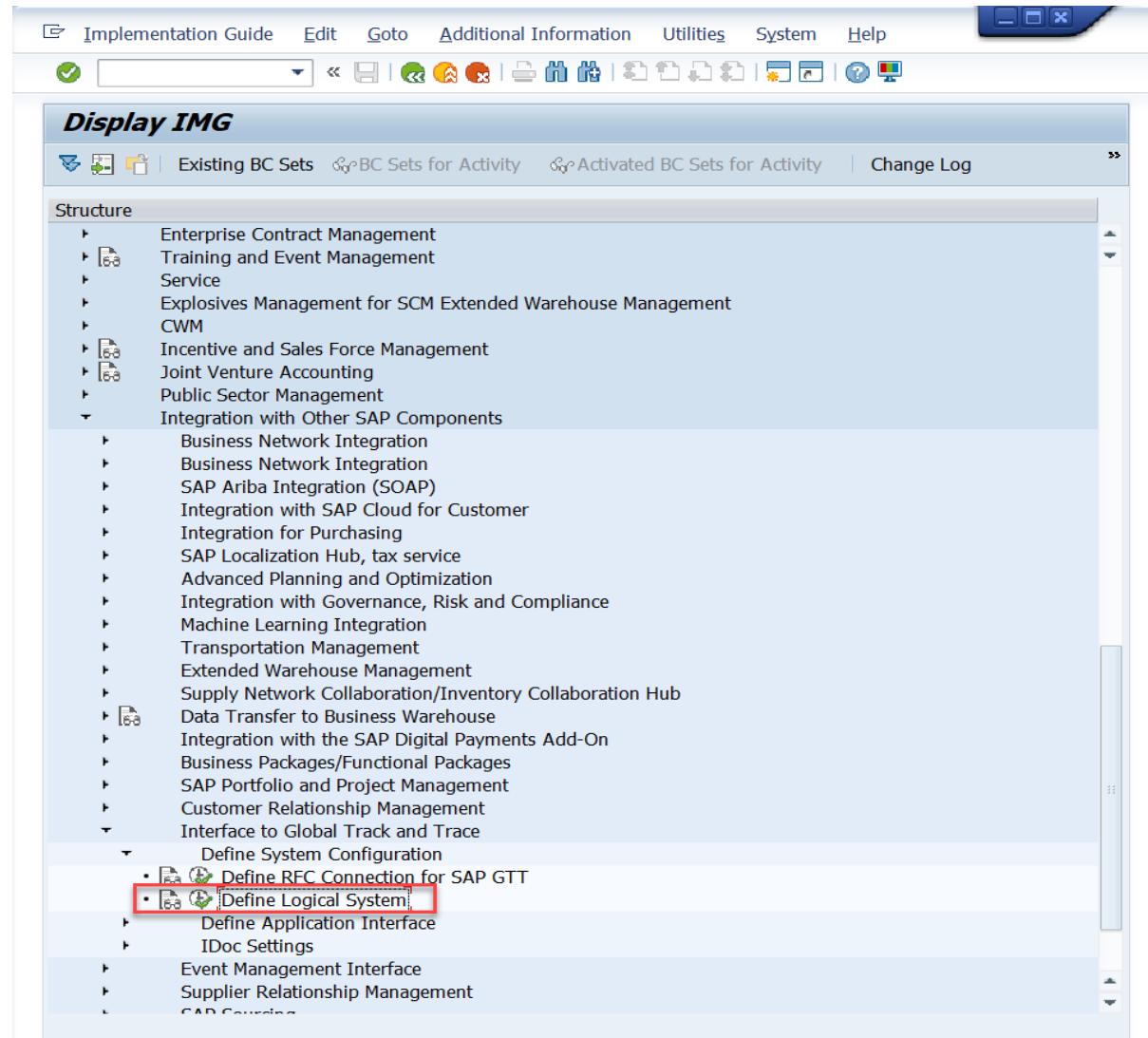
Caution: You need to configure two RFC Connections:

- 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:

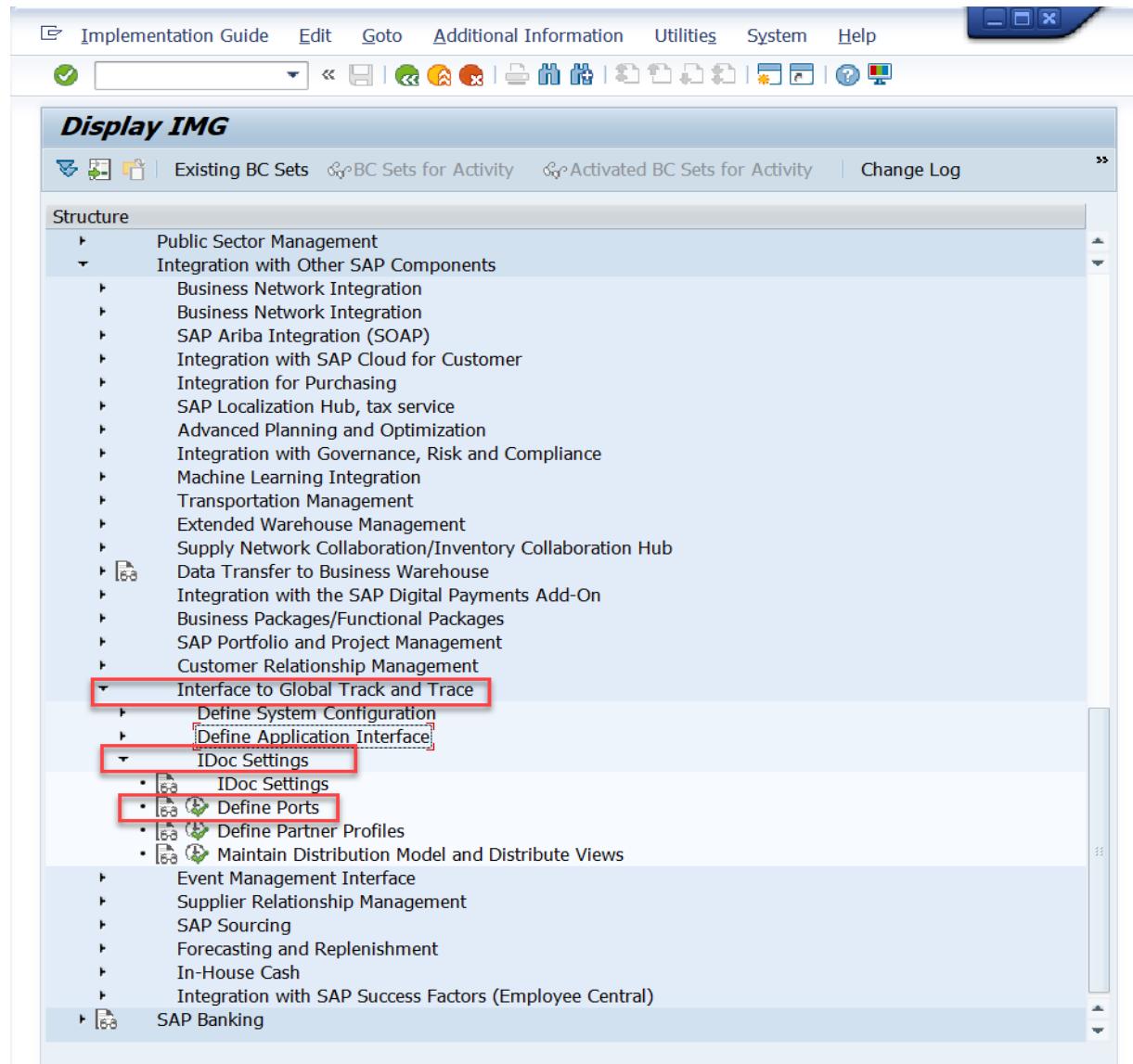
- Logical system code and
- Name of the new logical system

4. Save the configuration

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

STEP 3: Define Ports

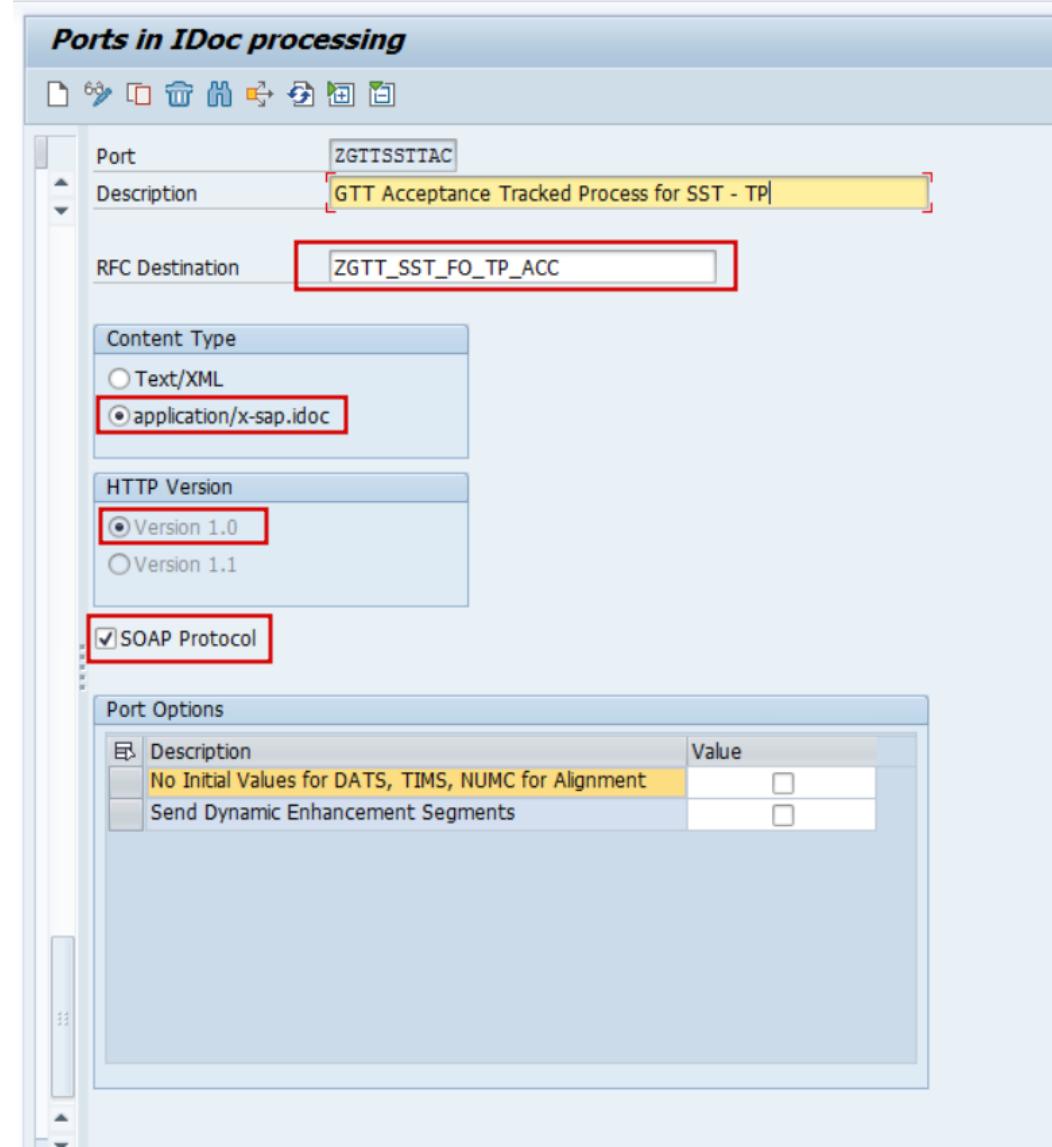
1. In Display IMG page, click **Integration with Other SAP Components -> Interface to Global Track and Trace -> IDoc Settings**
2. Choose activity **Define Ports**



STEP 3: Define Ports

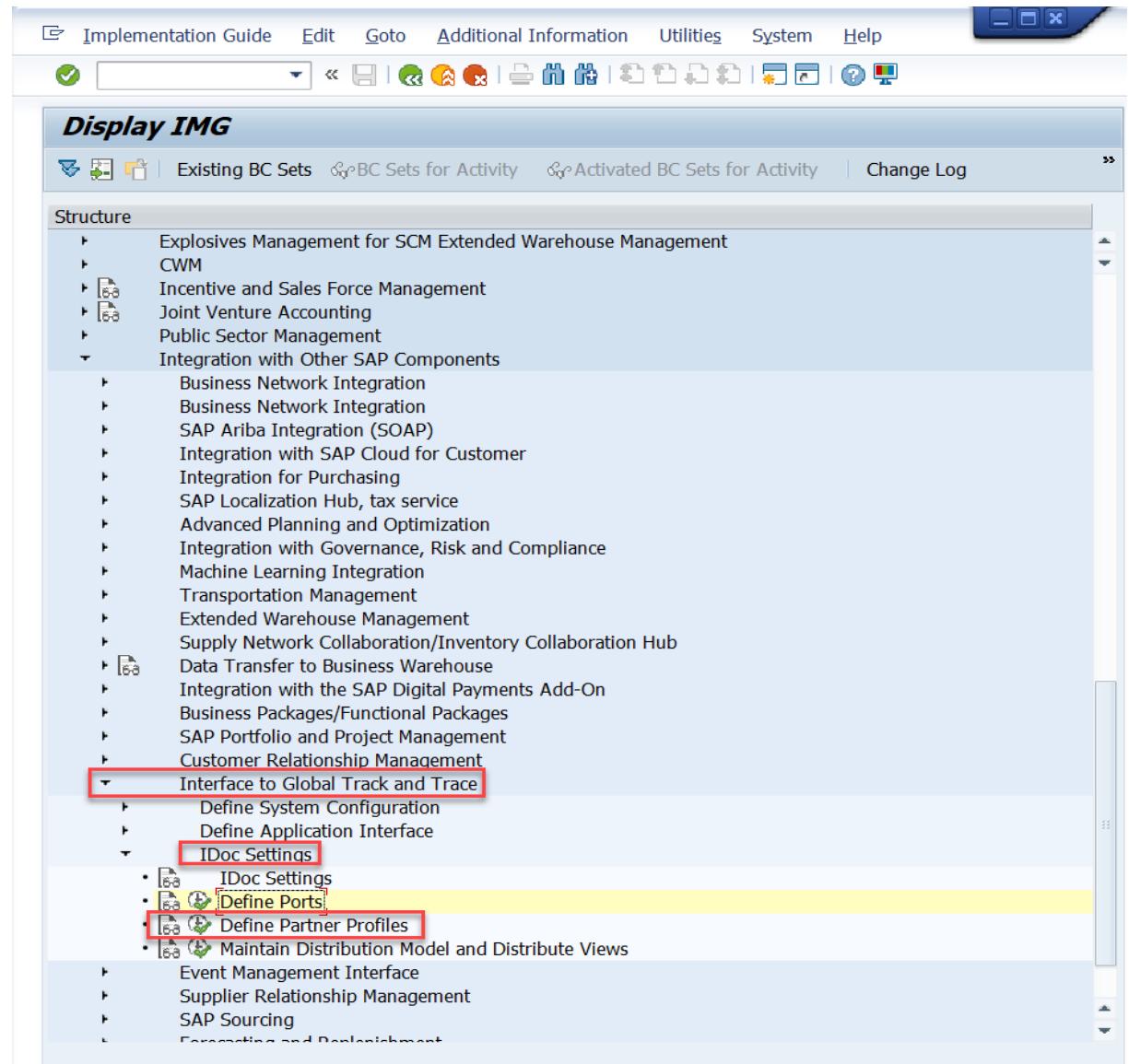
3. Choose **XML HTTP** folder, and click **Create** to create a new port
4. Fill in the **RFC Destination**, it is the RFC connection you created in STEP 1
5. Choose **Content Type** as *application/x-sap.idoc*
6. Choose **HTTP Version** as *Version 1.0*
7. Mark it as SOAP Protocol
8. Save the configuration

