



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

SAP Business Network
February 2021

PUBLIC

Objectives



After completing this learning module, you will be able to:

- Learn what prerequisites are necessary for the global track and trace option
- Learn how to maintain IDOC configurations in ERP for integration
- Learn how to maintain extractors in ERP for integration
- Learn how to download and implement sample ABAP codes from Github
- Learn how to customize the logic based on sample codes

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 ERP for GTT Version 2 shall be running on NETWEAVER 7.31 or higher

1-2: SAP NOTE 2937175 shall be implemented

1-3: The ABAP codes to support sample apps for GTT Version 2 shall be implemented in S4 HANA 1909 SP03 on premise or higher, which is not validated in lower release

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>

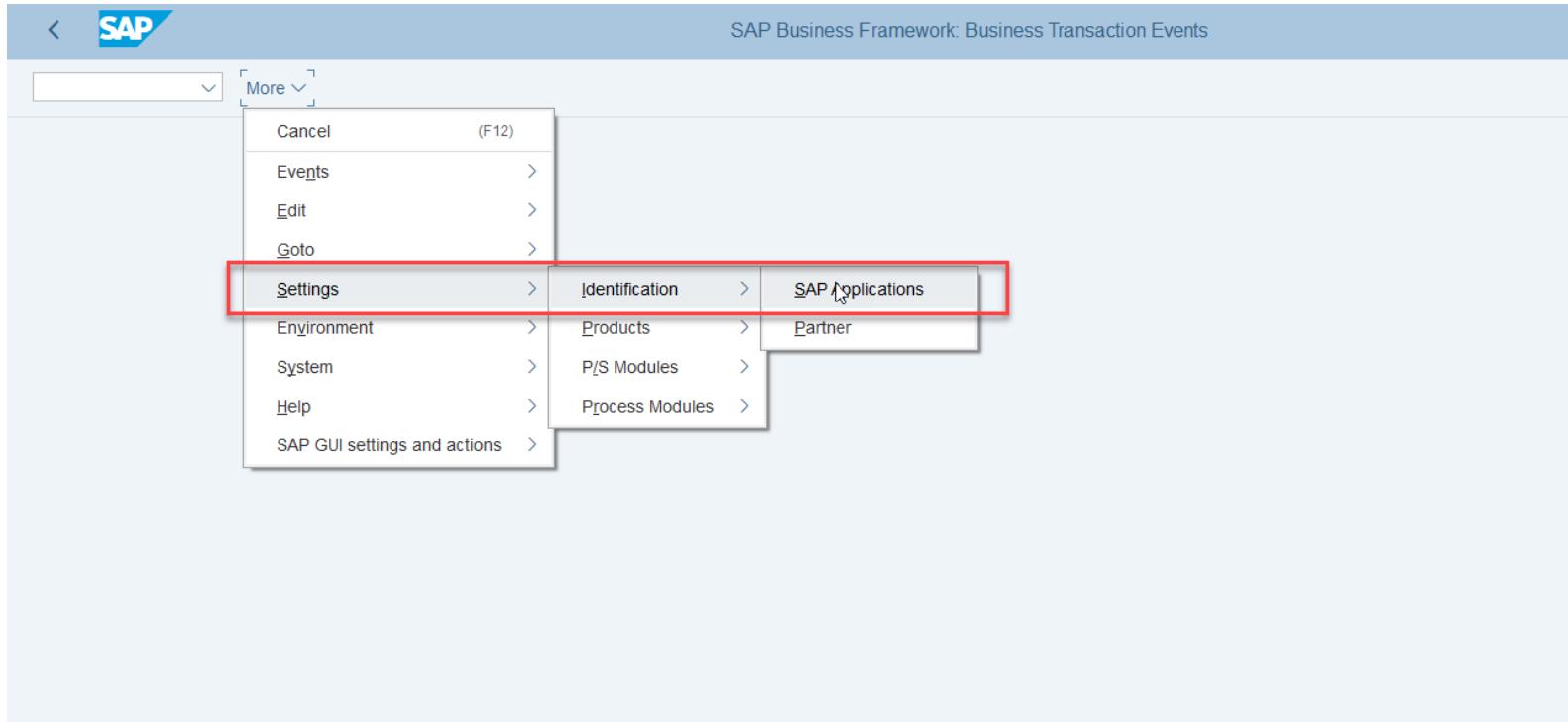
SAPNotes						
SAP Component	Number	Versi...	Score	Title	Changed On	Status
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer
SCM-EM-AS	2634395	1	1	Solving ATC Issues	27.09.2019	Released for Customer
SCM-EM-AS	2819787	1	1	TM-EM integration - analyzing errors	25.07.2019	In Process
SCM-EM-AS-CNF	2798670	1	1	IMG activity inactive: Define SAP EM Extraction Functions	29.05.2019	Released for Customer
SCM-EM-AS	2609449	4	1	Delete orphaned entries in table /SAPTRX/AOTREF (2)	11.07.2019	Pilot Release
SCM-EM-AS	2502086	2	1	Aligning the BAPI processing mode with the communication mode	11.07.2017	Pilot Release
SCM-EM-AS	2339984	2	1	Orphaned EM inbound queues in application systems	18.04.2019	Released for Customer
SCM-EM-AS	2159436	1	1	Runtime-Error "ABAP Programming" when trying to save delivery. System QSC-800	22.04.2015	In Process
SCM-EM-AS	1507998	4	1	Expert Consulting in the area of SAP Event Management	09.05.2011	Released for Customer
IS-R-PUR-PCC	896191	3	1	FAQ: EM seasonal procurement (Consulting, Tips, Customizing)	13.07.2006	Released for Customer

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**

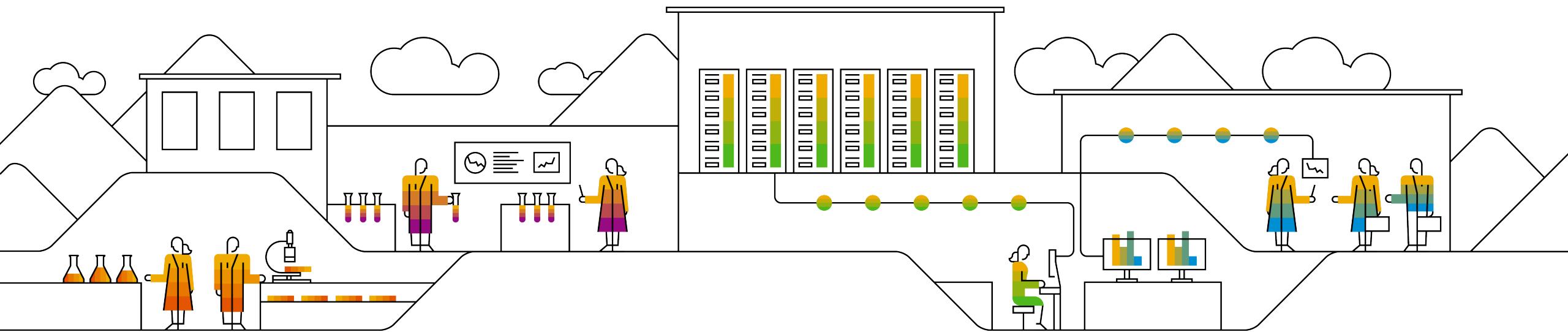
The screenshot shows a SAP application interface titled "Change View 'BTE Application Indicator': Overview". The main area is a table with two columns: "Appl." and "Text". The "Appl." column lists various application codes, and the "Text" column provides a brief description of each. The row for "PI-EM" is highlighted with a red box around its "Appl." value. A checkmark is present in the checkbox column next to "PI-EM", indicating it is active. Other applications listed include PM, PM-BW, PM-EQM, PM-PAM, PM-PC, PMAT, PMIPUR, PMPUSH, PP-BD, PP-DD, PP-MRP, PRICAT, PS-REP, PSRV, QBEXT, QBEXTP, QILPO, RDSVFI, and RDSVMD. The bottom right corner of the interface has "Save" and "Cancel" buttons.

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
PM-PC	<input checked="" type="checkbox"/>	Product Compliance
PMAT	<input checked="" type="checkbox"/>	Produkt - Material
PMIPUR	<input type="checkbox"/>	PMI Anschluss Einkauf
PMPUSH	<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

1-1: Log on to the business client

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

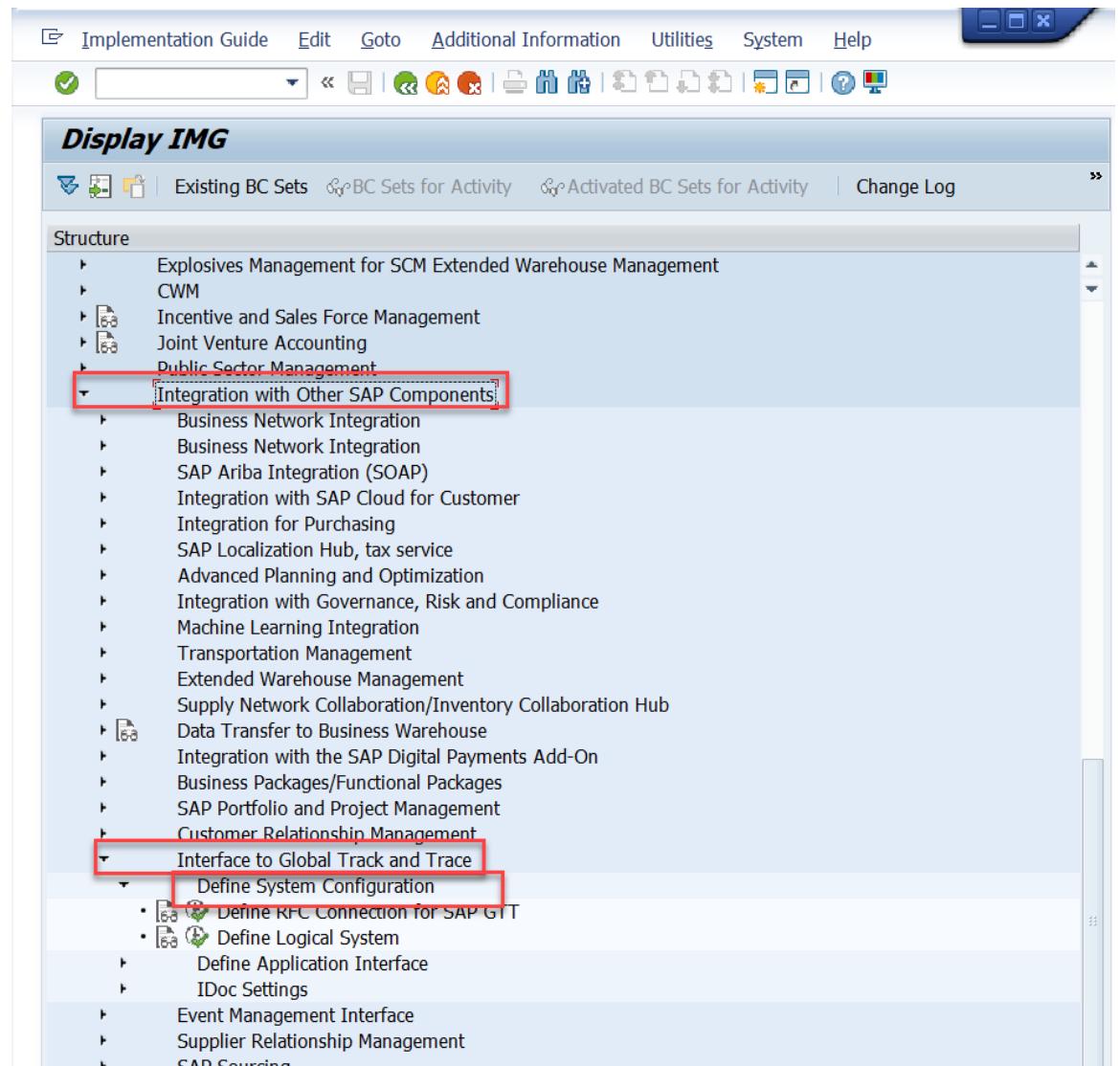
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