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



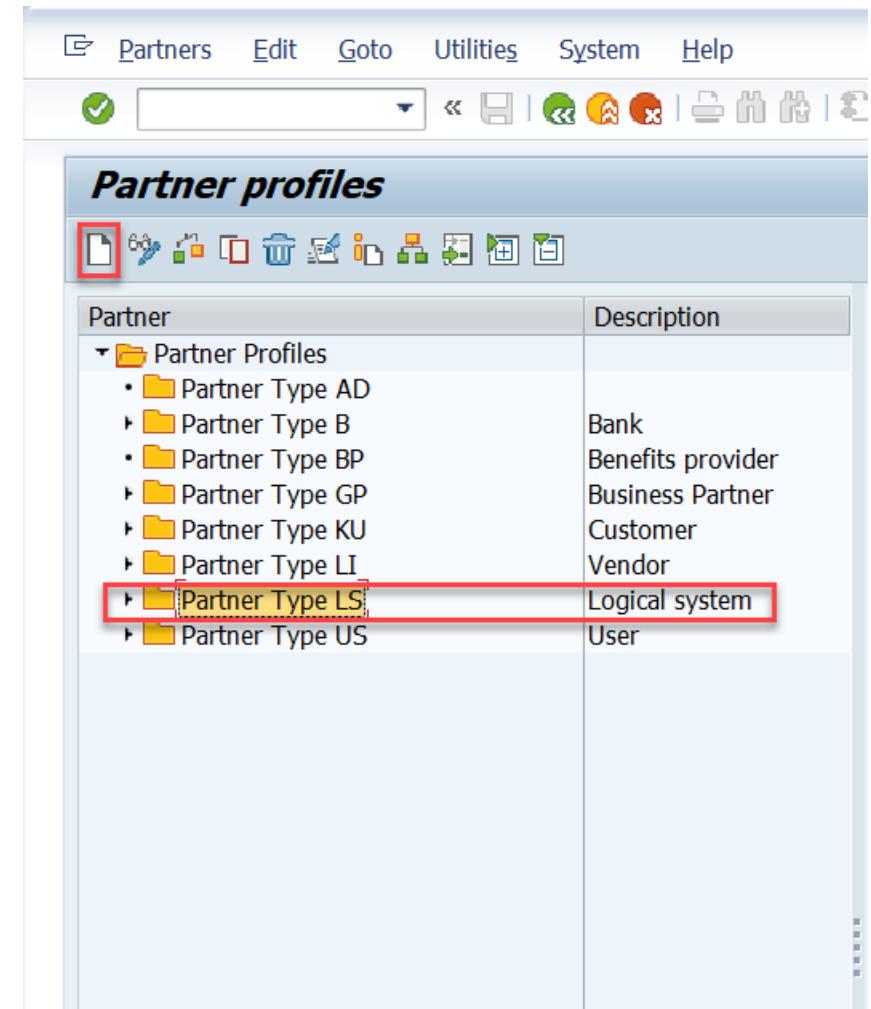
STEP 4: Define Partner Profiles

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 title bar includes 'Partners', 'Edit', 'Goto', 'Utilities', 'System', and 'Help'. Below the title bar 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' folder under the 'Partner' column. The table data is as follows:

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

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 divided into several sections:

- Partner No.:** ZGTTSSSTAC (Logical System For GTT SST - Accept) | Type: LS (Logical system). This section is highlighted with a red box.
- Processor:** Ty.: US | Processor: User (highlighted with a red box) | Lang.: EN (English)
- Outbound:** A table showing message types and their destinations. The table has columns: Partner Role, Message Type, Message Va..., Function, Test, Receiver P..., I..., Pa..., Basic Type. Data rows include:
 - AOPOST (Test: ZGTTSSSTAC, Receiver P...: EHPOST01, Basic Type: EHP01)
 - EVMSTA (Test: ZGTTSSSTEAC, Receiver P...: EVMSTA02, Basic Type: EVM02)
- Inbound:** An empty table with columns: Partner Role, Message Type, Message Va..., Function, Test, P.., Process Code.

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. ZGTTSSSTAC (Logical System For GTT SST - Accept), Type LS (Logical system). Underneath, there are tabs for Post Processing: Valid Processors, Classification, and Telephony. The 'Post Processing: Valid Processors' tab is selected, showing fields for Ty. (US), Processor (blurred), and Lang. (EN English). In the bottom half of the screen, there are two tables: 'Outbound' and 'Inbound'. The 'Outbound' table has columns for Partner Role, Message Type, Message Value, Function, Test, Receiver P..., I..., Pa..., and Basic Type. It contains rows for AOPOST and EVMSTA, both associated with ZGTTSSITAC and ZGTTSSSTEAC respectively. The 'Inbound' table has columns for Partner Role, Message Type, Message Value, Function, Test, P..., and Process Code. It currently has four empty rows.

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

Partner Role	Message Type	Message Va...	Function	Test	P...	Process Code
				<input type="checkbox"/>		
				<input type="checkbox"/>		
				<input type="checkbox"/>		
				<input type="checkbox"/>		

STEP 4: Define Partner Profiles

7. Fill in the Message Type.

For the event:

Message Type: EVMSTA

For the tracked Process:

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.

Partner profiles: Outbound parameters

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system
Partner Role		
Message Type	EVMSTA	
Message Code		
Message Function		<input type="checkbox"/> Test

Outbound Options

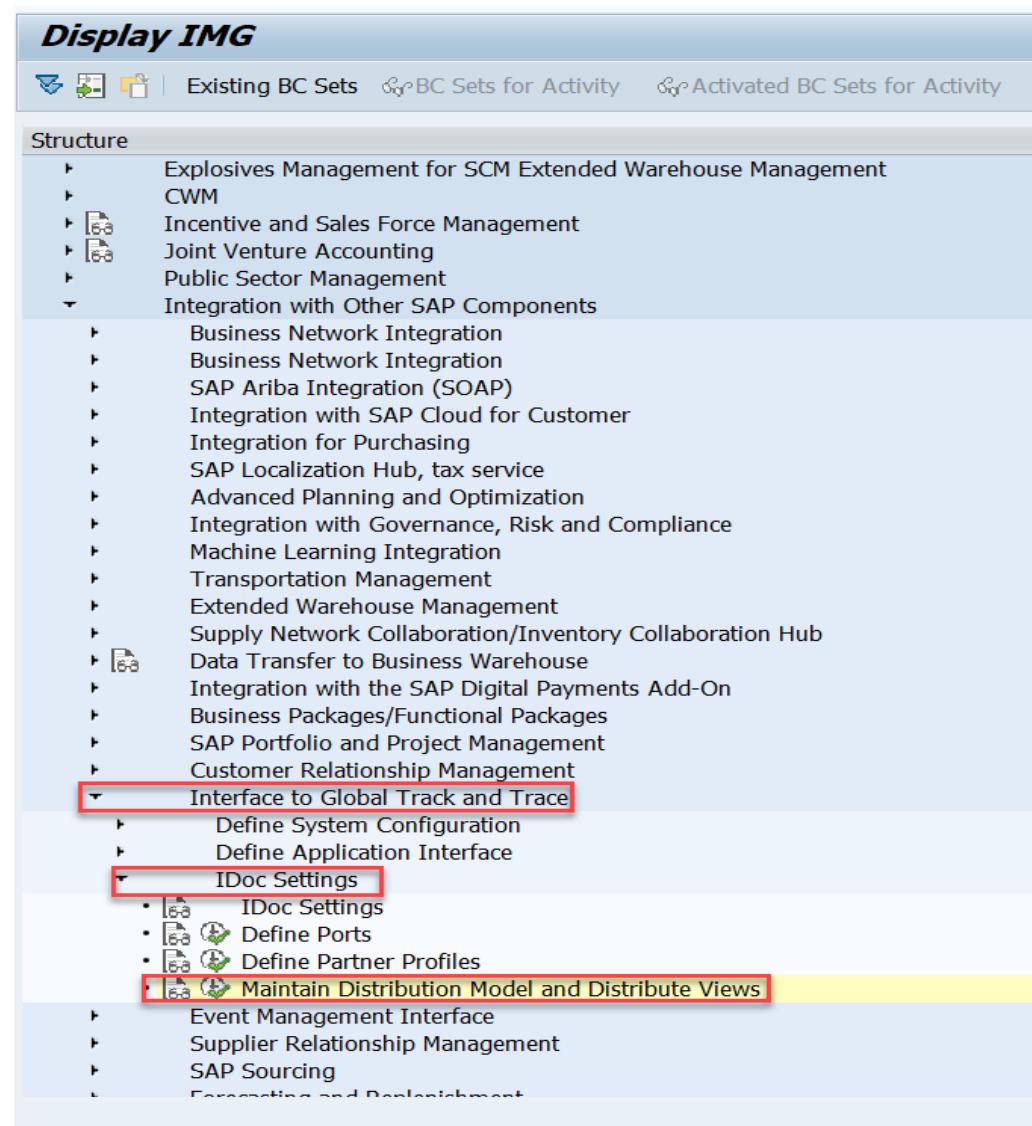
Receiver Port	ZGTTSSSTEAC	GTT Acceptance Tracked Proc...
Pack. Size		
<input type="checkbox"/> Queue Processing		
Output Mode		
<input checked="" type="radio"/> Pass IDoc Immediately		Output Mode 2
<input type="radio"/> Collect IDocs		

IDoc Type

Basic Type	EVMSTA02	SCEM: Event Message Input
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		<input type="checkbox"/> Application Release

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.

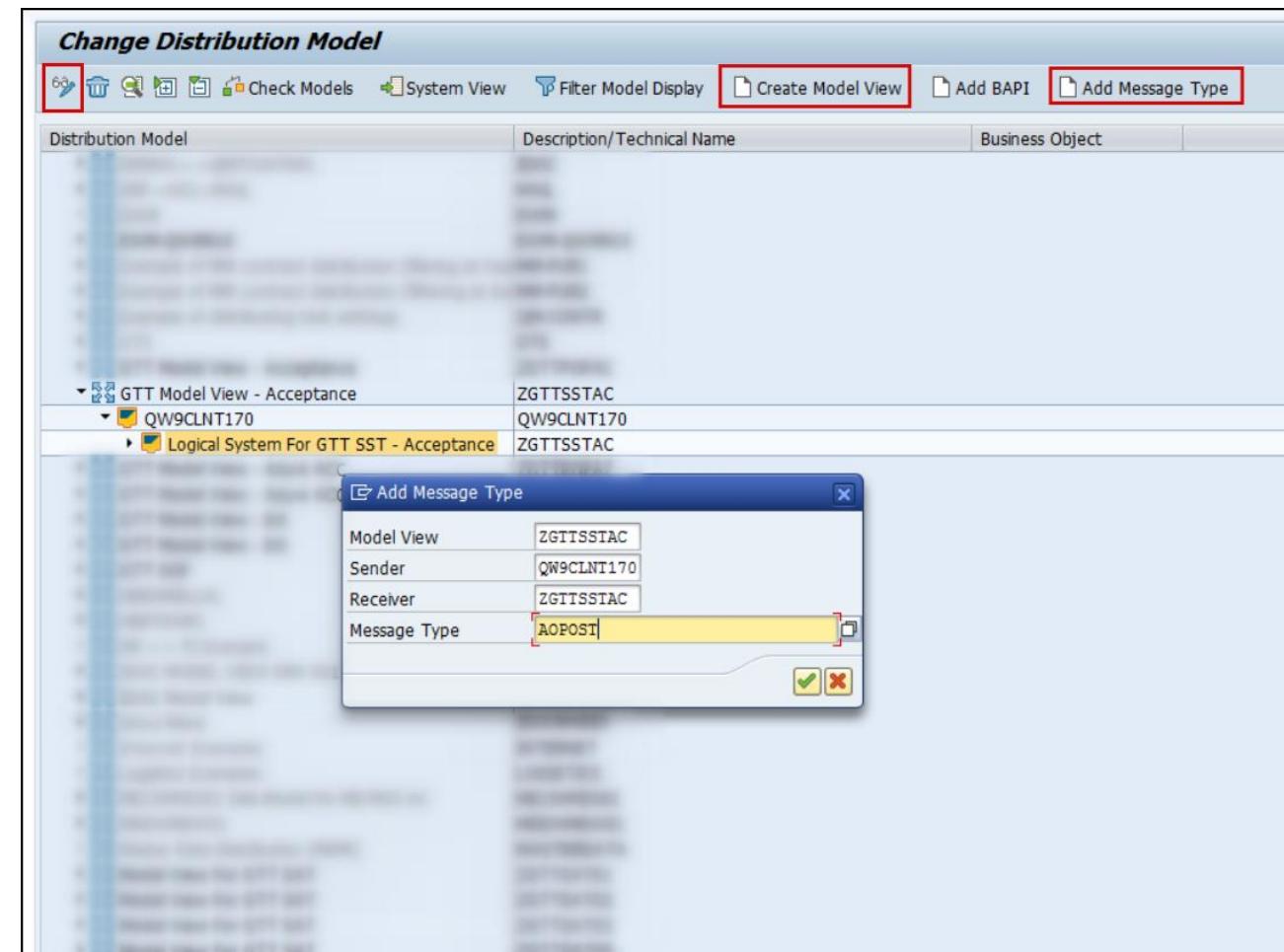
For the event:

Message Type: EVMSTA

For the tracked Process:

Message Type: AOPOST

7. Save the configuration



B) Configuration and Implementation

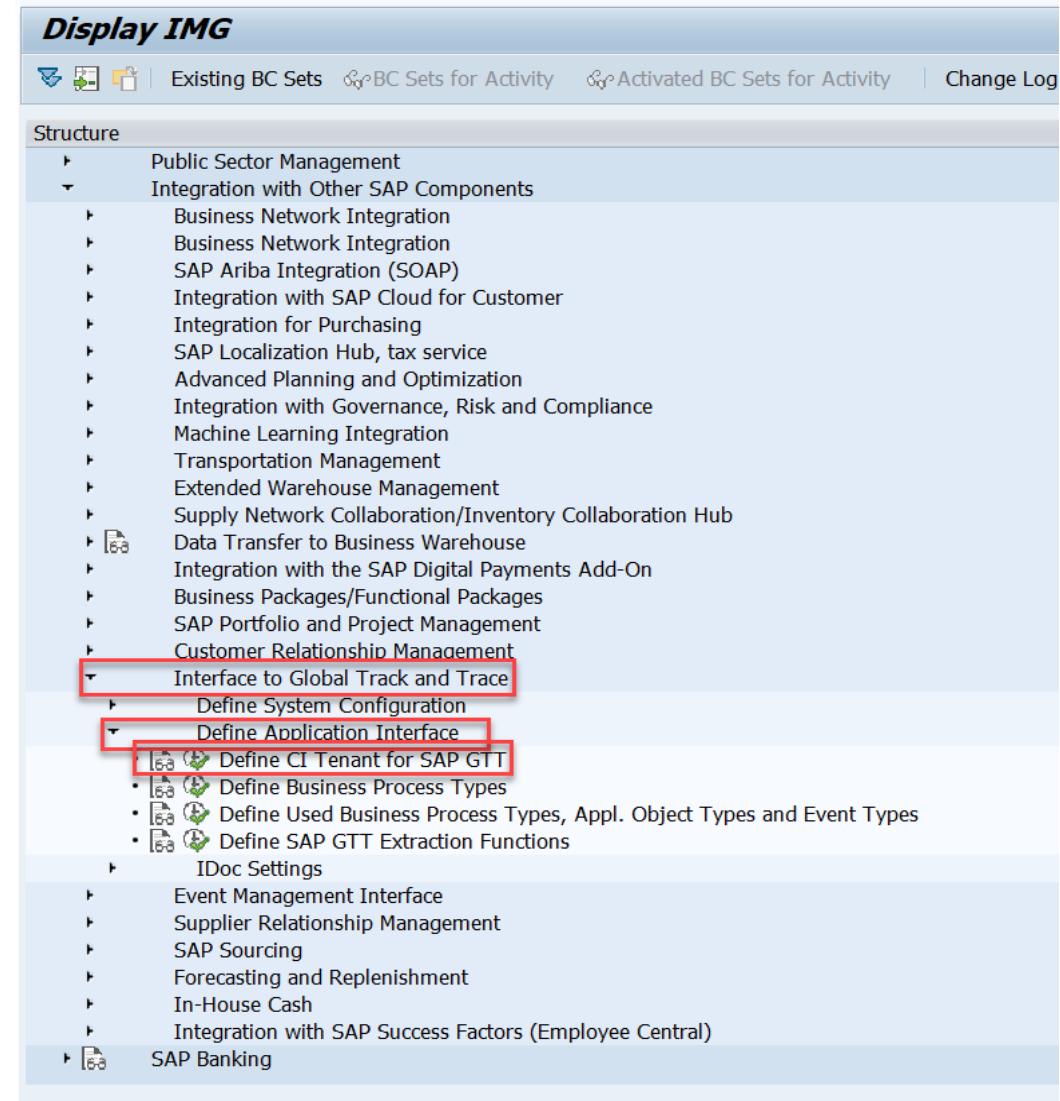
- Basic

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.

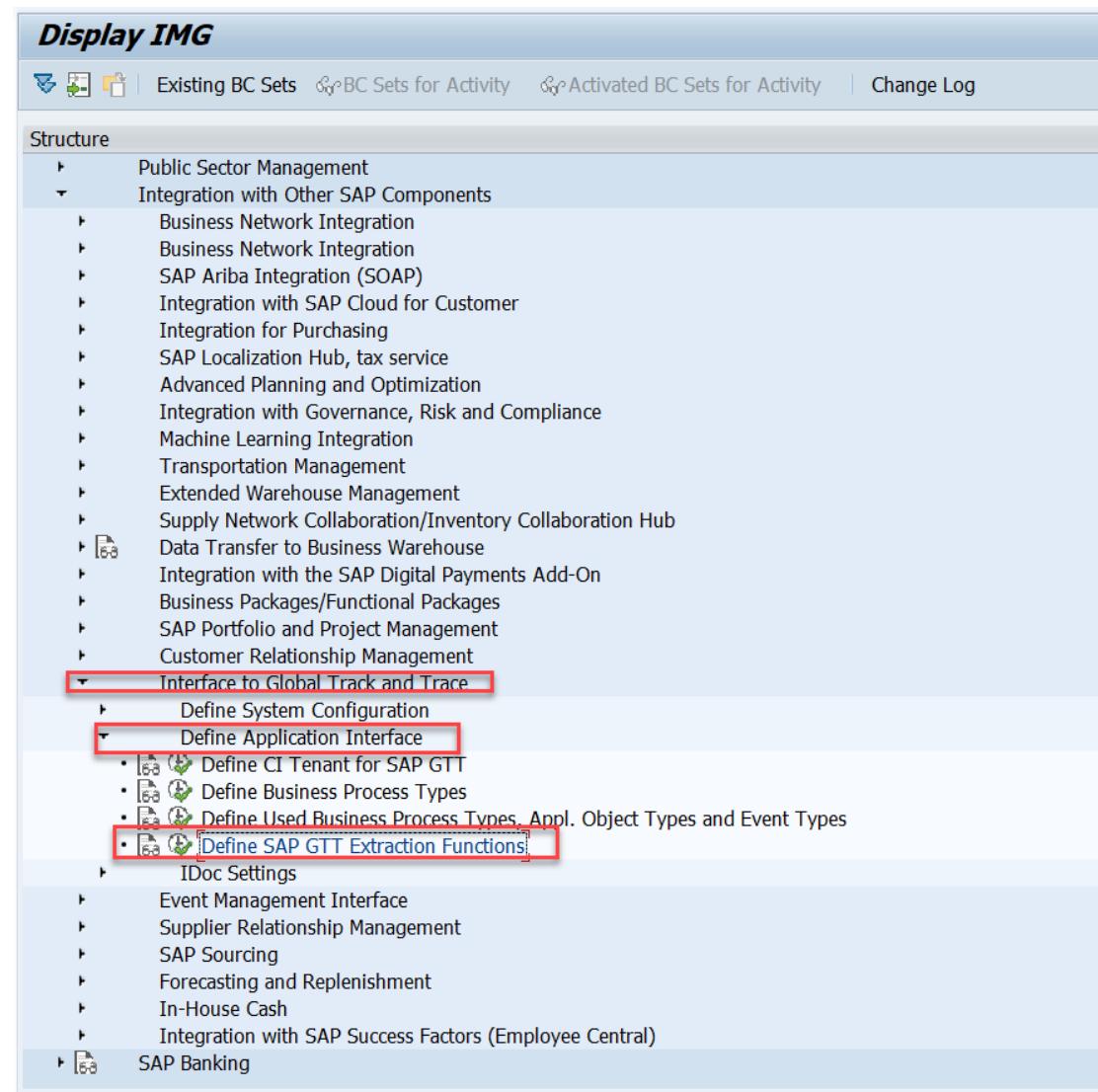
The screenshot shows the SAP Global Track & Trace Definitions Overview screen. At the top, there is a toolbar with various icons. Below the toolbar, the title "Change View 'SAP Global Track & Trace Definitions': Overview" is displayed. A sub-toolbar below the title has a "New Entries" button highlighted with a red box. The main area is a table titled "SAP Global Track & Trace Definitions". The table has four columns: "CI for Global Track & Trace", "CI Log. System", "SAP Track & Trace Version", and "Description". There is one visible row in the table. The "CI for Global Track & Trace" column contains "ZGTTSSSTAC", the "CI Log. System" column contains "ZGTTSSSTAC", the "SAP Track & Trace Version" column contains "Global Track & Trace", and the "Description" column contains "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

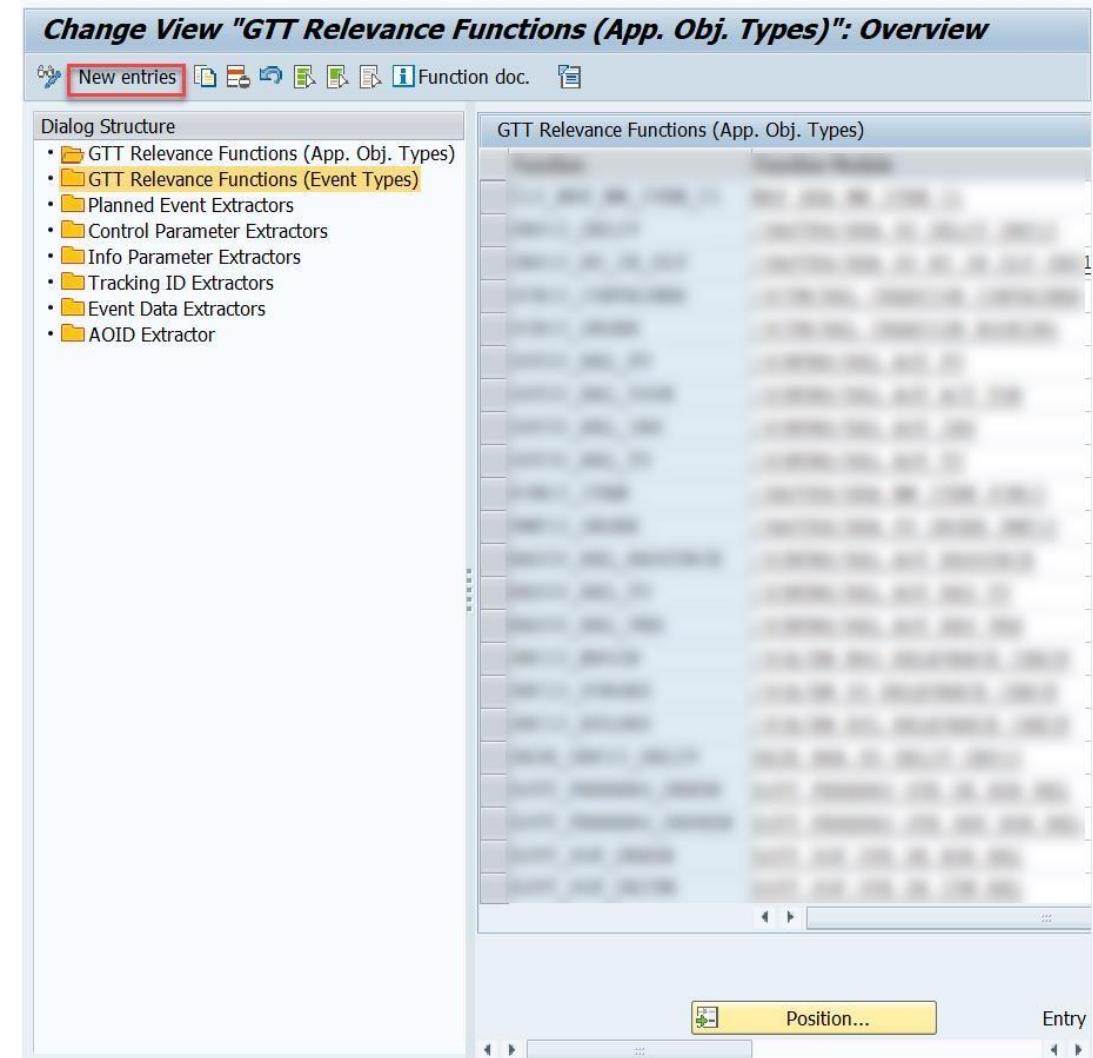
1. In Display IMG page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

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

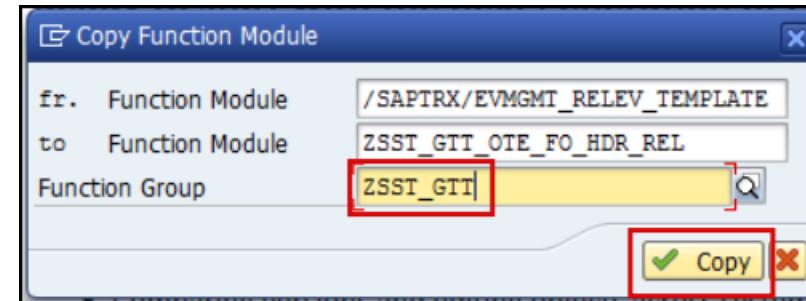
6. If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