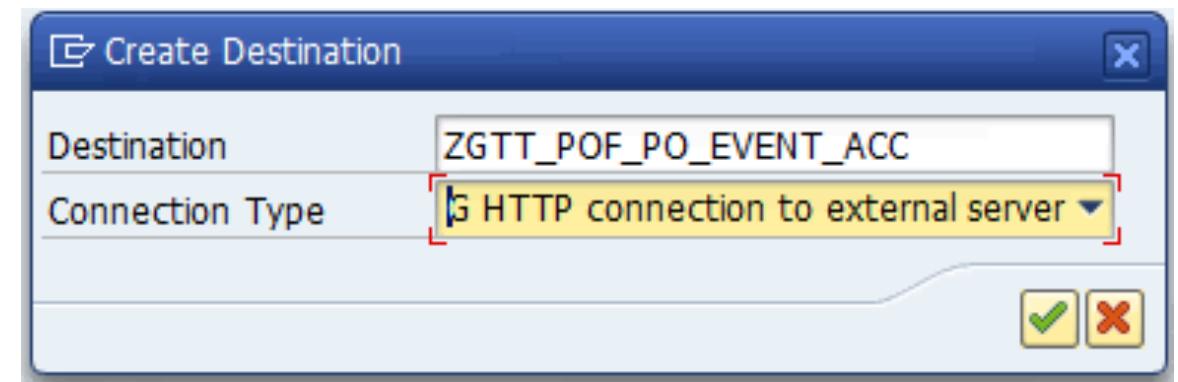
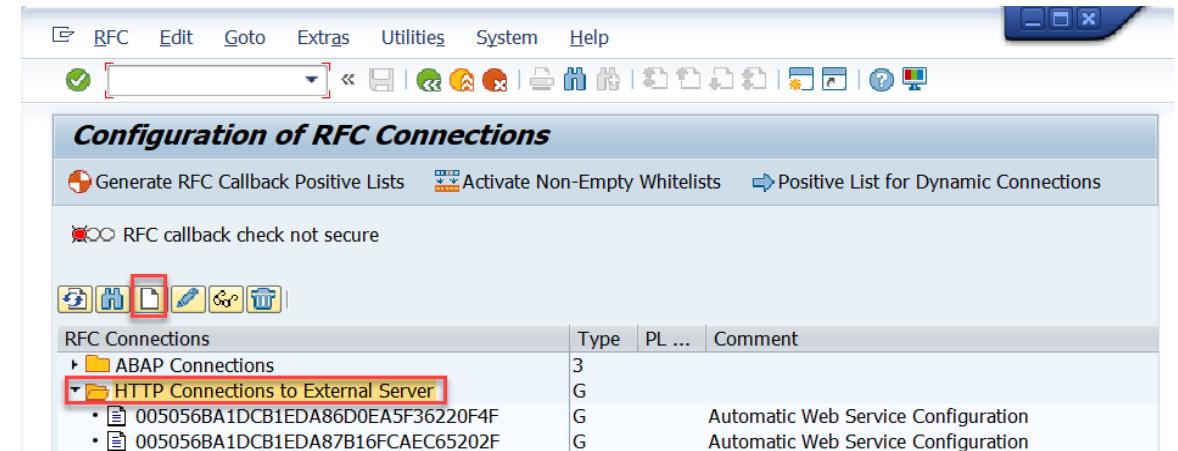
Define RFC Connection for SAP GTT



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 follows:

<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

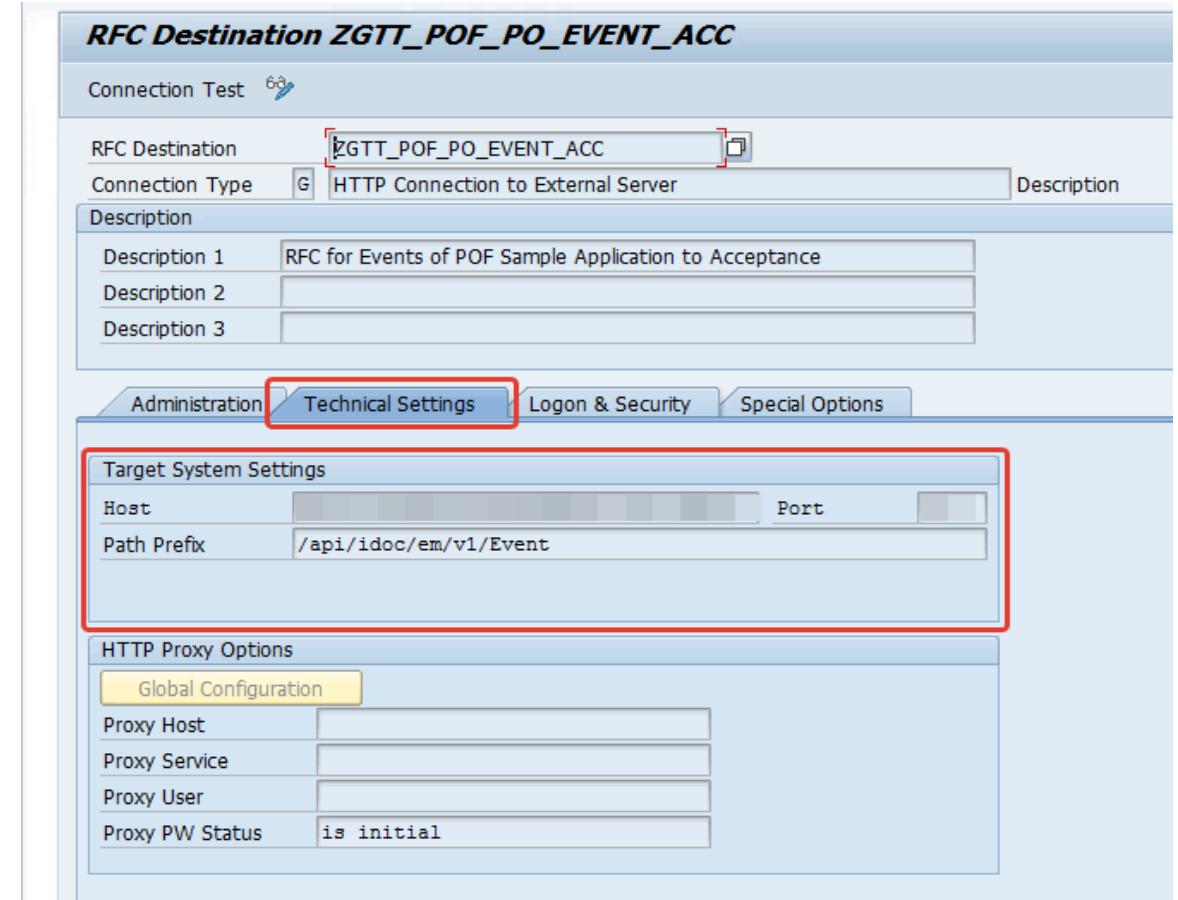
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

For the event:

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

For the tracked Process:

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



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*.

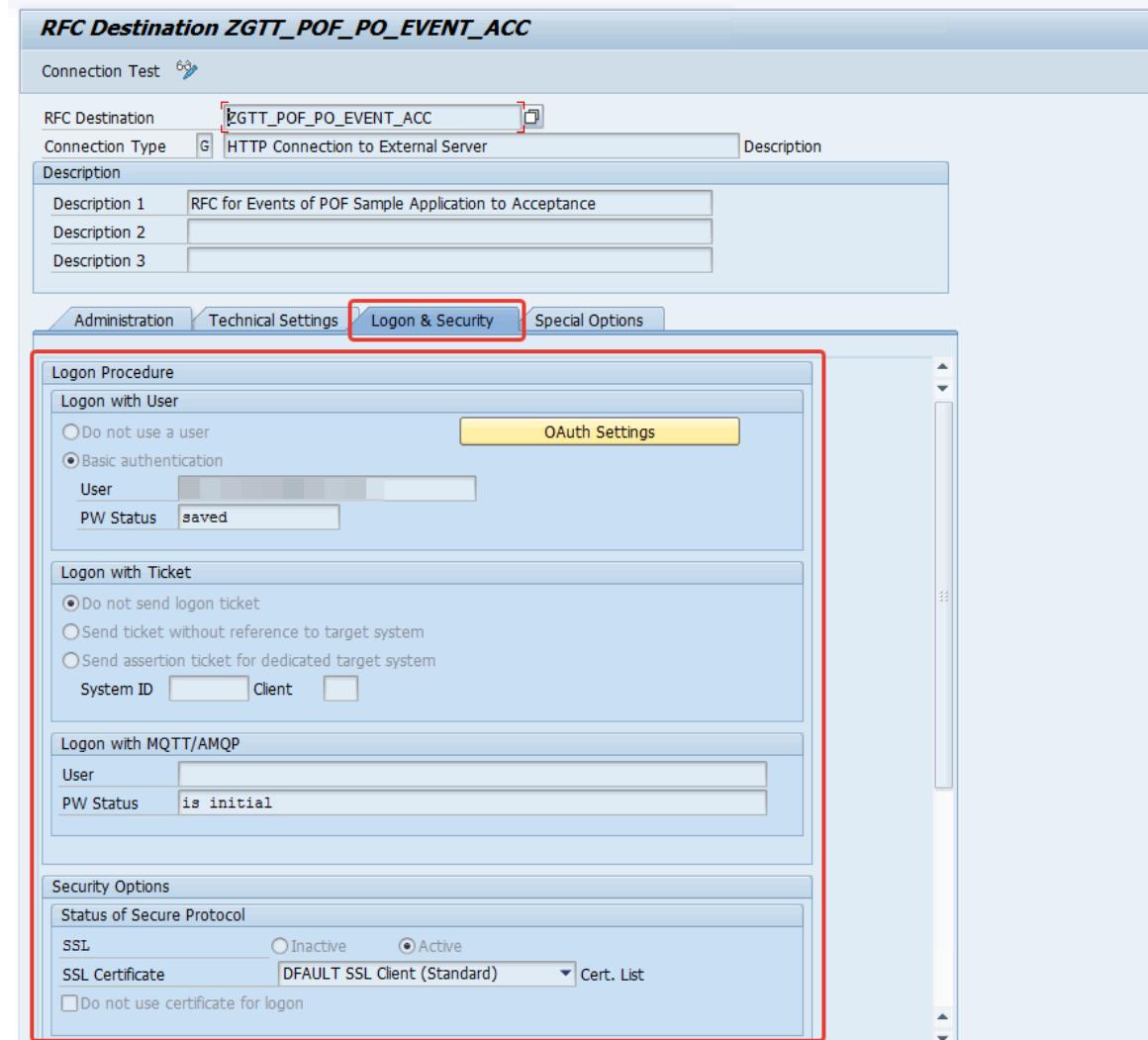
The recommended SSL Certificate is: *DEFAULT SSL Client (Standard)*.

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:

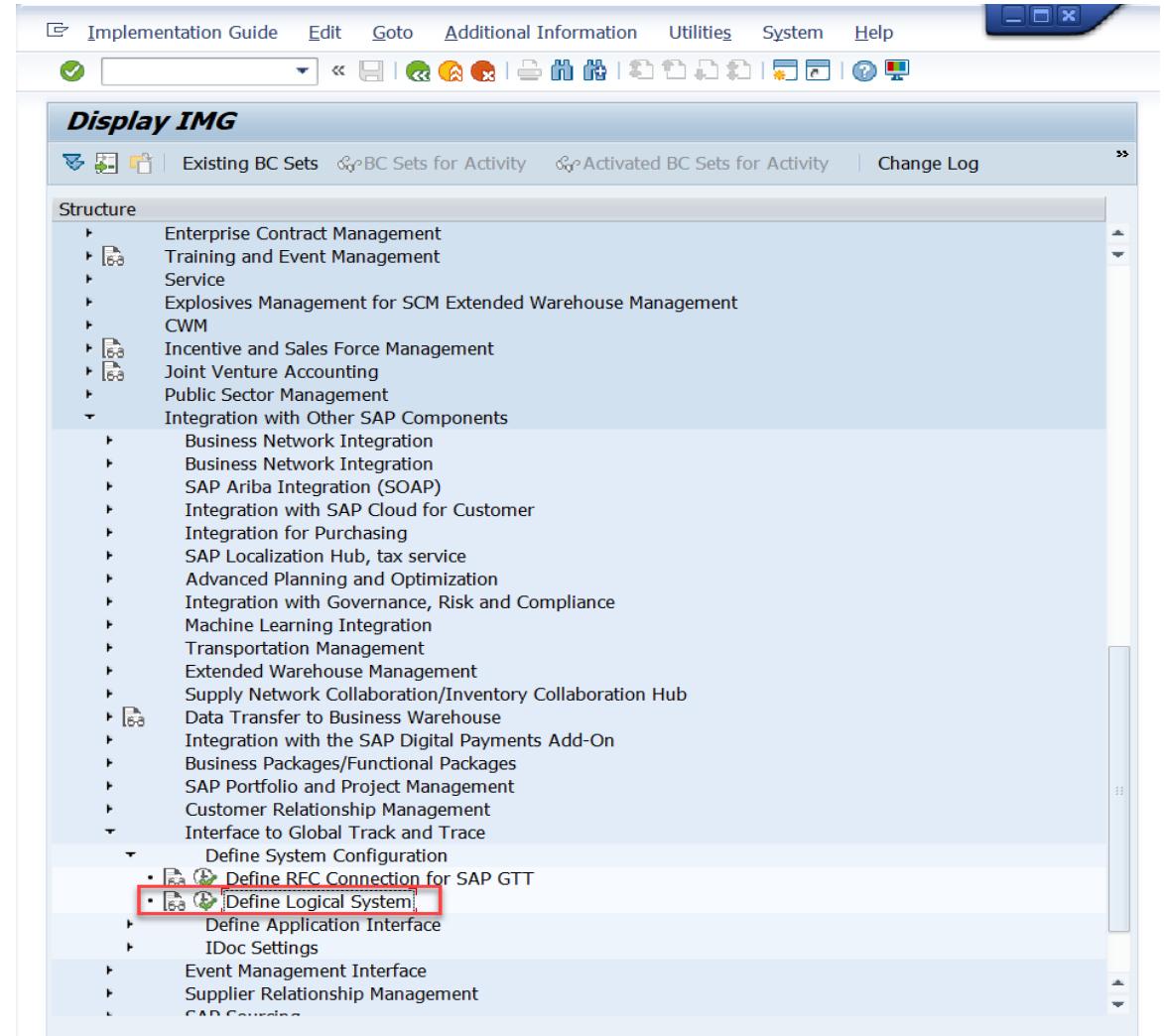
- one for event and
- the other for tracked process.