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

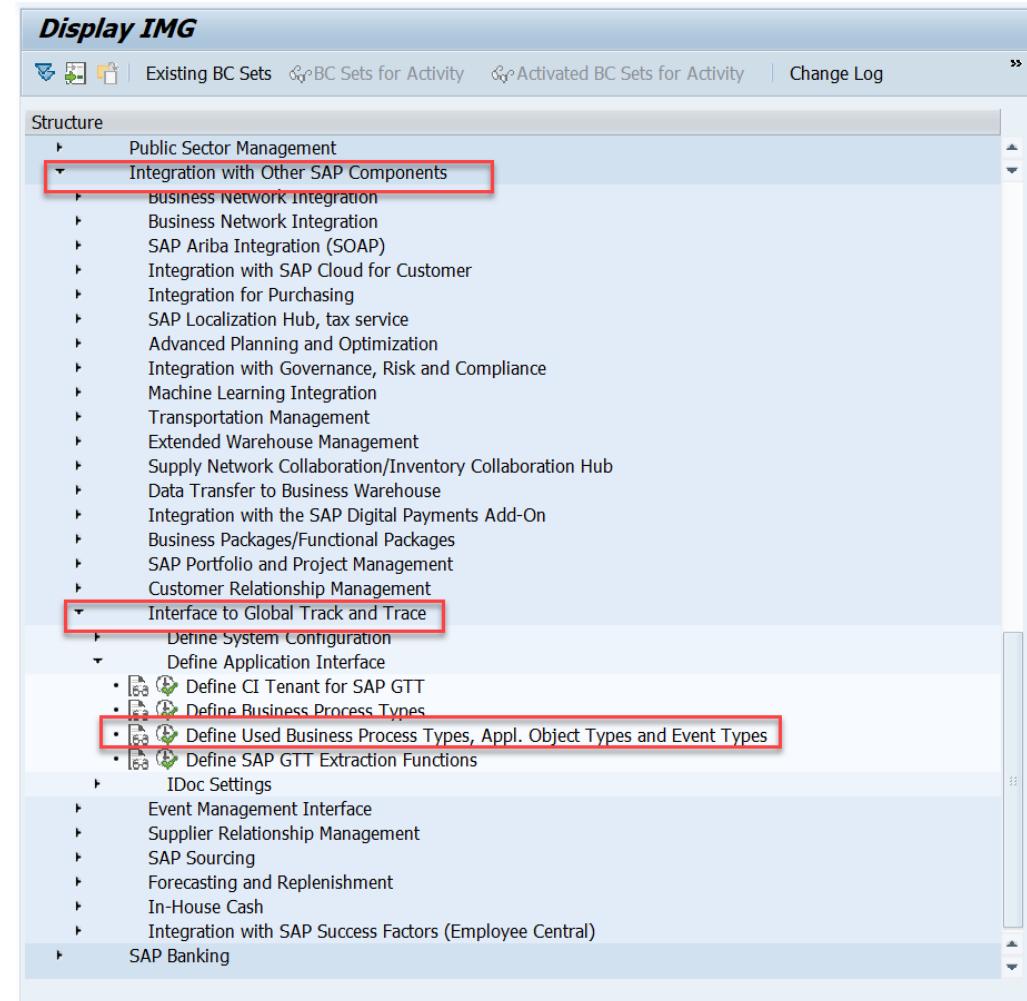
Caution: More information on how to implement extraction functions and the relevant sample code is introduced later.

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 defines a function with various imports, exports, tables, and exceptions. It also includes local data declarations and a TRY...EXCEPT block. A red box highlights the line "DATA: lt_app_objects TYPE trxsas_appobj_ctabs,".
- Code View:** The code is displayed in a syntax-highlighted format with line numbers.
- Status Bar:** The status bar at the bottom right indicates "Scope: FUNCTION ZSST_GTT_OTE_FO_HDR_REL", "ABAP", "Ln 9 Col 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 (▼ 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

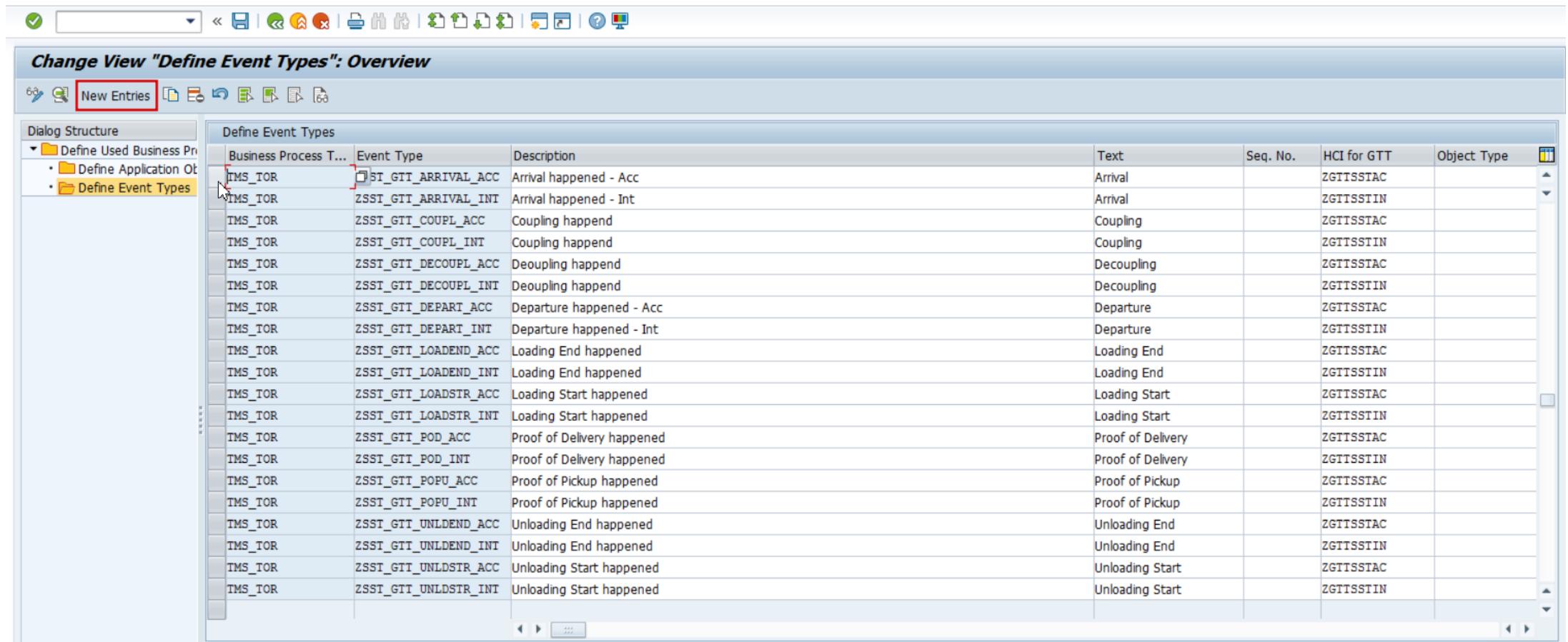
4. Double click **Define Event Types**

The screenshot shows the SAP Fiori interface for managing business process types. On the left, a tree view titled 'Dialog Structure' shows nodes like 'Define Used Business Process Types', 'Define Application Objects', and 'Define Event Types'. The 'Define Event Types' node is highlighted with a red box. On the right, a table titled 'Define Used Business Process Types' lists various entries. One entry, 'TMS_TOR', is highlighted with a red box and a yellow background, indicating it is selected. The table columns include 'Bus. Proc. Type', 'Update Mode', 'BPT Process Mode', and 'Description'. The 'Description' column contains detailed text for each entry.

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_NRKORD	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 "Define Event Types" interface. The title bar displays "Change View 'Define Event Types': Overview". Below the title bar is a toolbar with various icons. On the left, there is a "Dialog Structure" sidebar with a tree view showing "Define Used Business Pr..." expanded, with "Define Application Obj..." and "Define Event Types" under it. The main area is titled "Define Event Types" and contains a table with the following data:

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

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 the following fields:

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

Below these fields is a navigation bar with tabs: General Data, Control Tables, Object Identification (selected), Global Track & Trace Relevance, and Parameter Setup.

The main configuration area is divided into several sections:

- Sequencing / Destination:** Seq. No. 10, CI for GTT ZGTTSSSTAC, CI For GTT Freight Order Sample APP - Acceptance
- Business Object Reference:** Object Type (empty), BO Setup Fnct. (empty)
- Behavior:** A checkbox labeled "GTT Relevant" is checked and highlighted with a red border. Other options include "Stop AO Determ." and "Appl. Log Deact".
- Alt. BusProcType:** An empty input 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 with the following details:

- General Data:**
 - Bus. Proc. Type: TMS_TOR
 - Appl. Obj. Type: ZGTT_SHP_ACC_HD (highlighted with a red box)
 - Text: Extract freight order header information to Global Track and Trace-Acc
- Control Tables:**
 - Main Obj. Table: TOR_ROOT
 - Master Table: (empty)
- Data Source for Deleted Objects:**
 - Del.Obj. Table: TOR_ROOT
- Object Identification:**
 - Reference Between Main and Master Table:
 - First Field Reference from Main to Master Table
 - Second Field Reference from Main to Master Table
- Global Track & Trace Relevance:**
 - (No specific settings shown)
- Parameter Setup:**
 - (No specific settings shown)

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

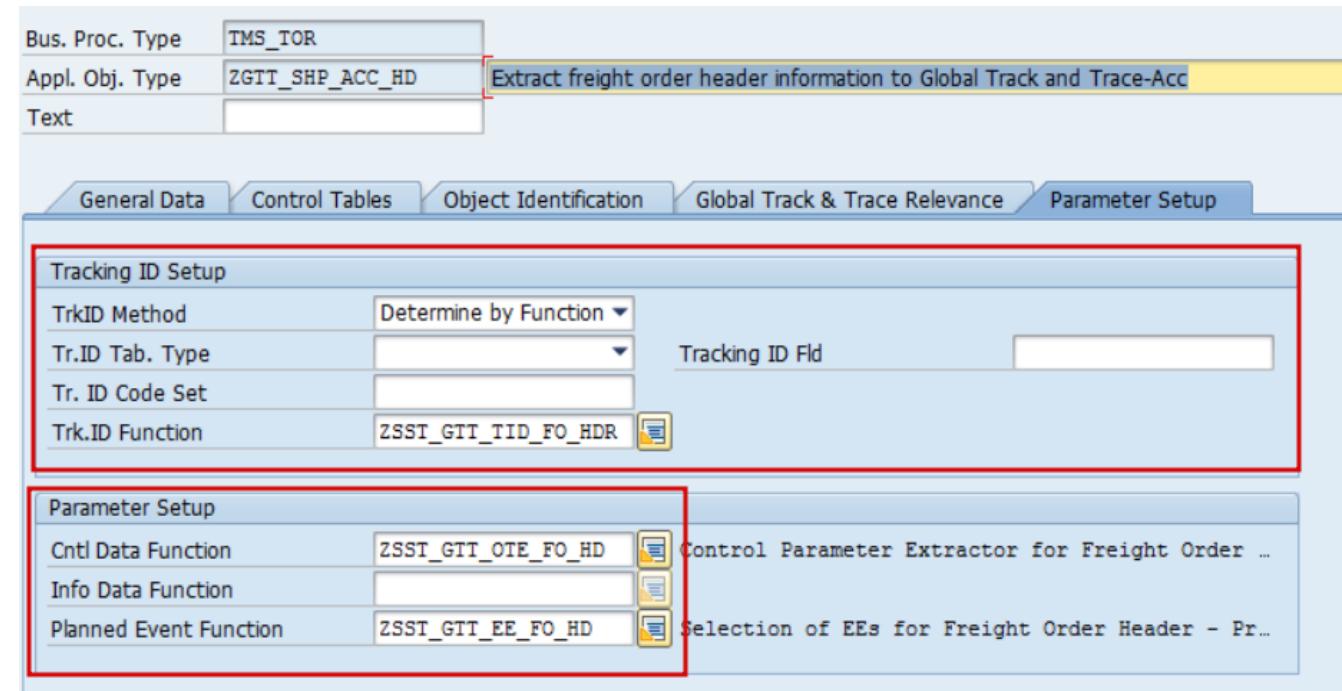
11: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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.

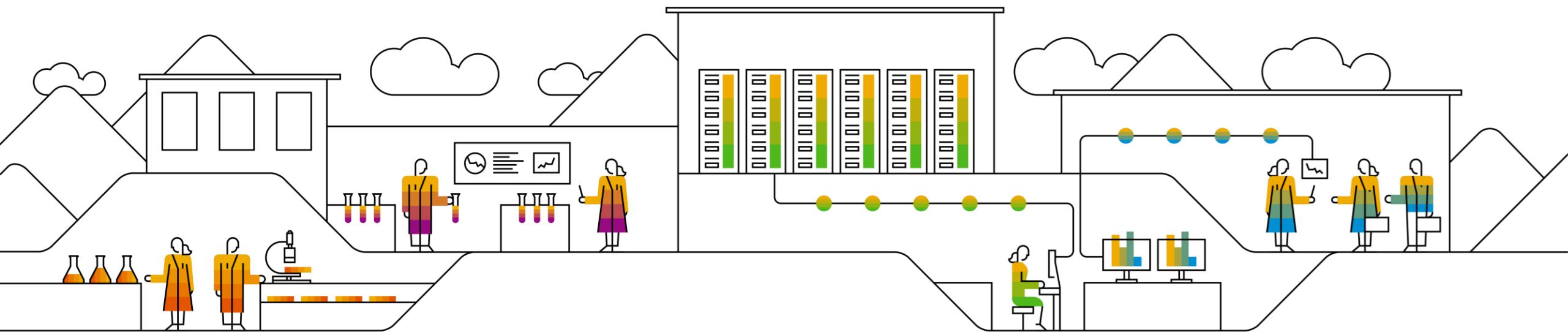
If no customized logic exists, for **TrkID Method** choose *Determine from Field*, then you need to fill the key field and name the Code Set for the AOT.

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.

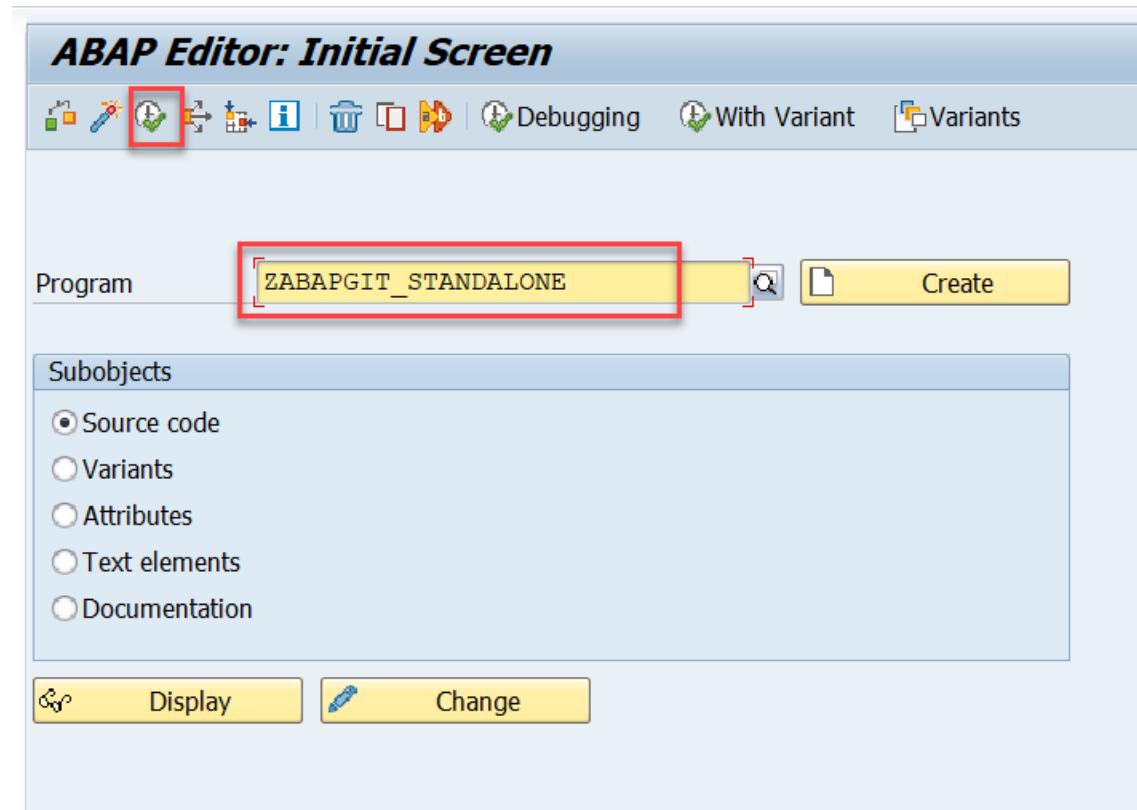
When installation is complete, a new report is created, **ZABAPGIT_STANDALONE**.

The screenshot shows the abapGit documentation website. The header reads "abapGit › documentation". The left sidebar has sections for "Getting Started" (with links to Installation, Upgrading, Uninstalling, and UI features), "Setup" (with links to SSL setup, Proxy configuration, and Development version), "Online Projects" (with links to Installing online repo, Keeping code up to date, Uninstall repository, First project, Moving package into git, and Contributing to a project), "Offline Projects" (with links to Import zip and Export zip), and "Reference" (with links to Repo Settings (.abapgit.xml), Supported object types, Icon Legend, User Exits, Authorizations, and Namespaces). The main content area starts with "Installation" (with a "Improve this page" link) and "Summary #". It states that abapGit exists in two flavours: "standalone" and "developer". The "standalone" version is targeted at users and consists of one program run via transaction SE38. The "developer" version is targeted at developers and consists of ABAP programs/classes/interfaces/etc. of the abapGit project, run via transaction ZABAPGIT. Below this is a "Prerequisites #" section stating that abapGit requires SAP BASIS version 702 or higher. A red box highlights the "Install standalone version #" section, which contains a numbered list of steps: 1. Download 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 says 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 says 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 followed by 'abapGit' and 'Repository List'. To the right of the navigation are buttons for 'New Online' (highlighted with a red box), 'New Offline', and other options. A search bar labeled 'Filter:' is present, along with checkboxes for '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. The bottom of the page features a footer with the 'abapGit' logo and version '1.98.0', and a status message 'js: OK'.

STEP 2: Download ABAP Code

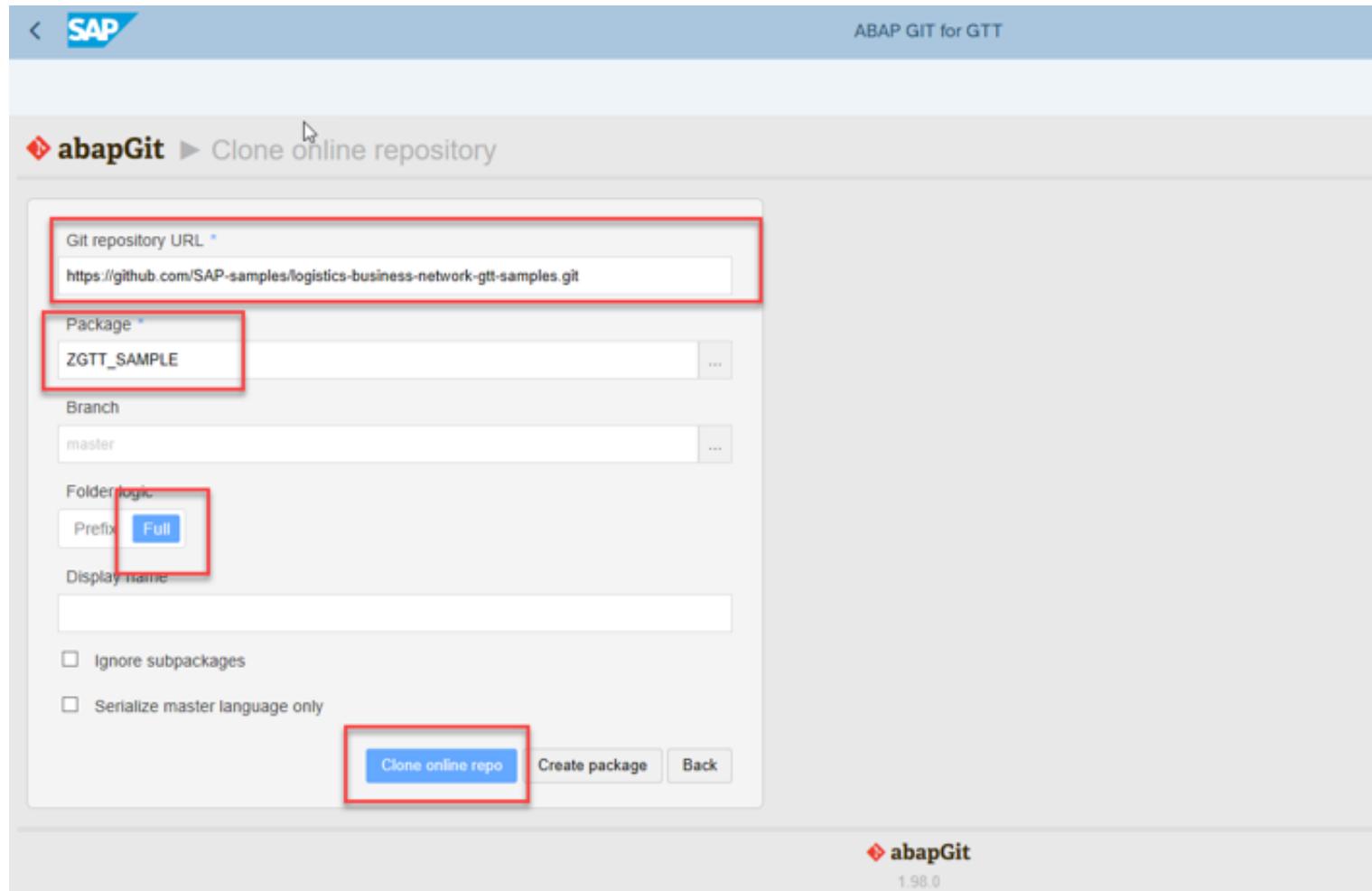
2-4: Fill in the **Git repository URL**:

<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

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

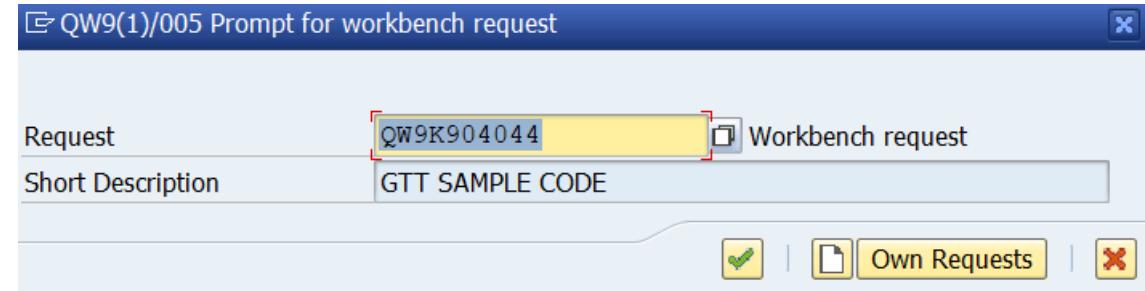
TIP: Clicking **Pull** action will download the whole package of the sample codes. If the user want to download the specified folder's codes in Github, please check details in Step 3-1 – 3-8

The screenshot shows the ABAP GIT for GTT interface. At the top, there is a navigation bar with the title "ABAP GIT for GTT", a logo for "abapGit", and a link to "Repository List". Below the navigation bar, the main area displays a list of repositories. One repository is selected: "logistics-business-network-gtt-samples" with the URL "https://github.com/SAP-samples/logistics-business-network-gtt-samples.git". The repository page shows a table of contents with various files and folders. A red box highlights the "Pull" button in the toolbar above the table. The table includes columns for file paths and status indicators (diff, M, A). At the bottom of the page, there is a footer with the "abapGit" logo and the text "js: OK".

non-code and meta files		
	/.abapgit.xml	diff
	/NOTICE	A
AVAS	/src/0894ef4577391eeaab910bd805b24f18.avas.xml	diff
CLAS	/src/zcl_gtt_sof_im_le_shipping.clas.abap	diff
	/src/zcl_gtt_sof_im_le_shipping.clas.xml	A
DEVC	/src/package.devc.xml	diff
TABL	/src/zgtt_sof_ee_rel.tabl.xml	A