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**

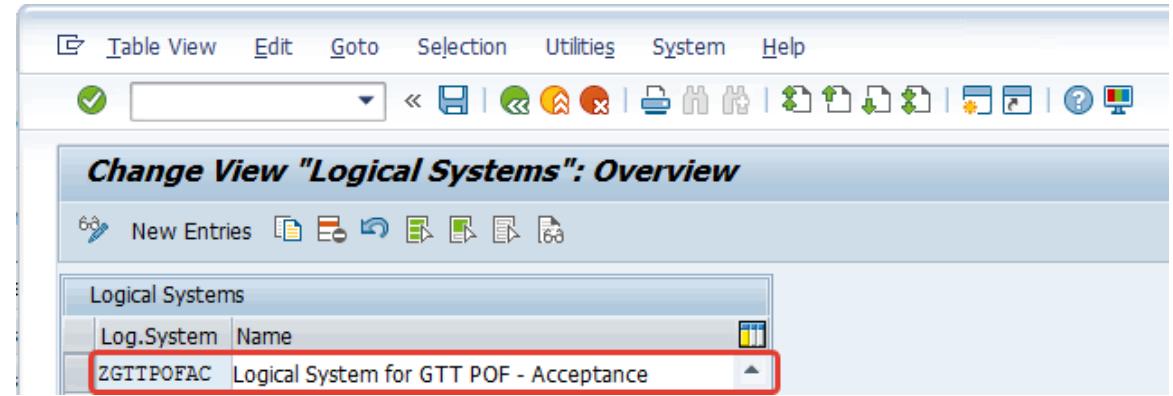


STEP 2: Define Logical System

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



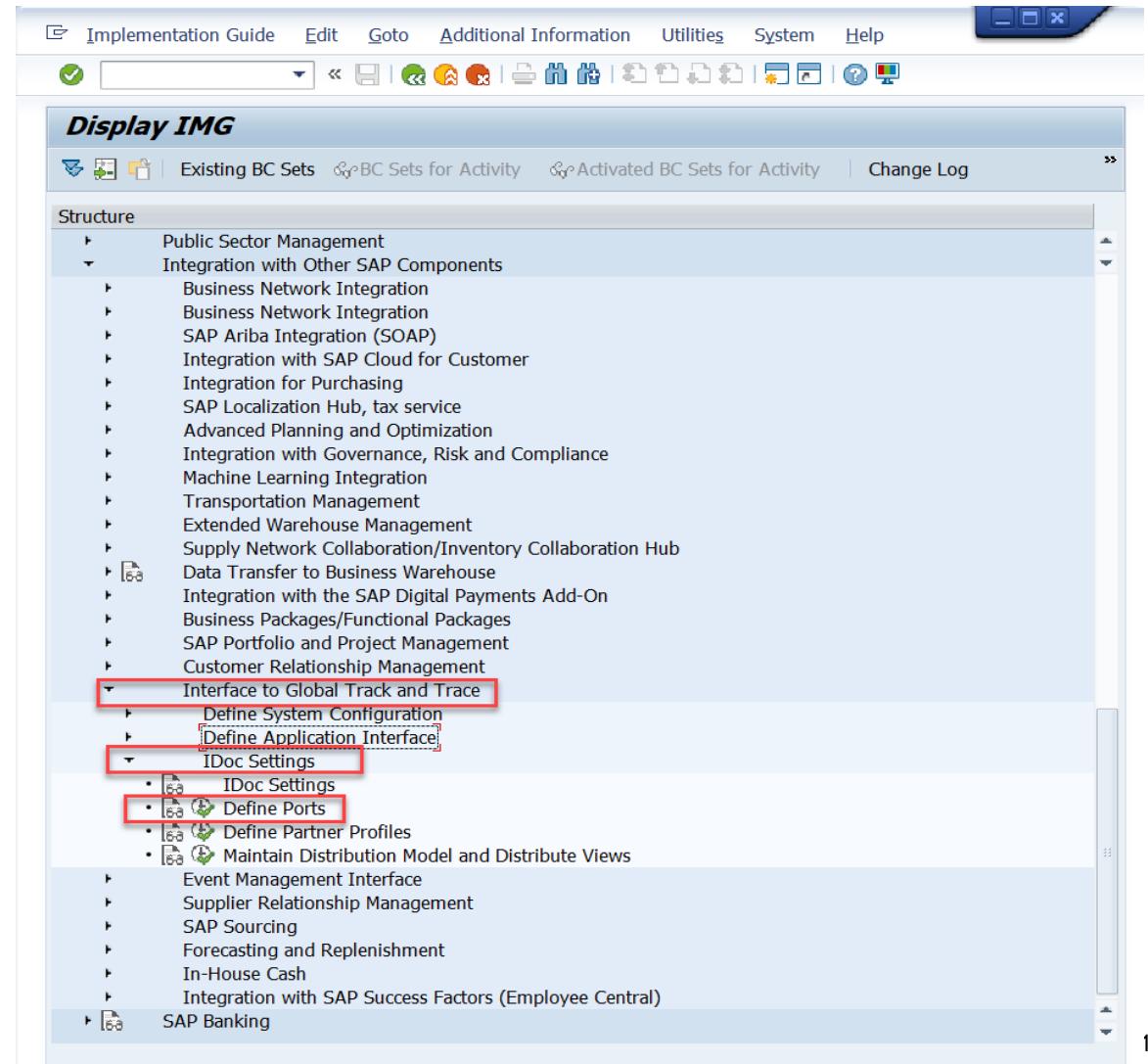
The screenshot shows the SAP Fiori interface for managing logical systems. The title bar reads "Change View 'Logical Systems': Overview". Below the title bar is a toolbar with various icons for navigation and actions. The main area is a table titled "Logical Systems" with two columns: "Log.System" and "Name". A single row is visible, containing "ZGTTPOFAC" in the Log.System column and "Logical System for GTT POF - Acceptance" in the Name column. This row is highlighted with a red border.

Log.System	Name
ZGTTPOFAC	Logical System for GTT POF - 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

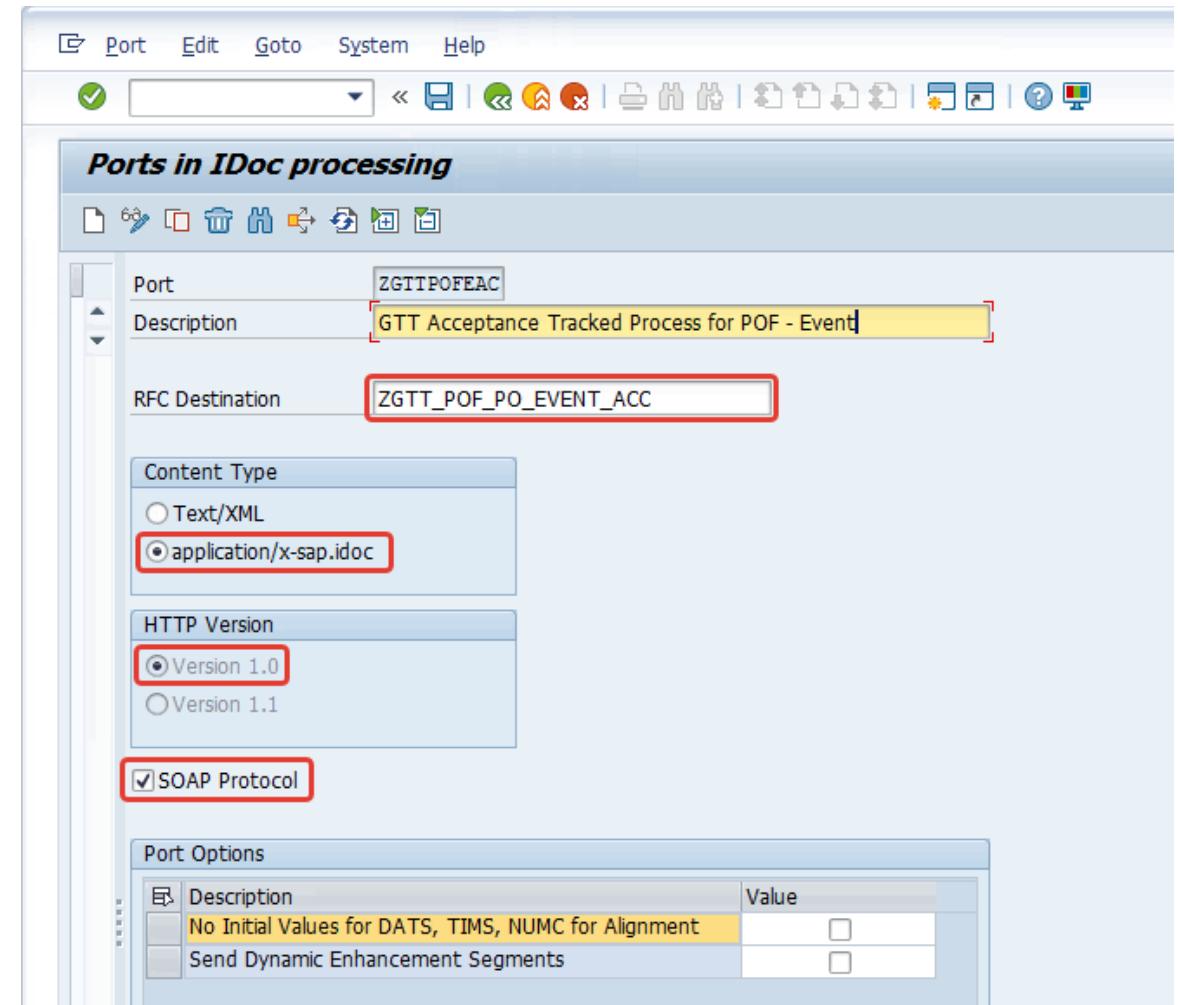
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

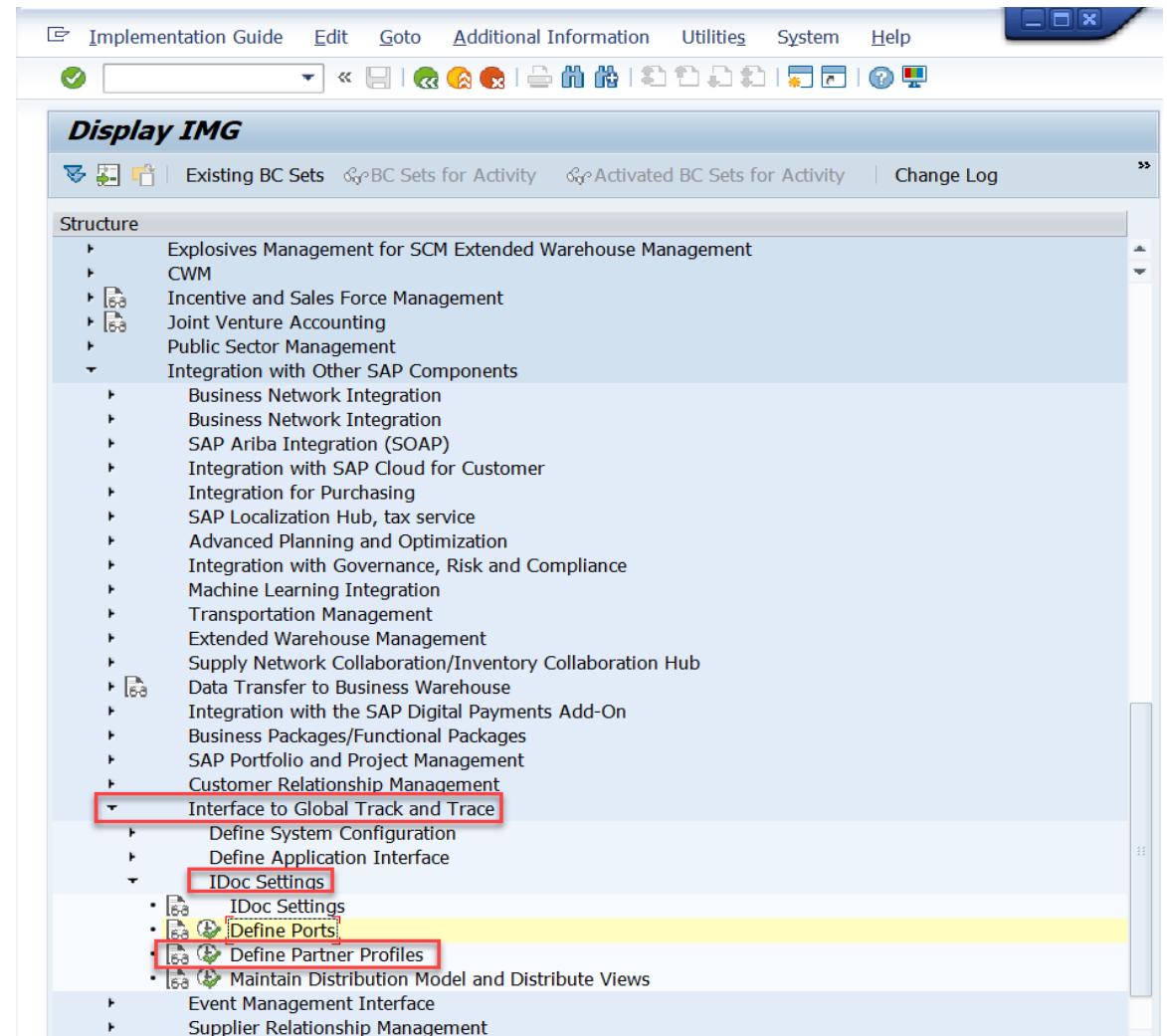
Caution: You need to define two ports, one for event and the other for tracked process.



STEP 4: Define Partner Profiles

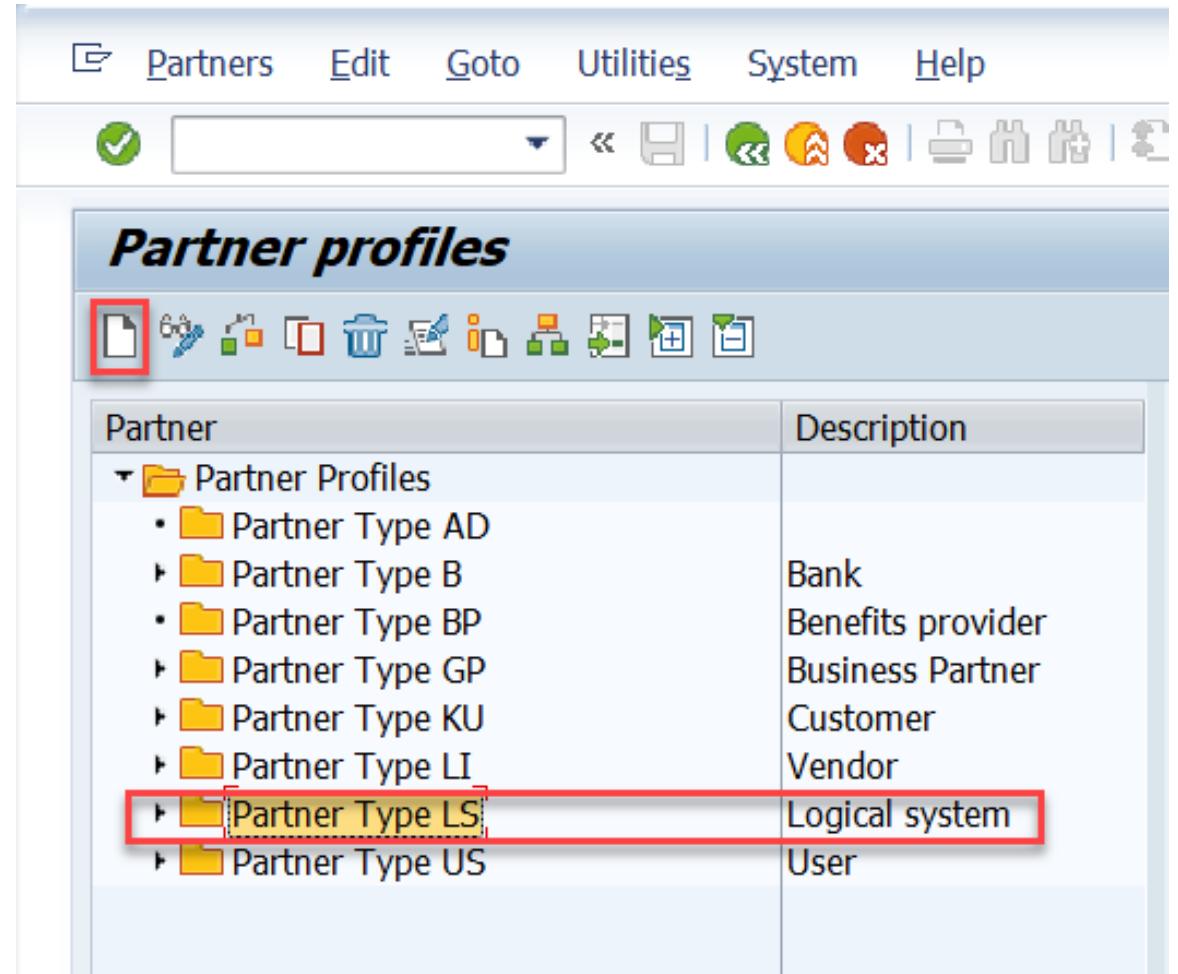
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

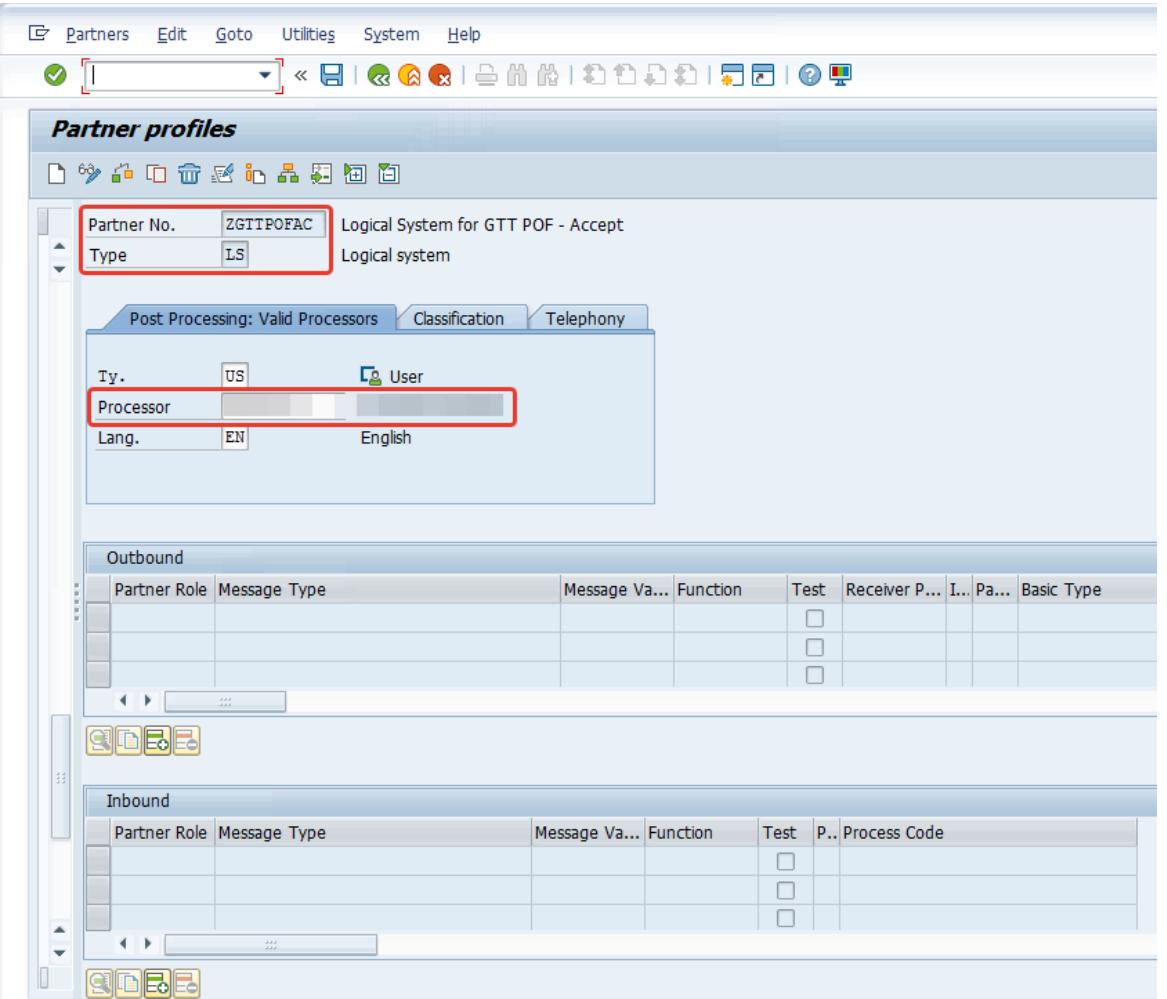
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

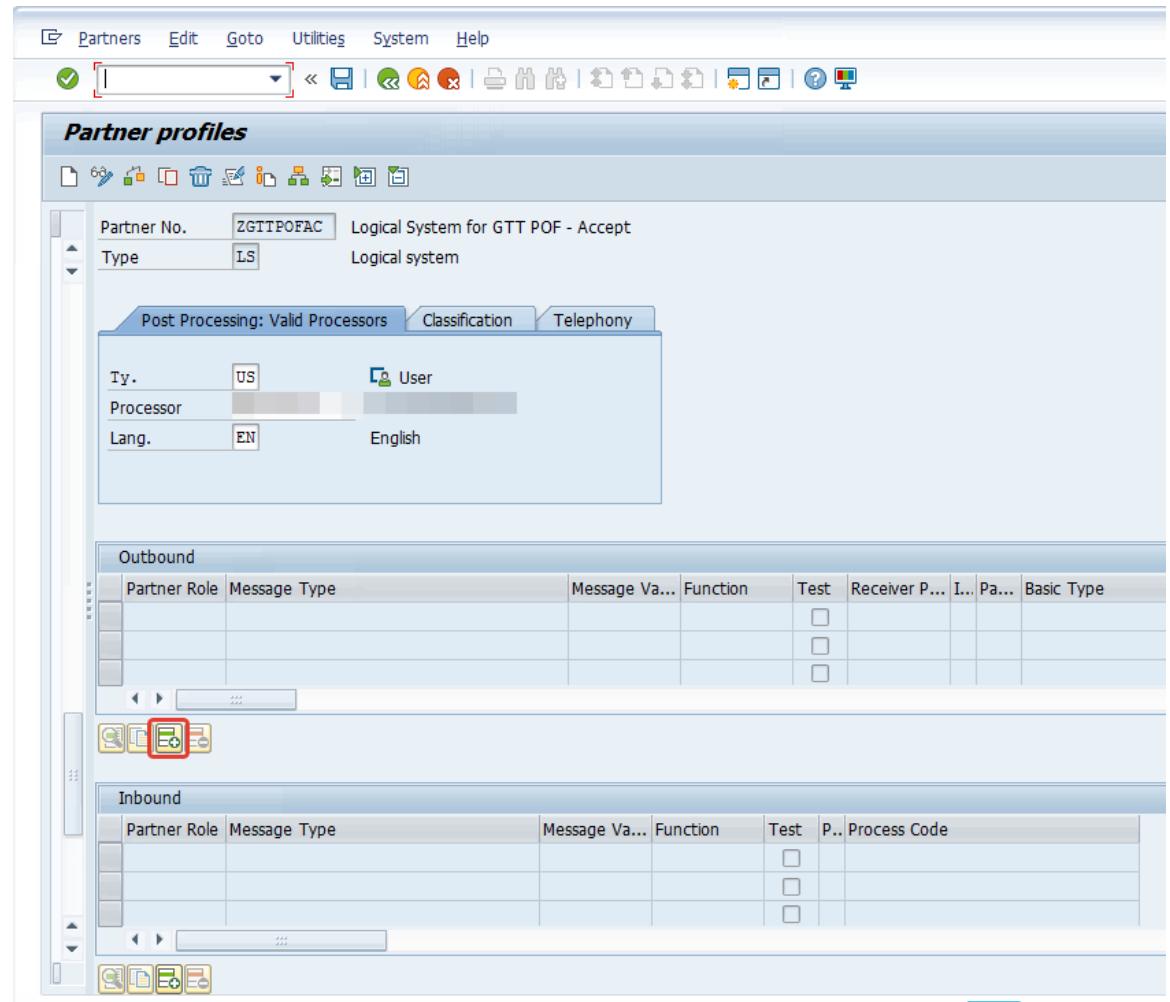
4-4: Fill in the **Partner No.** that you created in STEP 2