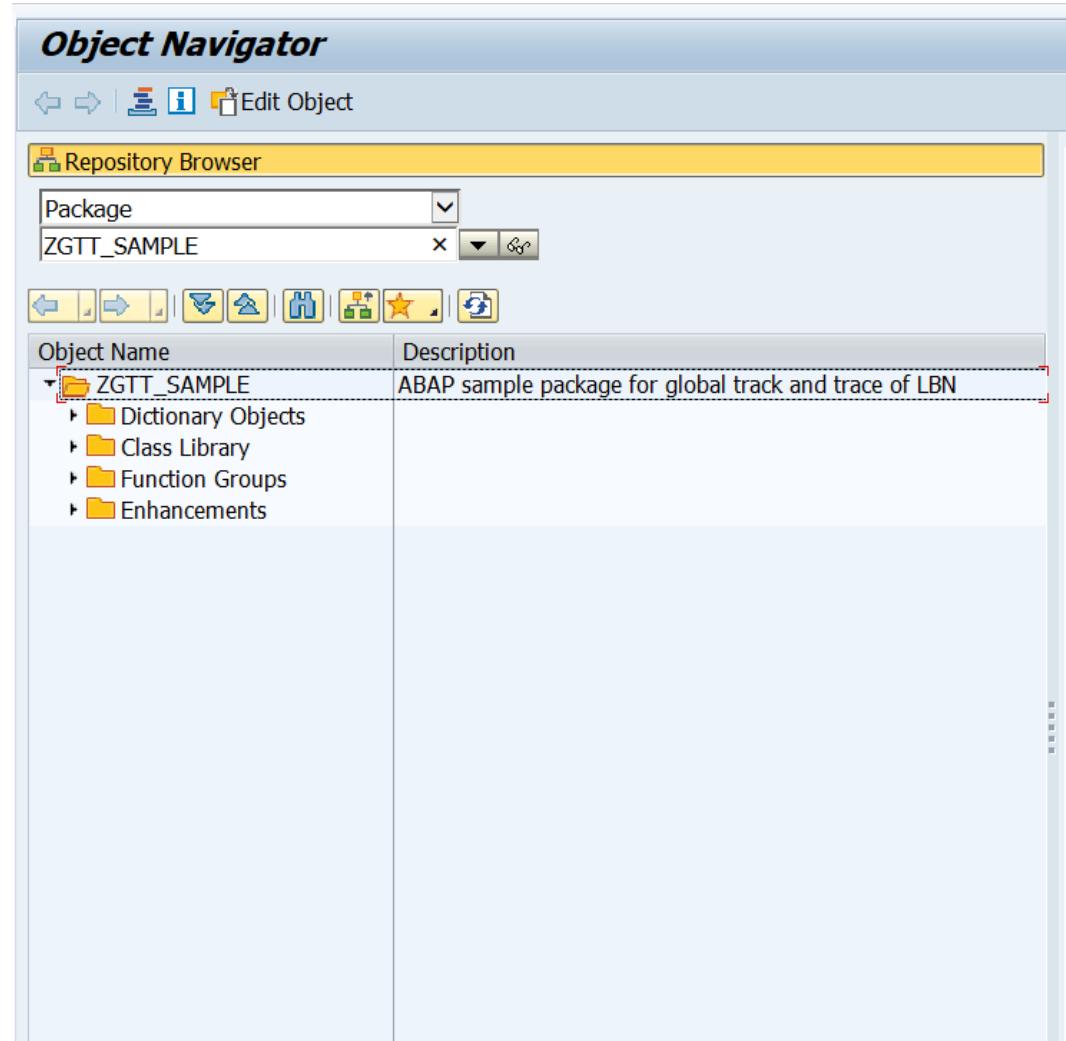
STEP 2: Download ABAP Code

2-9: 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

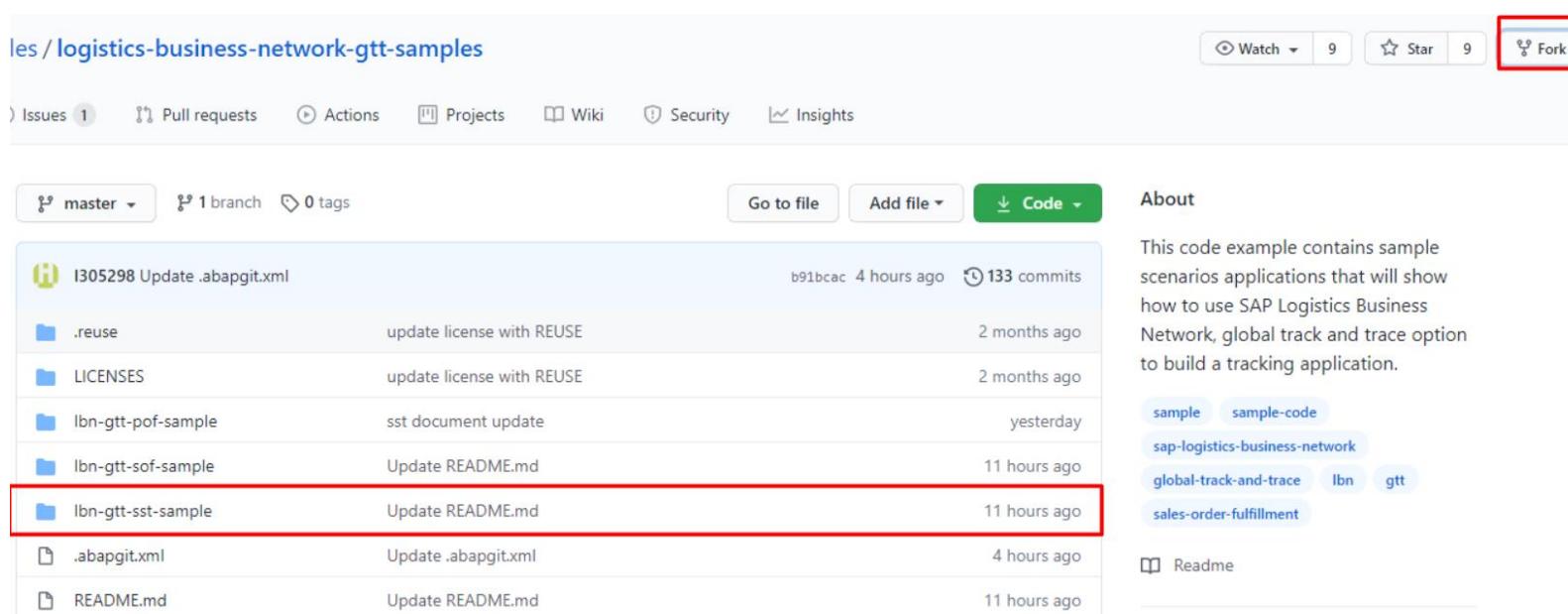
2-10: After you download the code, you can check them with T-code *SE80*.



Step 3: Download ABAP Code within the specified folder

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

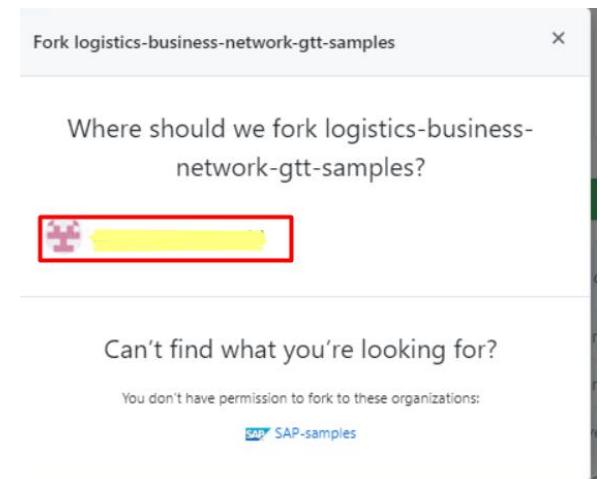
3-2: Click the user account and it will copy the newest version into the user's account.



The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples'. The top navigation bar includes 'Issues 1', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. Below the navigation is a summary: 'master' branch, '1 branch', '0 tags', 'Go to file', 'Add file', and a green 'Code' button. The main area displays a list of commits:

- I305298 Update .abapgit.xml (b91bcac, 4 hours ago, 133 commits)
- .reuse update license with REUSE (2 months ago)
- LICENSES update license with REUSE (2 months ago)
- Ibn-gtt-pof-sample sst document update (yesterday)
- Ibn-gtt-sof-sample Update README.md (11 hours ago)
- Ibn-gtt-sst-sample Update README.md (11 hours ago)** (highlighted with a red box)
- .abapgit.xml Update .abapgit.xml (4 hours ago)
- README.md Update README.md (11 hours ago)

To the right of the commits is an 'About' section with a detailed description of the code example and its purpose. Below the 'About' section are several blue buttons representing tags or categories: 'sample', 'sample-code', 'sap-logistics-business-network', 'global-track-and-trace', 'Ibn', 'gtt', and 'sales-order-fulfillment'. At the bottom of the page is a 'Readme' link.

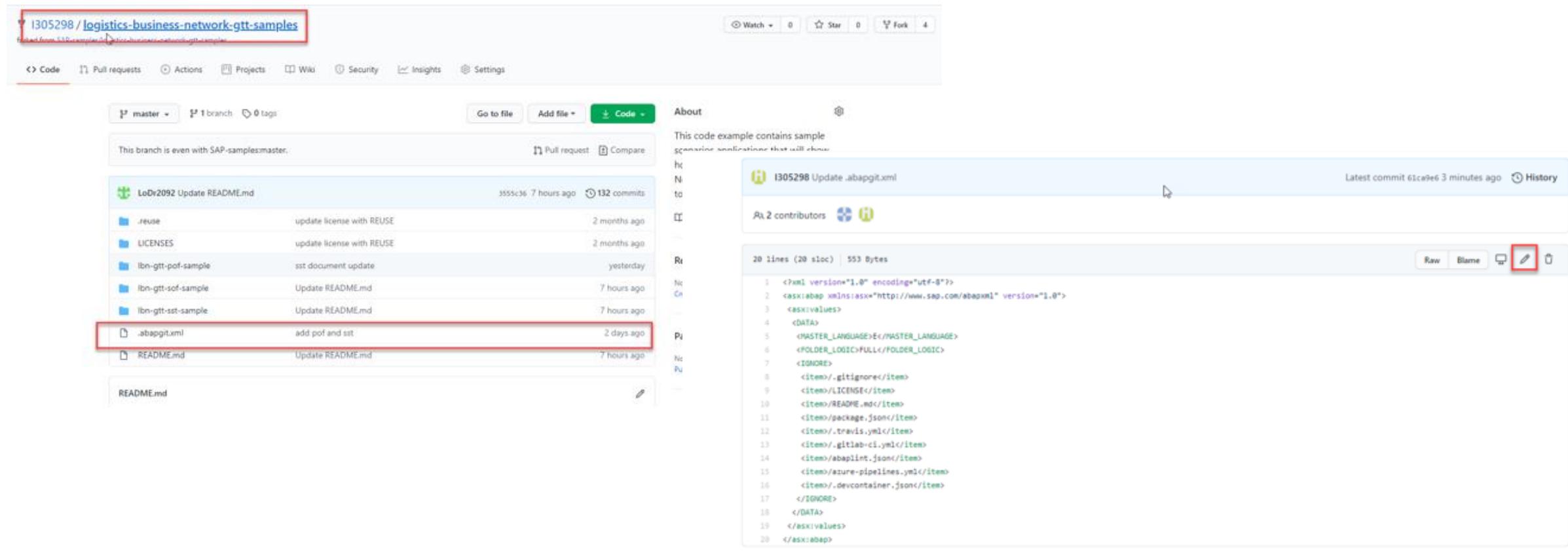


The screenshot shows a 'Fork' dialog window titled 'Fork logistics-business-network-gtt-samples'. It asks 'Where should we fork logistics-business-network-gtt-samples?'. A dropdown menu is open, showing a list of organizations, with the first item partially visible ('**SAP** SAP-samples') and the rest blurred. Below the dropdown is a message: 'Can't find what you're looking for?' followed by 'You don't have permission to fork to these organizations: SAP 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 'i305298 / logistics-business-network-gtt-samples'. The repository has 0 stars and 4 forks. The 'Code' tab is selected, showing a list of files. A red box highlights the '.abapgit.xml' file under 'LoDr2092 Update README.md'. Another red box highlights the 'Edit' button (pencil icon) in the top right corner of the code editor area.

This code example contains sample
scenarios annotations that will show

1 i305298 Update .abapgit.xml

Latest commit 61ca9e6 3 minutes ago History

Raw Blame Copy Edit

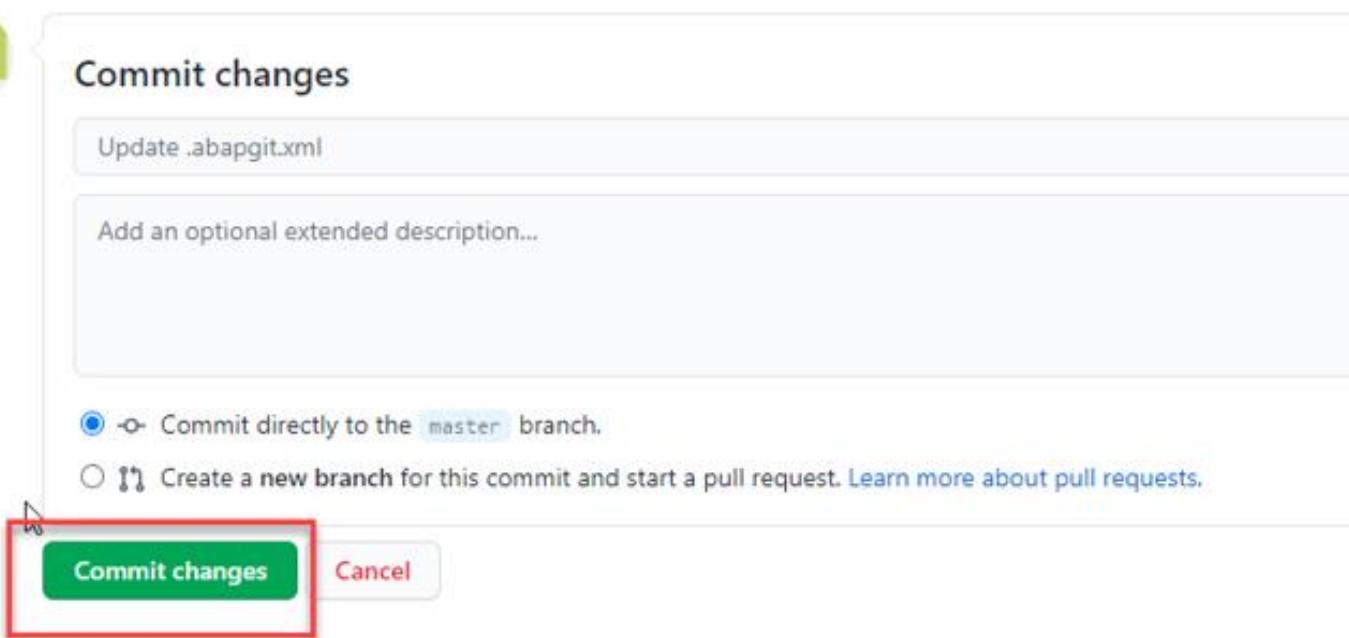
```
<?xml version="1.0" encoding="utf-8"?>
<asxi:abap xmlns:asxi="http://www.sap.com/abapxml" version="1.0">
<asxi:values>
<DATA>
<MASTER_LANGUAGE>E</MASTER_LANGUAGE>
<FOLDER_LOGIC>FULL</FOLDER_LOGIC>
<IGNORE>
<item>/.gitignore</item>
<item>LICENSE</item>
<item>README.md</item>
<item>package.json</item>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

Step 3: Download ABAP Code within the specified folder

3-5: Add the sentence of '<STARTING_FOLDER>/lbn-gtt-sst-sample/ABAP/src/</STARTING_FOLDER>' like below

3-6: Commit change

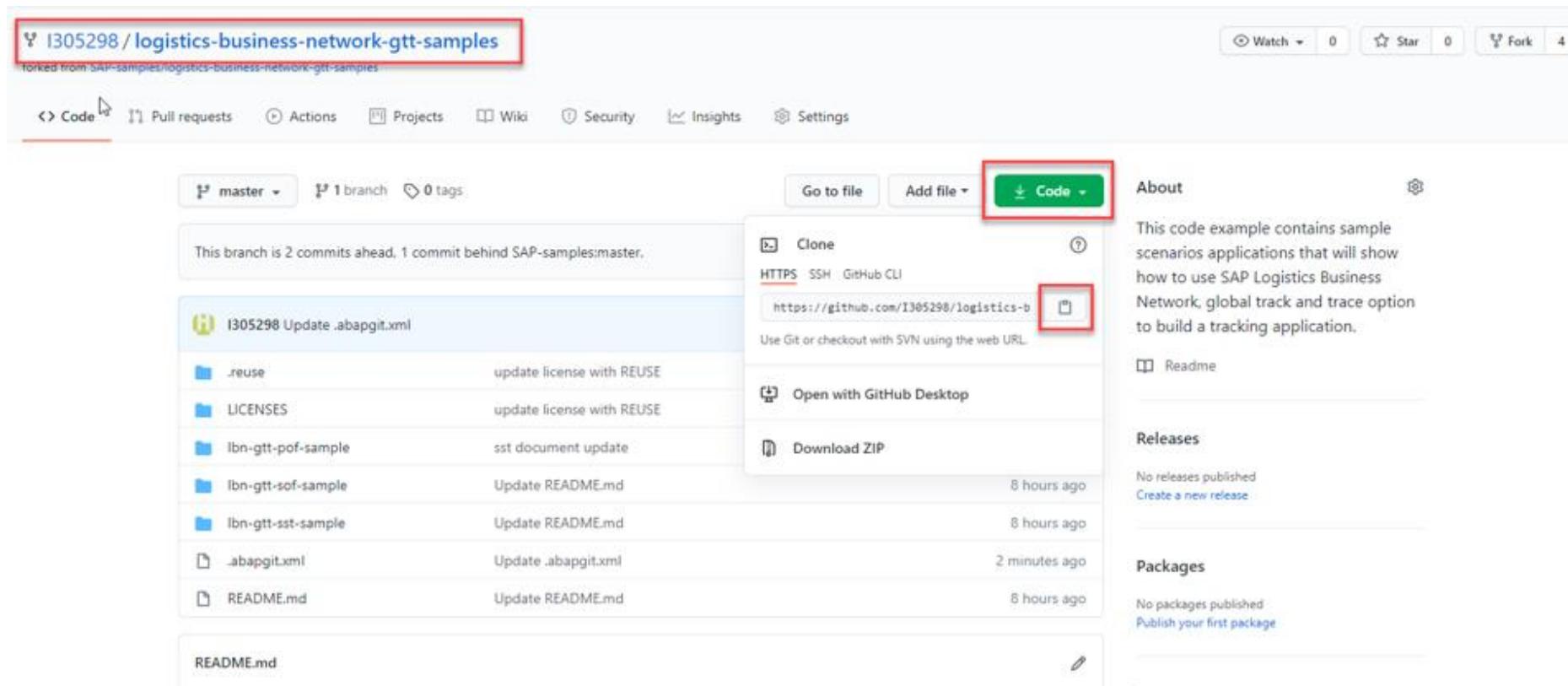
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.
3    <asx:values>
4      <DATA>
5        <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6        <STARTING_FOLDER>/lbn-gtt-sst-sample/ABAP/src/</STARTING_
7          <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8        <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>
```