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



STEP 4: Define Partner Profiles

4-6: Click **Add** under **Outbound** box to create a new outbound parameter



STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

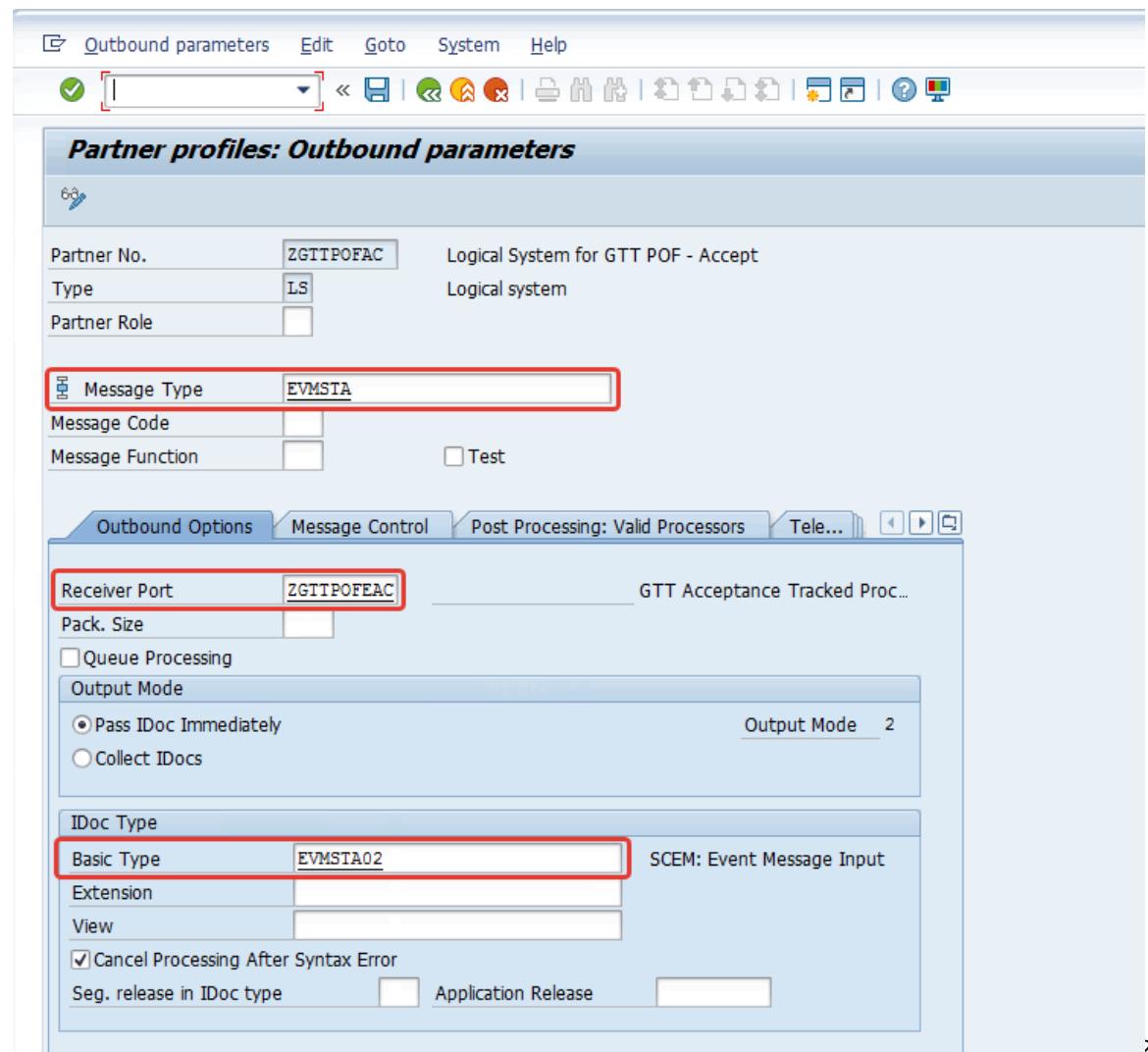
For the tracked Process:

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 4-6 to 4-9 to add two outbound parameters, one for event and the other for tracked process.



B) Configuration and Implementation

- Basic

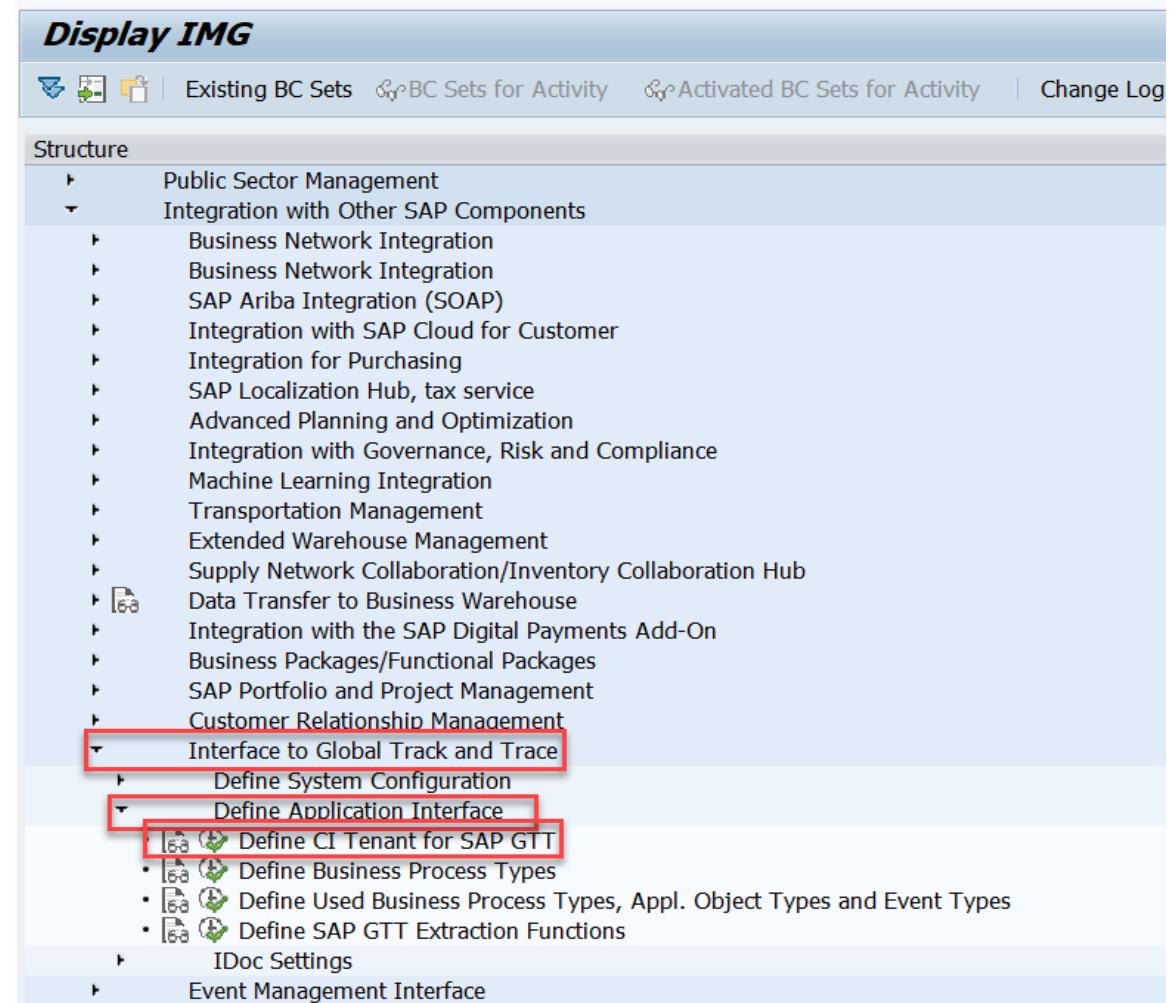
B2. Extractor Configuration



STEP 5: Define CI Tenant for GTT

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

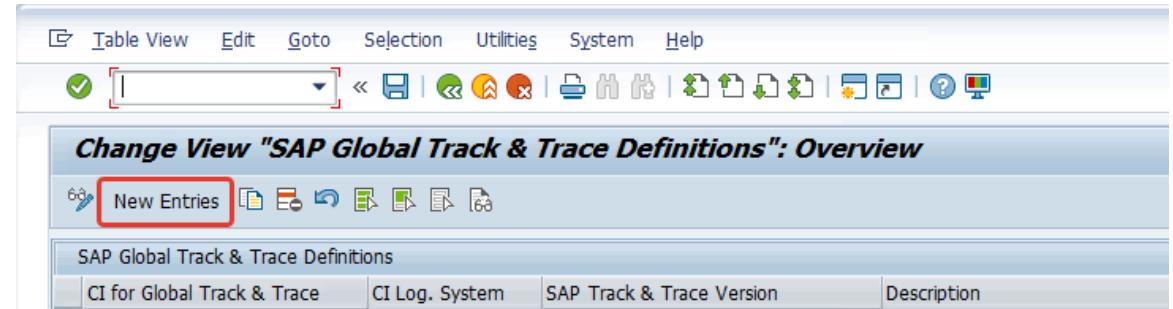
5-2: Choose activity
Define CI Tenant for SAP GTT



STEP 5: Define CI Tenant for GTT

5-3: Click **New Entries** to create a new CI tenant for GTT

5-4: Fill in the information for the new CI tenant. The **CI Log. System** is the logical system you created in STEP 2.



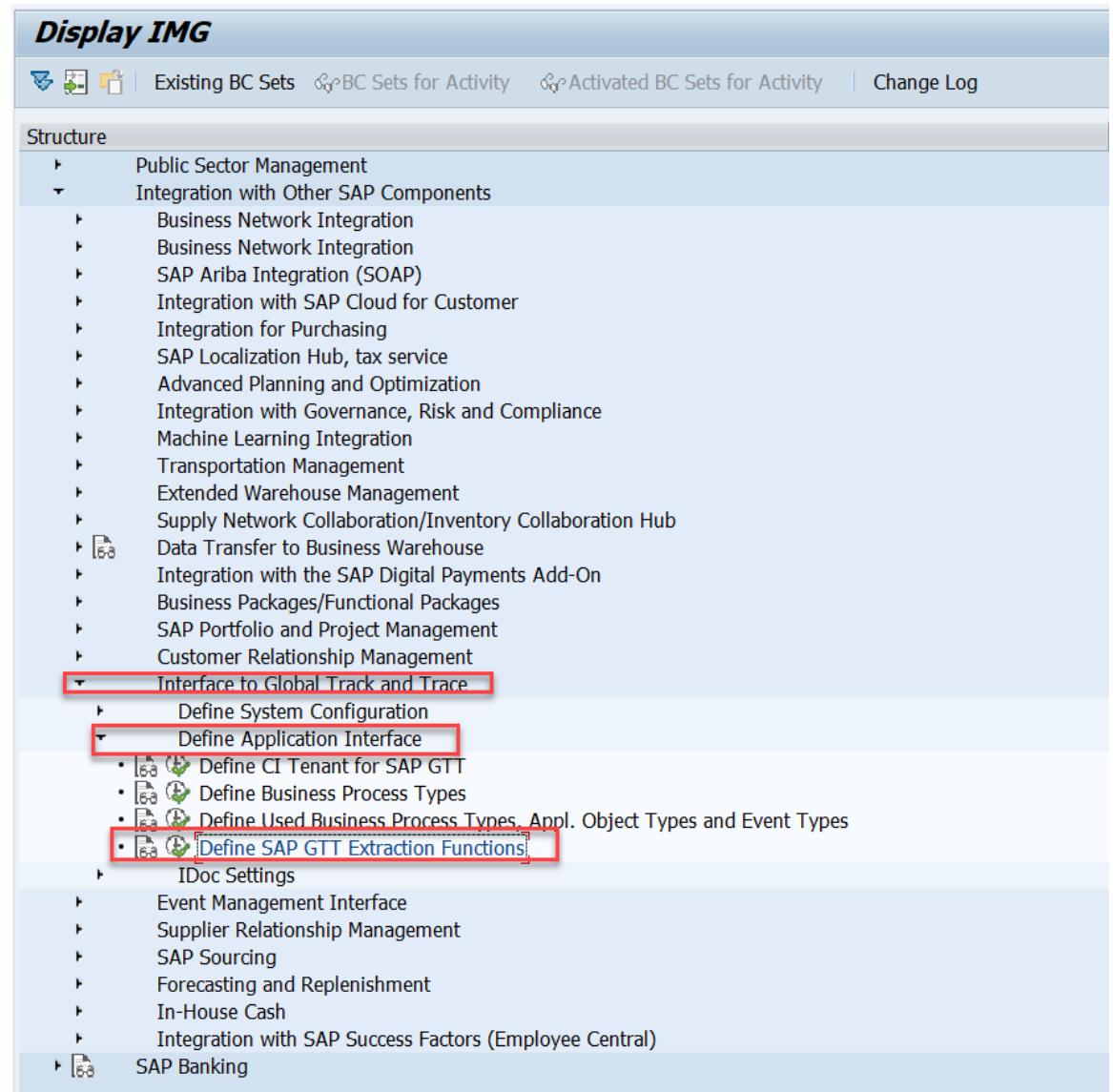
The screenshot shows the SAP Global Track & Trace Definitions overview screen. The toolbar and search bar are identical to the previous screenshot. The main title is 'Change View "SAP Global Track & Trace Definitions": Overview'. The 'New Entries' button in the toolbar is visible. The main area displays a table titled 'SAP Global Track & Trace Definitions' with columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. A single row is visible, which is highlighted with a red box: ZGTTPOFAC, ZGTTPOFAC, GTT1.0 Global Track & Trace, and CI For GTT Purchasing Order Sample APP - Acceptance.

CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTPOFAC	ZGTTPOFAC	GTT1.0 Global Track & Trace	CI For GTT Purchasing Order Sample APP - Acceptance

STEP 6: Define GTT Extraction Functions

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 SAP GTT Extraction Functions

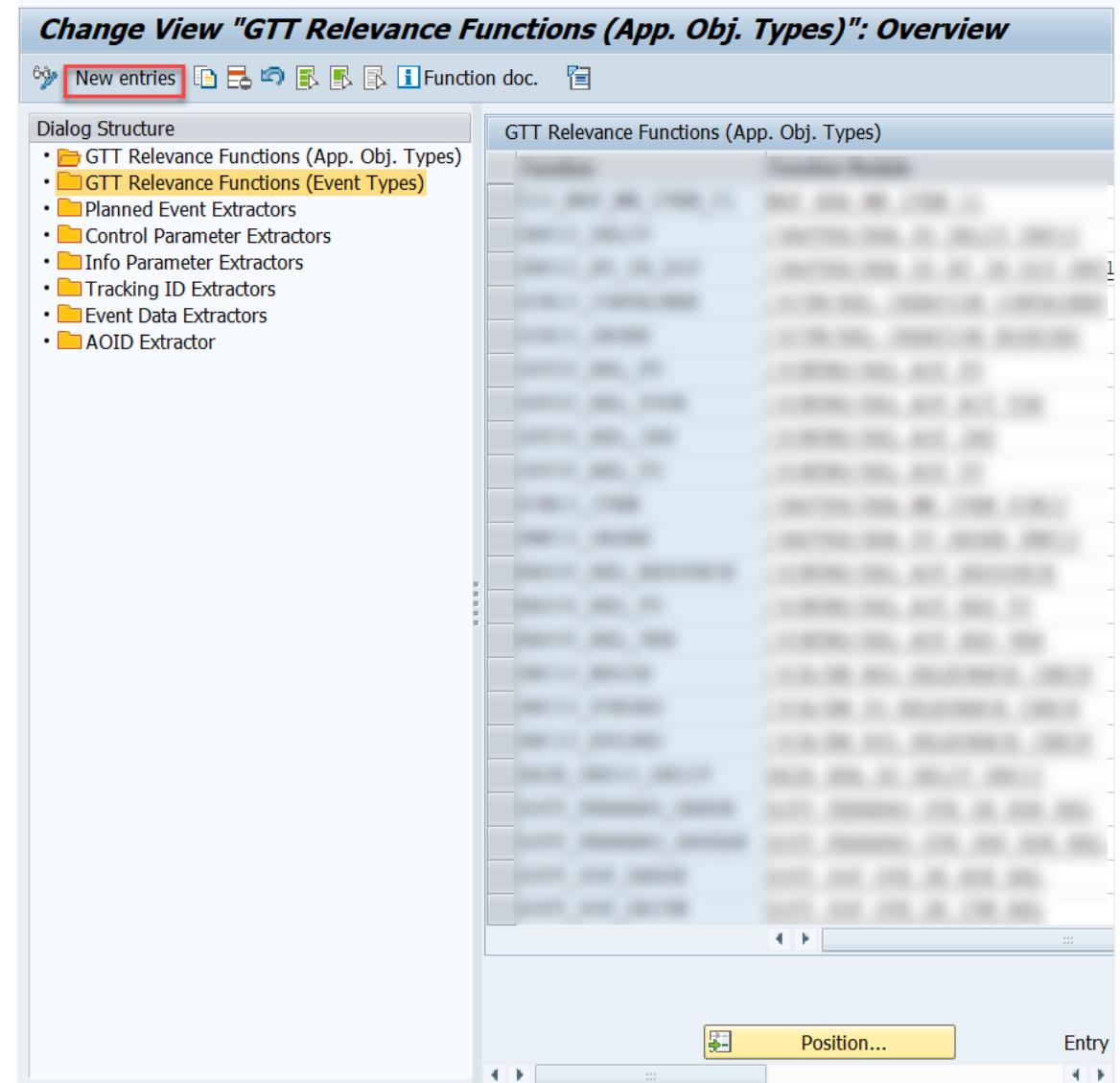


The screenshot shows the SAP Display IMG interface. The title bar reads "Display IMG". Below it is a toolbar with icons for search, refresh, and file operations, followed by links for "Existing BC Sets", "BC Sets for Activity", "Activated BC Sets for Activity", and "Change Log". The main area is titled "Structure" and contains a hierarchical list of SAP components. A red box highlights the path: "Integration with Other SAP Components" -> "Interface to Global Track and Trace" -> "Define Application Interface" -> "Define SAP GTT Extraction Functions!". The final step, "Define SAP GTT Extraction Functions!", is also highlighted with a red box.

- Public Sector Management
- Integration with Other SAP Components
 - Business Network Integration
 - Business Network Integration
 - SAP Ariba Integration (SOAP)
 - Integration with SAP Cloud for Customer
 - Integration for Purchasing
 - SAP Localization Hub, tax service
 - Advanced Planning and Optimization
 - Integration with Governance, Risk and Compliance
 - Machine Learning Integration
 - Transportation Management
 - Extended Warehouse Management
 - Supply Network Collaboration/Inventory Collaboration Hub
 - Data Transfer to Business Warehouse
 - Integration with the SAP Digital Payments Add-On
 - Business Packages/Functional Packages
 - SAP Portfolio and Project Management
 - Customer Relationship Management
- Interface to Global Track and Trace
 - Define System Configuration
 - Define Application Interface
 - Define CI Tenant for SAP GTT
 - Define Business Process Types
 - Define Used Business Process Types, Appl. Object Types and Event Types
 - Define SAP GTT Extraction Functions!
 - IDoc Settings
 - Event Management Interface
 - Supplier Relationship Management
 - SAP Sourcing
 - Forecasting and Replenishment
 - In-House Cash
 - Integration with SAP Success Factors (Employee Central)
- SAP Banking

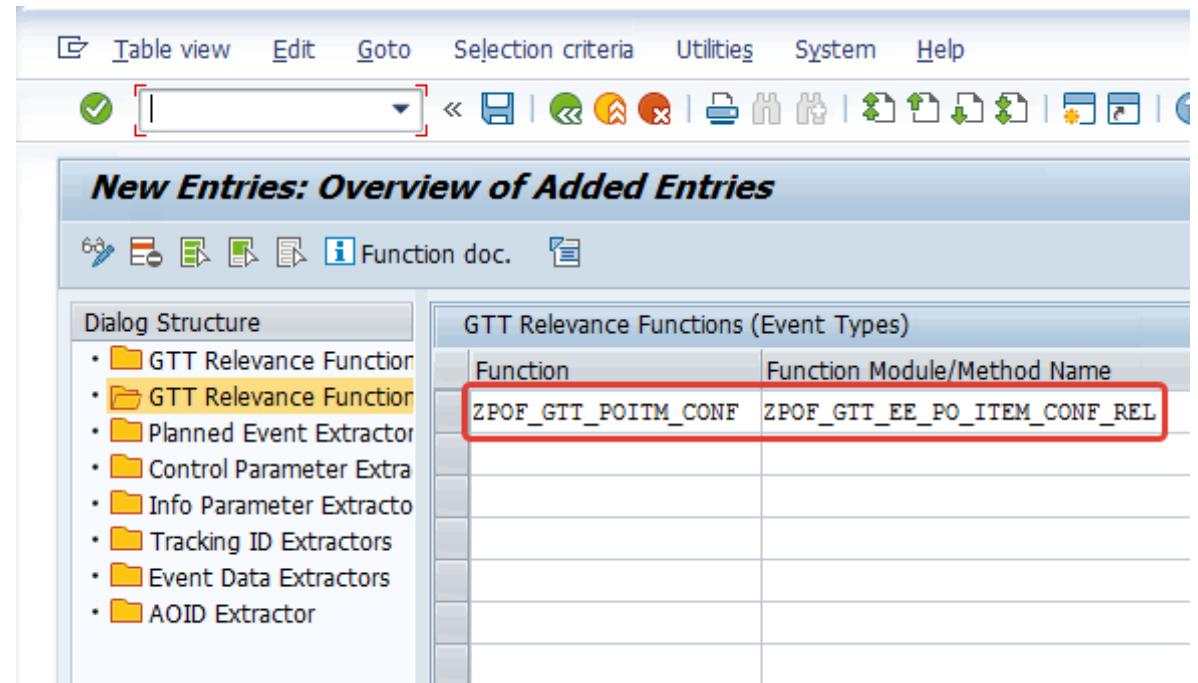
STEP 6: Define GTT Extraction Functions

6-3: Choose the type of Extraction Function you want to create from the **Dialog Structure**, and click **New entries**



STEP 6: Define GTT Extraction Functions

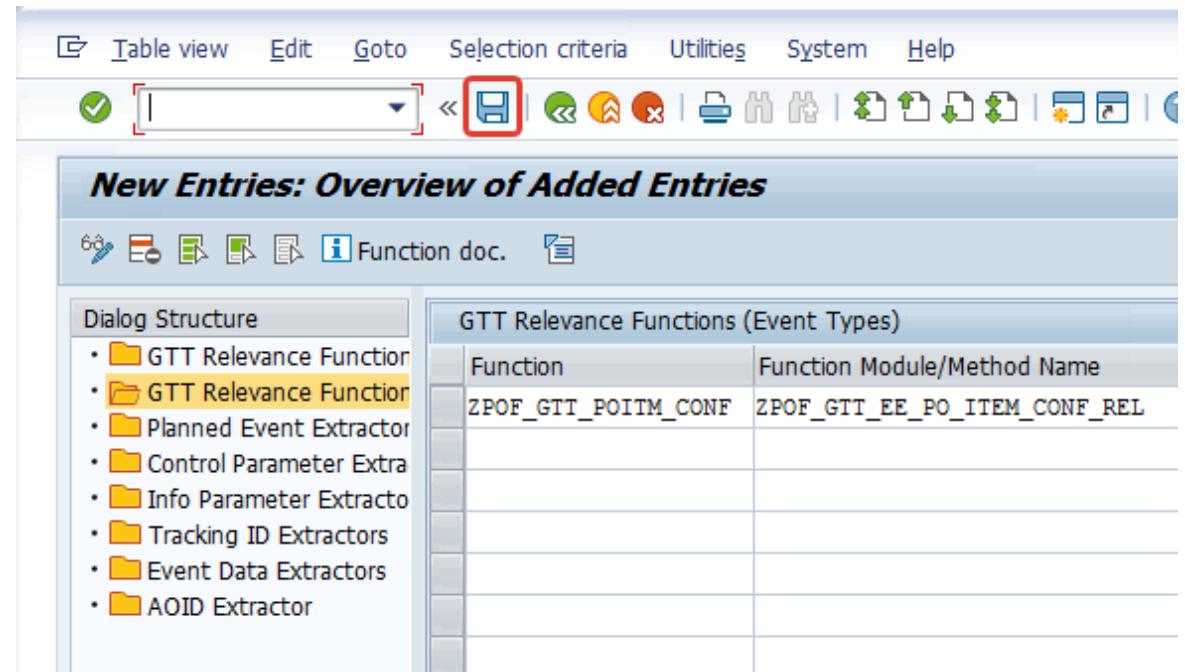
6-4: Input the **Function name** and **Function Module** for the newly created extraction function



GTT Relevance Functions (Event Types)	
Function	Function Module/Method Name
ZPOF_GTT_POITM_CONF	ZPOF_GTT_EE_PO_ITEM_CONF_REL

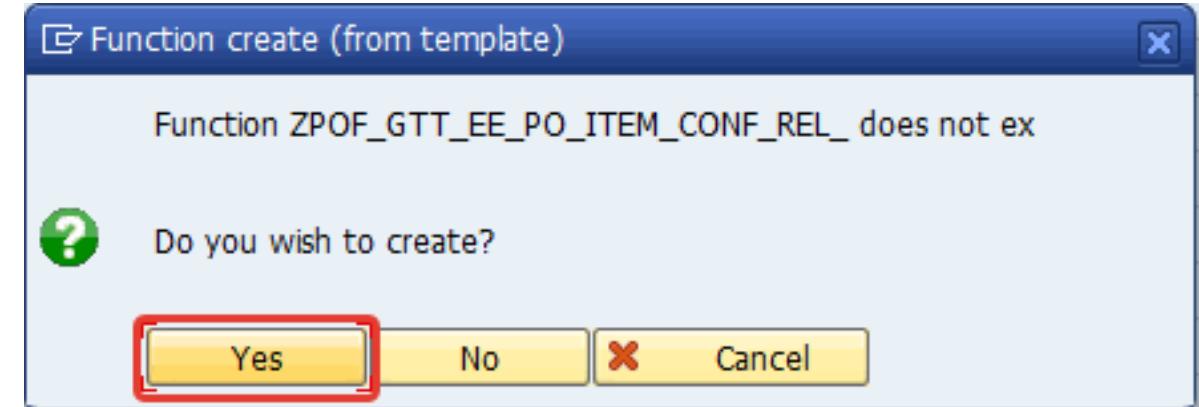
STEP 6: Define GTT Extraction Functions

6-5: Click **Save**



STEP 6: Define GTT Extraction Functions

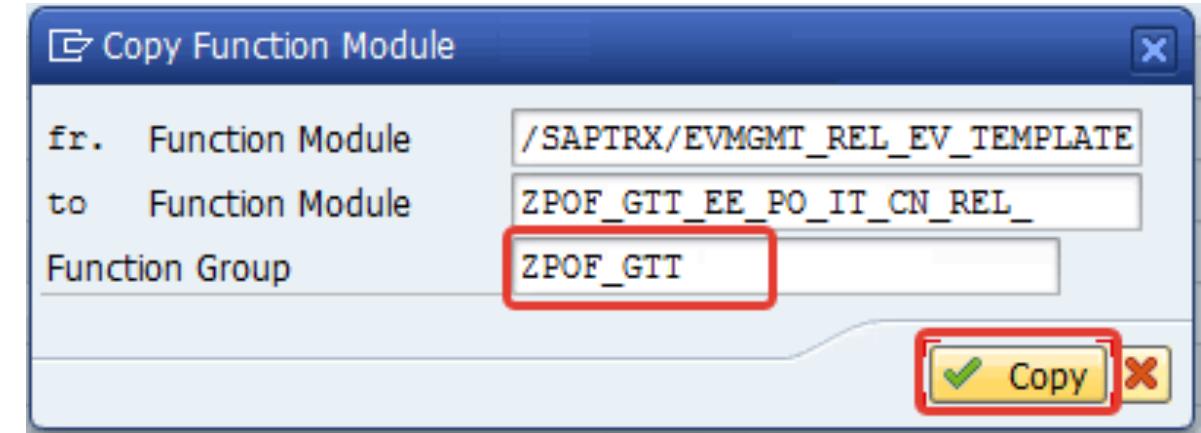
6-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



STEP 6: Define GTT Extraction Functions

6-7: Input the **Function Group** where the function module is to be created

6-8: Click **Copy**



STEP 6: Define GTT Extraction Functions

6-9: 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.