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



Potential Issue: Failure of New BADI importing

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

The screenshot shows the SAP abapGit interface for a repository named "GTT-V2-Sample-Apps-Sandbox". The repository URL is <https://github.wdf.sap.corp/TrackAndTrace/GTT-V2-Sample-Apps-Sandbox.git>. The current branch is master, and the commit hash is 0f7ed10. The interface includes standard git operations like Pull, Stage, Diff, Log, Branch, Tag, Advanced, Refresh, and a Repository List.

A red box highlights an error message in the top left corner: "error deserializing ENHO badi Import of object [REDACTED] failed".

The main pane displays a list of files and their corresponding XML representations:

File Path	XML Representation	Diff
/abapgit.xml	/bn-gtt-sof-sample/ABAP/src/zgtt_dlv_watch_stop.tabl.xml	diff
ZGTT_DLV_WATCH_STOP	/bn-gtt-sof-sample/ABAP/src/zgtt_kunablaz_txt.dtel.xml	diff
ZGTT_KUNABLAZ_TXT	/bn-gtt-sof-sample/ABAP/src/zgtt_lgnumaz.dtel.xml	diff
ZGTT_LGNUMAZ	/bn-gtt-sof-sample/ABAP/src/zgtt_lgortaz_txt.dtel.xml	diff
ZGTT_LGORTAZ_TXT	/bn-gtt-sof-sample/ABAP/src/zgtt_lgtraz_txt.dtel.xml	diff
ZGTT_LGTRAZ_TXT	/bn-gtt-sof-sample/ABAP/src/zgtt_loccat.dtel.xml	diff
ZGTT_LOCCAT	/bn-gtt-sof-sample/ABAP/src/zgtt_loccat_dm.dom.xml	diff
ZGTT_LOCCAT_DM	/bn-gtt-sof-sample/ABAP/src/zgtt_locid.dtel.xml	diff
ZGTT_LOCID	/bn-gtt-sof-sample/ABAP/src/zgtt_loctype.dtel.xml	diff
ZGTT_LOCTYPE	/bn-gtt-sof-sample/ABAP/src/zgtt_lstelz_txt.dtel.xml	diff
ZGTT_LSTELZ_TXT	/bn-gtt-sof-sample/ABAP/src/zgtt_pln_evt_datetime.dtel.xml	diff
ZGTT_PLN_EVT_DATETIME	/bn-gtt-sof-sample/ABAP/src/zgtt_sof_ee_rel.tabl.xml	diff
ZGTT_SOF_EE_REL	/bn-gtt-sof-sample/ABAP/src/zgtt_stopcnt.dtel.xml	diff
ZGTT_STOPCNT	/bn-gtt-sof-sample/ABAP/src/zgtt_stopid.dtel.xml	diff
ZGTT_STOPID	/bn-gtt-sof-sample/ABAP/src/zgtt_stop_info.tabl.xml	diff
ZGTT_STOP_INFO		diff

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 screenshot shows three SAP Class Builder windows. The top window is the 'Initial Screen' with the title 'Class Builder: Initial Screen'. It has a toolbar with various icons, a search bar, and a menu bar. Below the toolbar is a navigation bar with 'Class Browser' and 'More'. The main area shows an object type 'CL_ENH_BADI_IMPL_UTILITY' selected in a dropdown, and buttons for 'Display', 'Change', and 'Create'. A red box highlights the 'Display' button. The middle window is titled 'Class Builder: Display Class CL_ENH_BADI_IMPL_UTILITY'. It shows the 'Properties', 'Interfaces', and 'Friends' tabs. The 'Properties' tab is active. The 'Method' section lists 'CHECK_FILTER', 'REPAIR_FILTER', and 'ADD_BADIIMPLDIRENTRY'. A red box highlights 'ADD_BADIIMPLDIRENTRY'. The bottom window is titled 'Class Builder Class CL_ENH_BADI_IMPL_UTILITY.Display'. It shows the source code for the 'ADD_BADIIMPLDIRENTRY' method. The code is as follows:

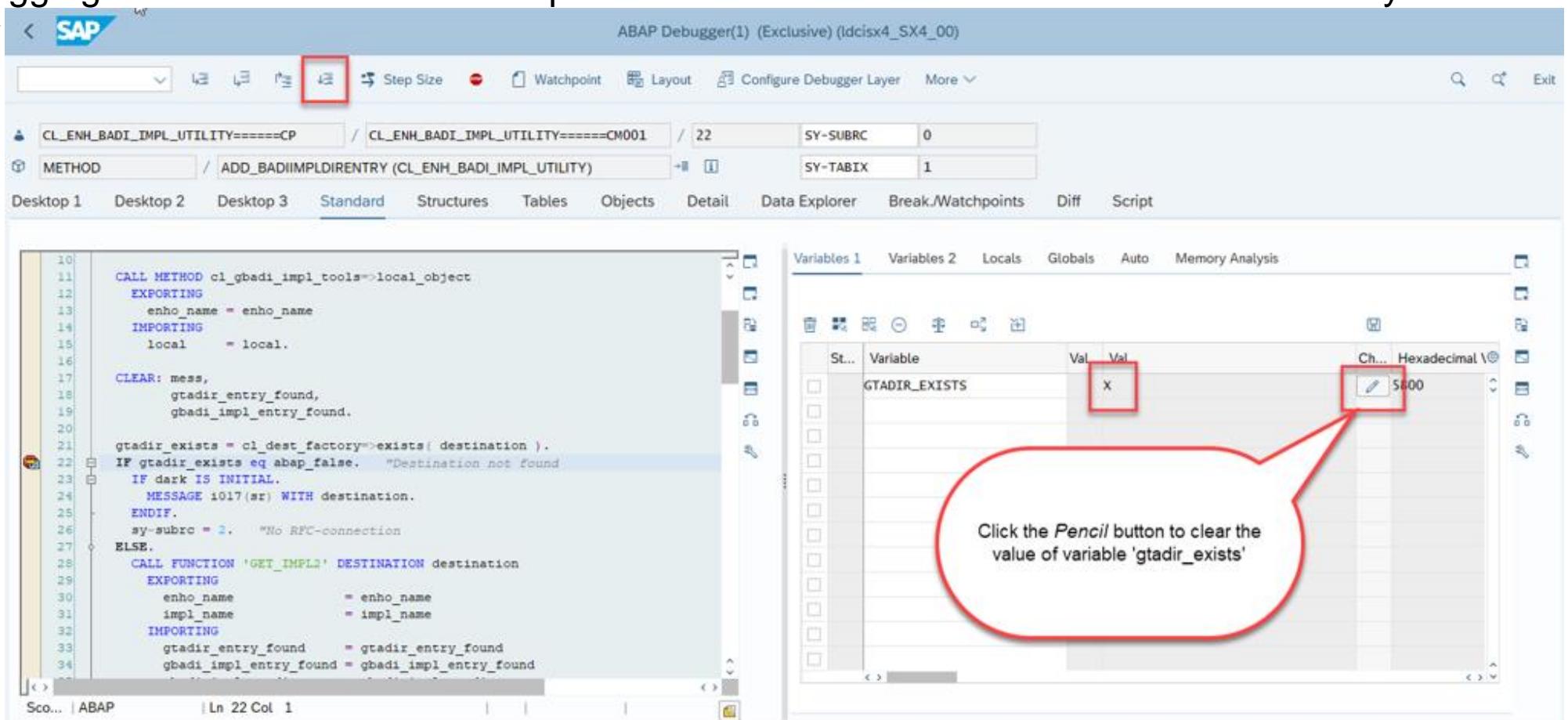
```
Method: ADD_BADIIMPLDIRENTRY
7   gbadiml_entry_found TYPE char1,
8   gtadir_exists      TYPE abap_bool,
9   destination(32) VALUE 'GTADIR_SERVER'.
10
11  CALL METHOD cl_gbadiml_tools>local_object
12    EXPORTING
13      enho_name = enho_name
14    IMPORTING
15      local   = local.
16
17  CLEAR: mess,
18        gtadir_entry_found,
19        gbadiml_entry_found.
20
21  gtadir_exists = cl_dest_factory->exists(destination).
22  IF gtadir_exists eq abap_false. "Destination not found
23  IF dask IS INITIAL.
24    MESSAGE i017(z) WITH destination.
25  ENDIF.
26  sy-subrc = 3. "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      gbadiml_entry_found = gbadiml_entry_found
35      gbadiml_gtadir = gbadiml_gtadir
36    EXCEPTIONS
37      system_failure      = 2 MESSAGE mess
38      communication_failure = 2 MESSAGE mess
39      OTHERS              = 3.
40
```

Two breakpoints are set in the code: one at line 22 and another at line 37. Red boxes highlight both the line numbers and the breakpoints.

Potential Issue: Failure of New BADI importing

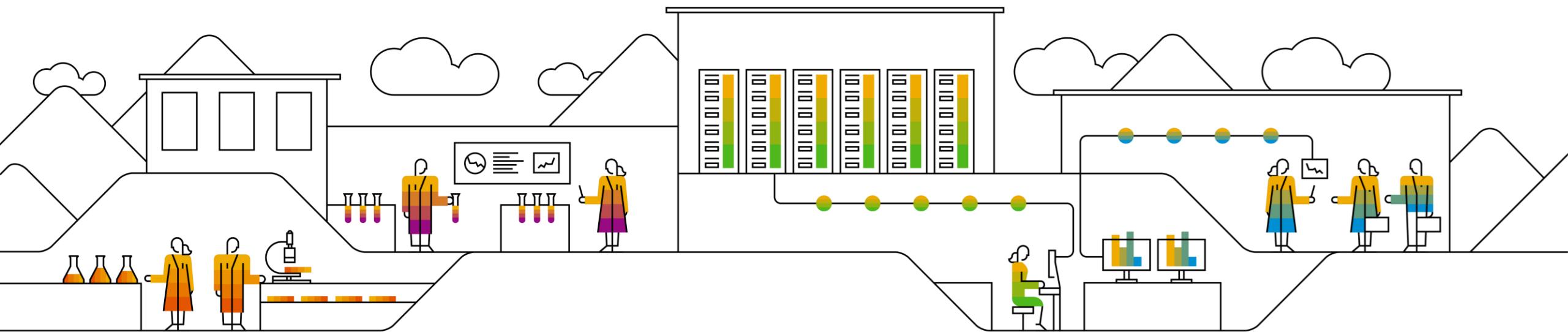
Solution: After setting the breakpoint, rerun the report ZABAPGIT_STANDALONE and repull the codes from the repository, it will goes to the screen of debug mode. Please clear the variable 'gtadir_exists' and then execute by clicking button, the BADI will be updated successfully.

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 the same as the name which is defined in the corresponding model in Manage Models application in GTT V2.

The screenshot illustrates the configuration of an Application Object Type (AOT) for a Sales Order Fulfillment process (SOF). On the left, the 'Define Application Object Types' dialog shows the 'Appl. Obj. Type' field set to 'ZGTT_SHP_ACC_HD', which is highlighted with a red box. This field corresponds to the 'Application Object Type' field in the 'Model Details' view on the right, also highlighted with a red box. Both fields contain the value 'ZGTT_SHP_ACC_HD'. The 'Model Details' view displays various tabs like General Data, Control Tables, Object Identification, Global Track & Trace Ref, IDOC Integration, and Fields. In the IDOC Integration tab, the tracked process is set to 'Shipment'. The Fields section lists several IDOC segments and their corresponding SAP fields, such as 'shipmentNo' and 'YN_SHP_NO'.

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCP	YN_SHP_NO
serviceAgentLbnId	E1EHPCP	YN_SHP_SA_LBN_ID
transportationMode	E1EHPCP	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPCP	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 extractorfunction:

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 ParameterExtractors(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

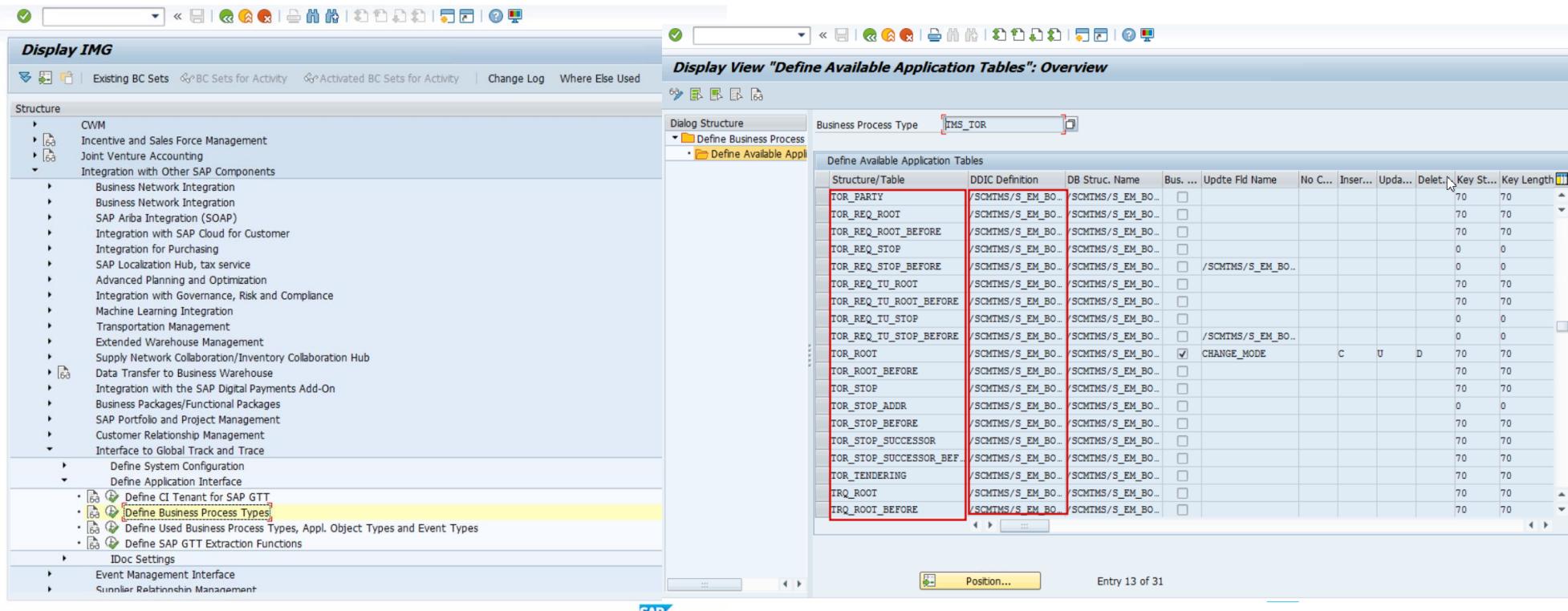
3: Sample Codes for Standard Shipment Tracking Application

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.

Left Screen (Display IMG): Shows the navigation path under 'Integration with Other SAP Components' to 'Interface to Global Track and Trace' and then 'Define Application Interface'. The 'Define Application Interface' node is highlighted with a yellow background and a red border around its icon.

Right Screen (Display View "Define Available Application Tables": Overview): Shows the configuration of application tables for the 'TMS_TOR' business process type. The table lists the following context tables:

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
TIQ_ROOT	/SCMTMS/S_EM_BO...	/SCMTMS/S_EM_BO...							70	70
TIQ_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*

The screenshot shows two SAP ABAP editors side-by-side. The left editor displays the source code for the function module *ZSST_GTT_OTE_FO_HDR_REL*. The right editor displays the source code for the include module *LZSST_GTT_D20*. Both editors have toolbars and status bars at the top and bottom respectively.

Function Builder: Display ZSST_GTT_OTE_FO_HDR_REL

ABAP Editor: Display Include LZSST_GTT_D20

```
Function Module: ZSST_GTT_OTE_FO_HDR_REL active
Include: LZSST_GTT_D20 Active

--> DATA: lt_app_objects TYPE trxas_appobj_ctabs,
          lo_udm_message TYPE REF TO cx_udm_message,
          ls_bapiret      TYPE bapiret2.

        lt_app_objects = VALUE #( ( i_app_object ) ).

      TRY.
        e_result = lcl_ef_performer->check_relevance(
          is_definition           = VALUE #( maintab = lif_sst ),
          io_bo_factory           = NEW lcl_tor_factory( ),
          iv_appsyst              = i_appsyst,
          is_app_obj_types         = i_app_obj_types,
          it_all_appl_tables      = i_all_appl_tables,
          it_app_objects           = lt_app_objects ).

      CATCH cx_udm_message INTO lo_udm_message.
        lcl_tools->get_errors_log(
          EXPORTING
            io_udm_message = lo_udm_message
            iv_appsyst    = i_appsyst
          IMPORTING
            es_bapiret    = ls_bapiret ).

        APPEND ls_bapiret TO c_logtable.
        CASE lo_udm_message->textid.
          WHEN lif_ef_constants=>cs_errors-stop_processing.

Scope: \FUNCTION ZSST_GTT_OTE_FO_HDR_REL\TRY
      ABAP
      Ln 35

METHOD lif_bo_reader-check_relevance.
  FIELD-SYMBOLS <ls_header> TYPE /scmtms/s_em_bo_tor_root.

  ASSIGN is_app_object-maintabref-* TO <ls_header>.
  IF sy-subrc <> 0.
    MESSAGE e010(zsst_gtt) INTO DATA(lv_dummy).
    lcl_tools->throw_exception( ).
  ENDIF.

  rv_result = lif_ef_constants->cs_condition-false.

  IF is_app_object-maintabdef = lif_sst_constants->cs_tabledef-fo_header_new AND
    (<ls_header>-track_exec_rel = lif_sst_constants->cs_track_exec_rel-execution OR
     <ls_header>-track_exec_rel = lif_sst_constants->cs_track_exec_rel-exec_with_extern_event_mngr ) AND
    <ls_header>-lifecycle = lif_sst_constants->cs_lifecycle_status-in_process AND
    (<ls_header>-execution = lif_sst_constants->cs_execution_status-in_execution OR
     <ls_header>-execution = lif_sst_constants->cs_execution_status-ready_for_transp_exec OR
     <ls_header>-execution = lif_sst_constants->cs_execution_status-executed ) AND
    <ls_header>-tspid IS NOT INITIAL AND
    (<ls_header>-tor_cat = /scmtms/if_tor_const->sc_tor_category-active OR
     <ls_header>-tor_cat = /scmtms/if_tor_const->sc_tor_category-booking ).

  CASE is_app_object-update_indicator.
    WHEN lif_ef_constants=>cs_change_mode-insert.
      rv_result = lif_ef_constants->cs_condition-true.
    WHEN lif_ef_constants=>cs_change_mode-update OR
        lif_ef_constants=>cs_change_mode-undefined.
      rv_result = lcl_tools->are_structures_different(
        ir_data1 = lif_bo_reader->get_data( is_app_object = is_app_object )
        ir_data2 = lif_bo_reader->get_data( is_app_object = is_app_object ) )
  ENDIF.
  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..ENDFI..* block. It exports *lcl_ef_performer->get_track_id_data* and imports *et_track_id_data* of type *e_trackiddata[]*. It also handles errors by catching *cx_udm_message* and logging them via *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_appsyst = i_appsyst
      i_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_trackiddata[]
  ).

  CATCH cx_udm_message INTO lo_udm_message.
  lcl_tools->get_errors_log( )
  EXPORTING
    io_udm_message = lo_udm_message
    iv_appsyst = i_appsyst
  IMPORTING
    es_bapiret = ls_bapiret .
  ENDTRY.

  " 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 takes *lr_item* and *lr_item_old* as parameters and returns *lt_track_id_data*. It also handles *FIELD-SYMBOLS* for *lt_item* and *lt_item_old*. The method uses *ASSIGN* statements to map parameters to local variables and then calls *lif_ef_reader->get_data* with specific parameters like *iv_tabledef* and *is_app_object-maintabref->**.

```
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,
                  <lt_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 outputtable *E_CONTROL_DATA*.
3. GTT v2 asks for full transport for all the control parameters, which means that all the fields needsto 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 mandatoryfields 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 modelfield 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 configuration interface. At the top, there are tabs for 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below these tabs, there is a section labeled 'Integration Switch' with a toggle switch set to 'ON'. The application object type is specified as 'ZGTT_SHP_ACC_HD'. A red box highlights a table titled 'Fields' under the 'Event to Action' tab, which maps model fields to IDOC segments and fields. The table data is as follows:

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.

```

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
46      CASE lo_udm_message->textid.

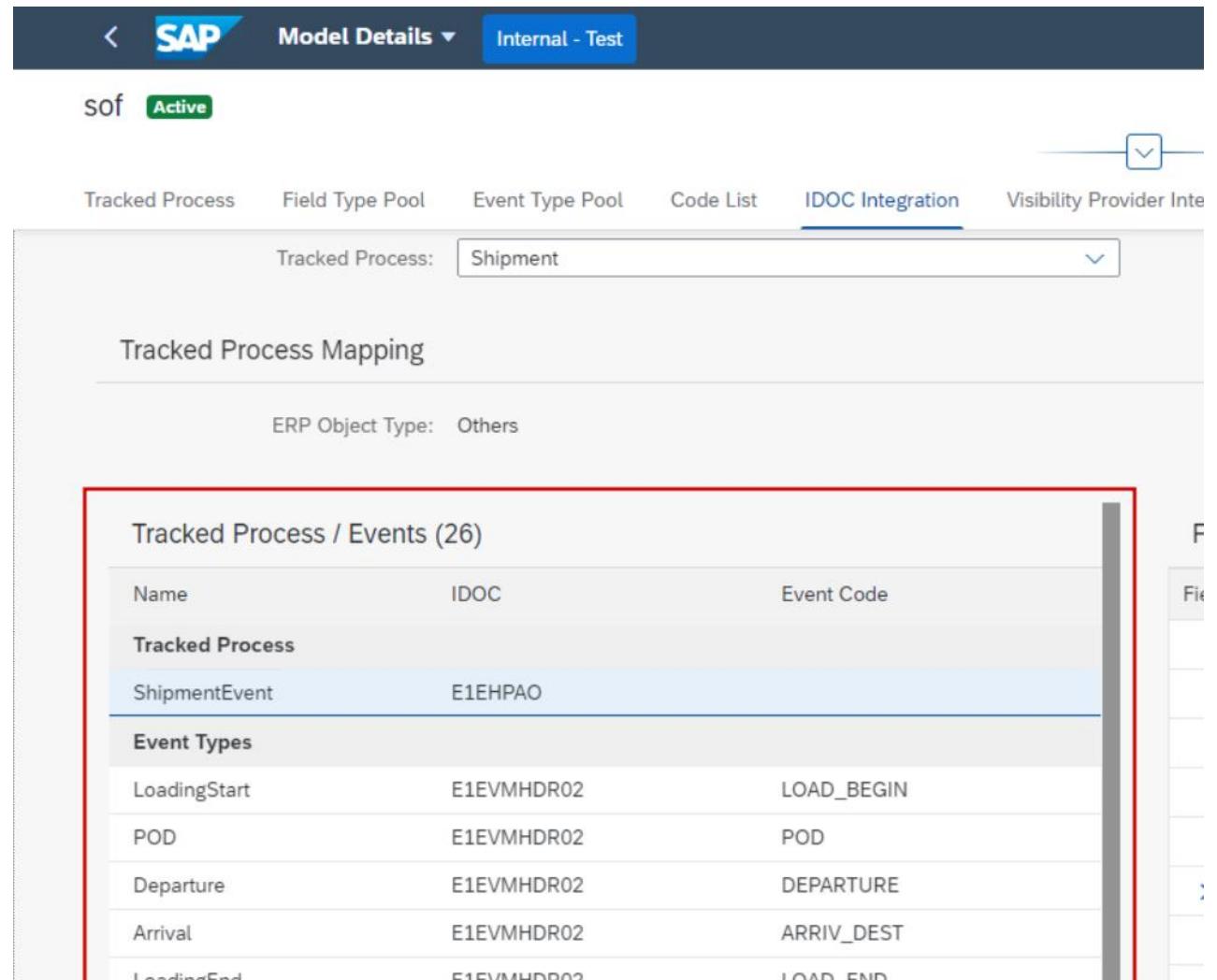
```

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_DATETIME* 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 GTTV2.
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 model named 'sof' (status: Active). The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process' is set to 'Shipment'. The 'Tracked Process / Events' section lists 26 entries, each with a Name, IDOC, and Event Code. The first entry is 'ShipmentEvent' with IDOC 'E1EHPAO'. Below this, there are sections for 'Event Types' listing 'LoadingStart', 'POD', 'Departure', 'Arrival', and 'LoadingEnd' with their respective event codes: 'LOAD_BEGIN', 'POD', 'DEPARTURE', 'ARRIV_DEST', and 'LOAD_END'.

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 LZZST_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_appsyst         = i_appsyst
      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_appsyst     = i_appsyst
      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 for a tracked process named 'Shipment'. The 'Tracked Process' tab is selected, showing a mapping table for 'ERP Object Type: Others'. A red box highlights the 'Tracked Process / Events (26)' section, which lists various event types and their corresponding IDOC codes and event codes. The table structure is 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

9: Coding Tips in the Event Data Function Modules

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL

Repository Browser

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

SAP

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL_REL

Repository Browser

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      TYPE /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
```

SAP

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[®]