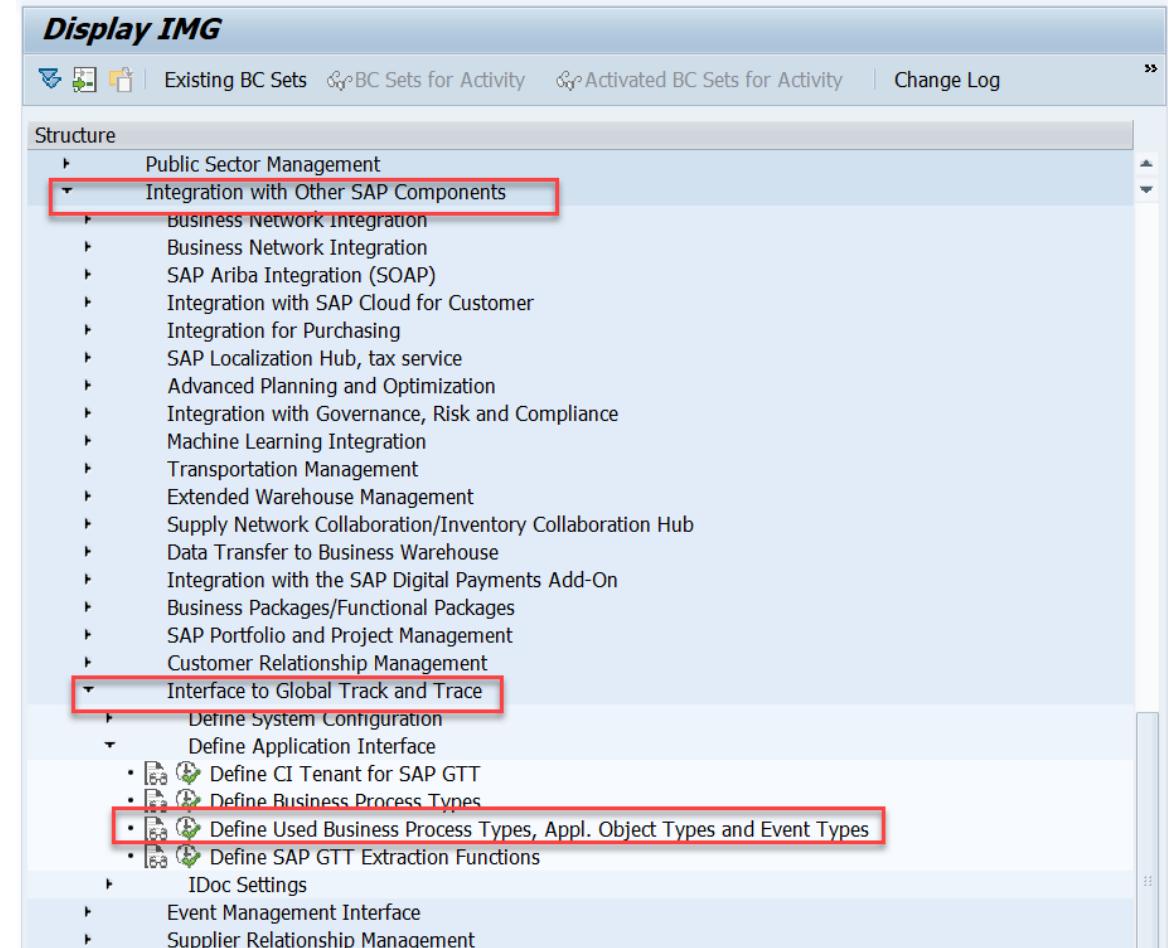
The screenshot shows the SAP SE80 Function Builder interface. The title bar reads "Function Builder: Display ZPOF_GTT_EE_PO_IT_CN_REL_". The left side features a "Repository Browser" with a "Function Group" dropdown set to "ZPOF_GTT" (highlighted with a red box). Below it is an "Object Name" tree view under "Function Modules", listing various function modules like ZPOF_GTT_CTP_DL_TO_PO, ZPOF_GTT_EE_DL_HDR, etc. One specific function, ZPOF_GTT_EE_PO_IT_CN_REL_, is highlighted with a red box in the tree view. The main right panel displays the source code for the selected function:

```
1 FUNCTION ZPOF_GTT_EE_PO_IT_CN_REL_.  
2  
3 *--> Local Interface:  
4   IMPORTING  
5     REFERENCE(I_APPSYS) 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    *-->  
20    * Top Include  
21    * TYPE-POOLS:trxas.  
22  
23  
24  
25  ENDFUNCTION.
```

STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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 Used Business Process Types, Appl. Object Types and Event Types**



STEP 7: 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 7-3 to 7-10 demonstrate how to create an *Event Type* for a given business process type
- Steps 7-11 to 7-21 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
•	• Define Application Object Types	Update Mode
•	• Define Event Types	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 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-3: Choose the business process type from the **Define Used Business Process Types** on the right side

7-4: 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_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 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-5: Click **New Entries** to create a new event type

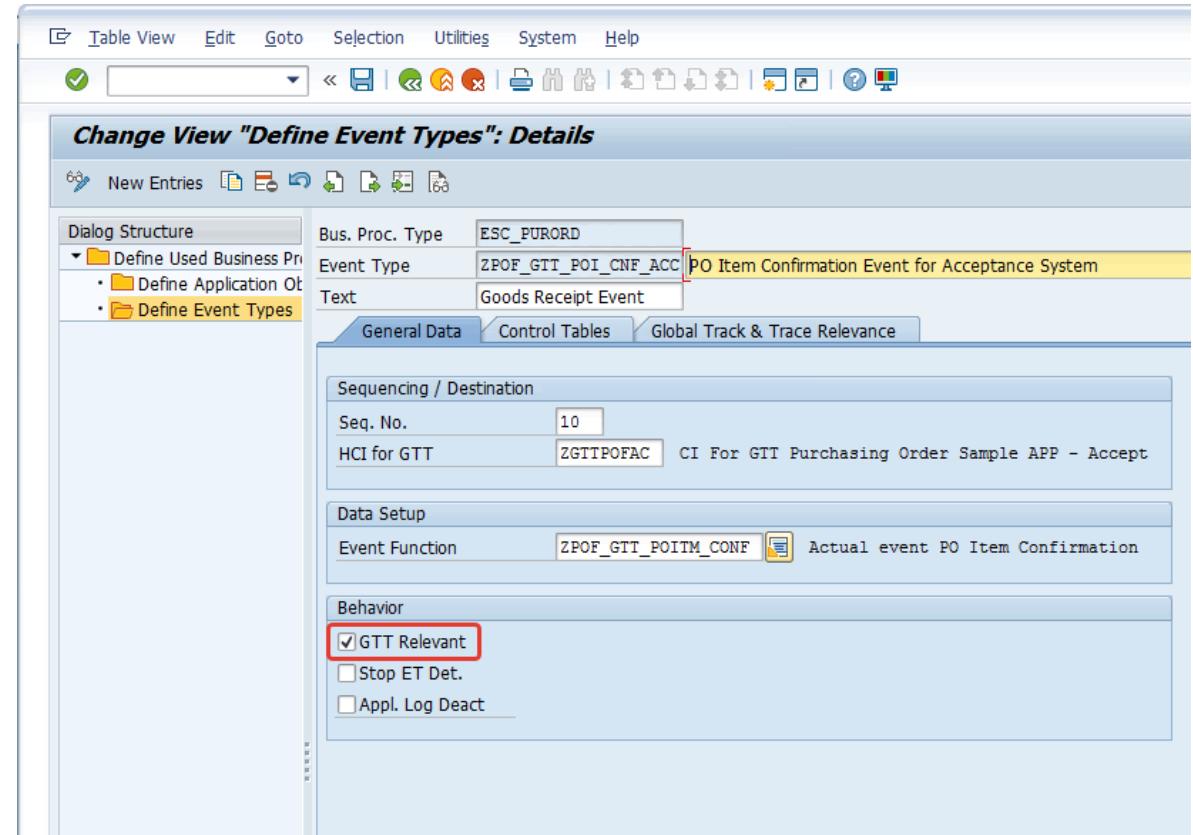
Business Process Type	Event Type	Description
ESC_PURORD	ZPOF_GTT_POI_DEL_ACC	PO Item Confirmation Event for Acceptance System
ESC_PURORD	ZPOF_GTT_POI_DEL_ACZ	PO Item Confirmation Event for Azure System
ESC_PURORD	ZPOF_GTT_POI_DEL_INT	PO Item Confirmation Event for Integration System

STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-6: Fill in the **Event Type** and **Text** fields

7-7: Fill in the information required in the **General Data** tab. **HCI for GTT** is the CI Tenant you created in STEP 5. **Event Function** is the extractor function you created in STEP 6.

7-8: Check **GTT Relevant**



STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-9: 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	ESC_SHIPMT	
Event Type	ZPOF_GTT_SHH_ARR_ACC	Shipment Header Arrival Event for Acceptance System
Text	Arrival Event	
General Data Control Tables Global Track & Trace Relevance		
Data Source for Events		
Main Obj. Table	SHIPMENT_HEADER_NEW	Event on Header Level
Master Table		
Old Main Obj. Table	SHIPMENT_HEADER_OLD	
Old Master Table		

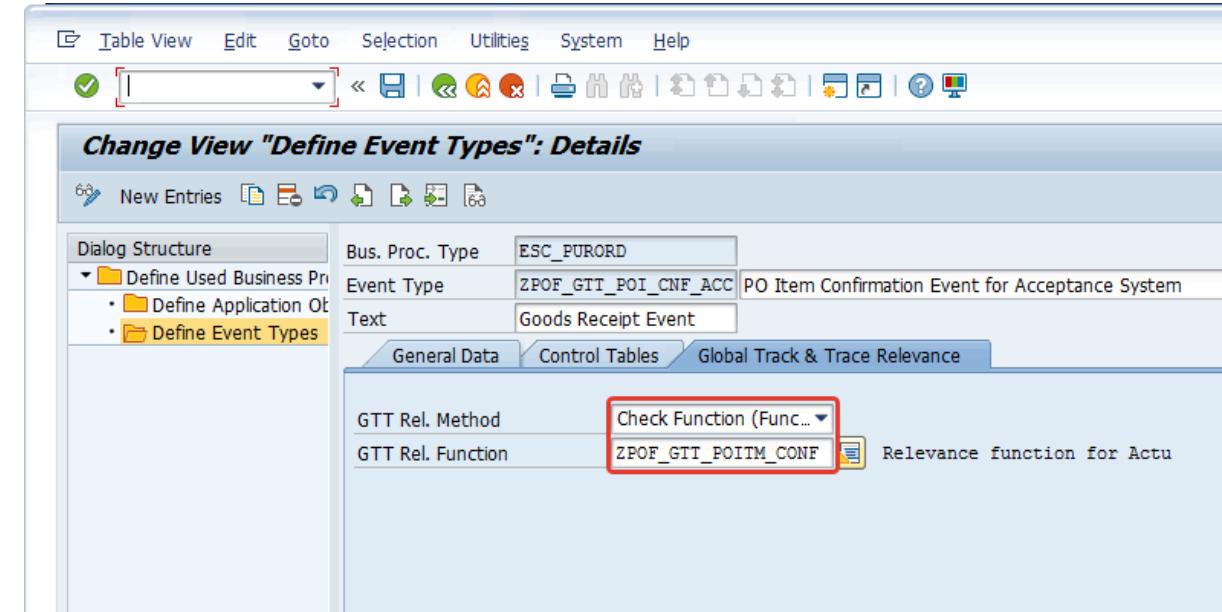
Bus. Proc. Type	ESC_PURORD	
Event Type	ZPOF_GTT_POI_CNF_ACC	PO Item Confirmation Event for Acceptance System
Text	Goods Receipt Event	
General Data Control Tables Global Track & Trace Relevance		
Data Source for Events		
Main Obj. Table	PURCHASE_ITEM_NEW	Event on Item Level
Master Table	PURCHASE_ORDER_HEADER_NEW	
Old Main Obj. Table	PURCHASE_ITEM_OLD	
Old Master Table	PURCHASE_ORDER_HEADER_OLD	
Reference Between Main and Master Table		
First Field Reference from Main to Master Table		
Uplink Field	EBELN	Uplink Mode <input checked="" type="checkbox"/> R
Uplink Target Fld	EBELN	Uplink Const

STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-10: 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 6, and fill in the relevance function name here.

Click **Save**.



STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

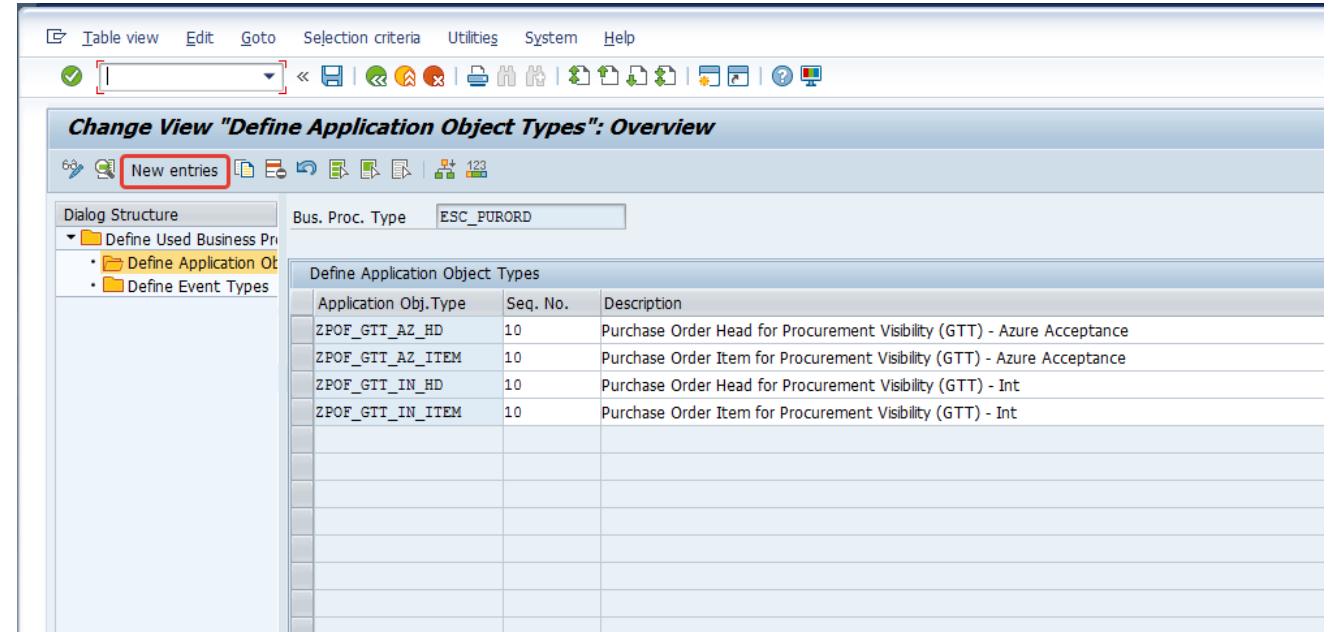
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

7-12: Double click **Define Application Object 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	D Dialog Upd...	Active	Booking Order in Ocean Carrier Booking Process
SNC_MSGIN	D Dialog Upd...	Active	SNC Inbound messages
SNC_PURORD	D Dialog Upd...	Active	SNC Purchase Order
SNC_RPLORD	D Dialog Upd...	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 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-13: Click **New Entries** to create a new Application Object Type



The screenshot shows the SAP Fiori interface for defining application object types. The title bar reads "Change View 'Define Application Object Types': Overview". The toolbar includes standard SAP icons for table view, edit, goto, selection criteria, utilities, system, and help. A red box highlights the "New entries" button in the toolbar. The left sidebar displays a dialog structure with sections for "Define Used Business Procs", "Define Application Obj.", and "Define Event Types". The main area shows a table titled "Define Application Object Types" with columns for "Application Obj. Type", "Seq. No.", and "Description". The table contains four entries:

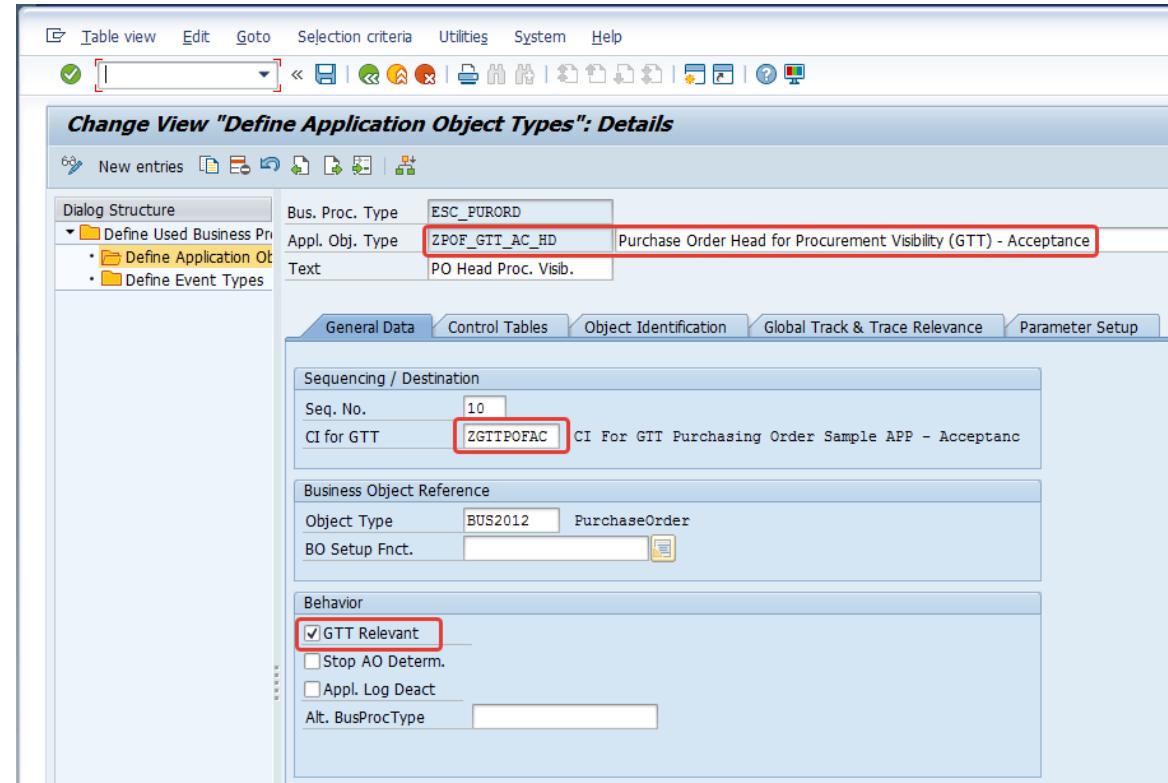
Application Obj. Type	Seq. No.	Description
ZPOF_GTT_AZ_HD	10	Purchase Order Head for Procurement Visibility (GTT) - Azure Acceptance
ZPOF_GTT_AZ_ITEM	10	Purchase Order Item for Procurement Visibility (GTT) - Azure Acceptance
ZPOF_GTT_IN_HD	10	Purchase Order Head for Procurement Visibility (GTT) - Int
ZPOF_GTT_IN_ITEM	10	Purchase Order Item for Procurement Visibility (GTT) - Int

STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-14: Fill in the **Application Object Type** and **Text** fields

7-15: Fill in the information required in the **General Data** tab. **CI for GTT** is the CI Tenant you created in STEP 5.

7-16: Check **GTT Relevant**



STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-17: 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**.

Change View "Define Application Object Types": Details

Bus. Proc. Type	ESC_PURORD	Purchase Order Head for Procurement Visibility (GTT) - Acceptance
Appl. Obj. Type	ZPOF_GTT_AC_HD	
Text	PO Head Proc. Visib.	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameter Setup		
Data Source for Created and Updated Objects		
Main Obj. Table	PURCHASE_ORDER_HEADER_NEW	AOT on Header Level
Master Table		
Data Source for Deleted Objects		
Del.Obj. Table		

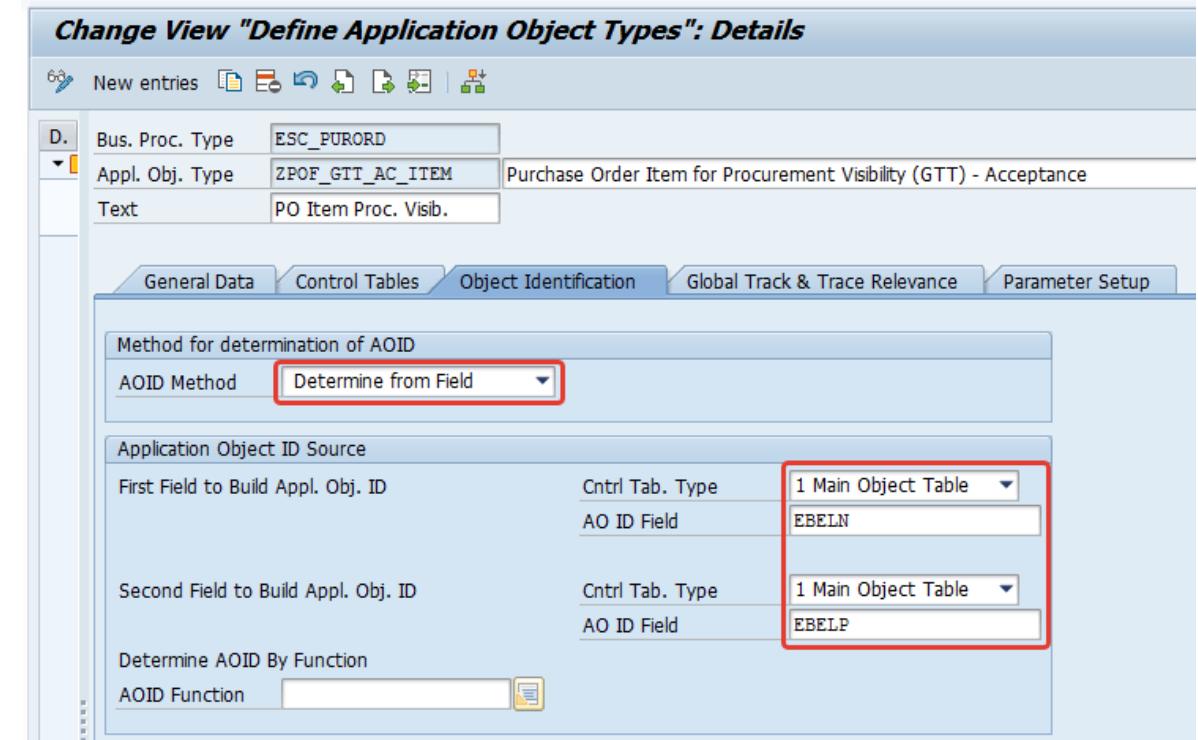
Change View "Define Application Object Types": Details

Bus. Proc. Type	ESC_PURORD	Purchase Order Item for Procurement Visibility (GTT) - Acceptance
Appl. Obj. Type	ZPOF_GTT_AC_ITEM	
Text	PO Item Proc. Visib.	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameter Setup		
Data Source for Created and Updated Objects		
Main Obj. Table	PURCHASE_ITEM_NEW	
Master Table	PURCHASE_ORDER_HEADER_NEW	
Data Source for Deleted Objects		
Del.Obj. Table	PURCHASE_ITEM_OLD	
Reference Between Main and Master Table		
First Field Reference from Main to Master Table		
Uplink Field	EBELN	Uplink Mode R
Uplink Target Fld	EBELN	Uplink Const

STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-18: If there is no customized logic to determine the AOT ID, choose **Determine from Field** and use the key field to fill in the AO ID fields

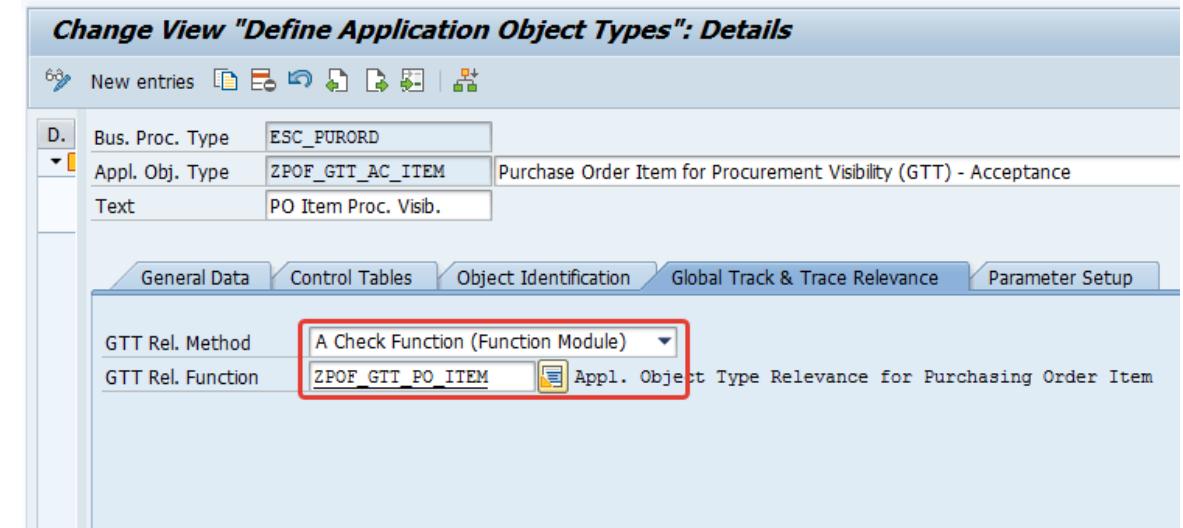
7-19: When choosing **Determine by Function**, you must enter the customized information in the AOID Function field.



STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-20: In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need.

If you choose the **GTT Relevance Method Check Function**, you need to define a relevance function according to STEP 6, and fill in the relevance function name here.



STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

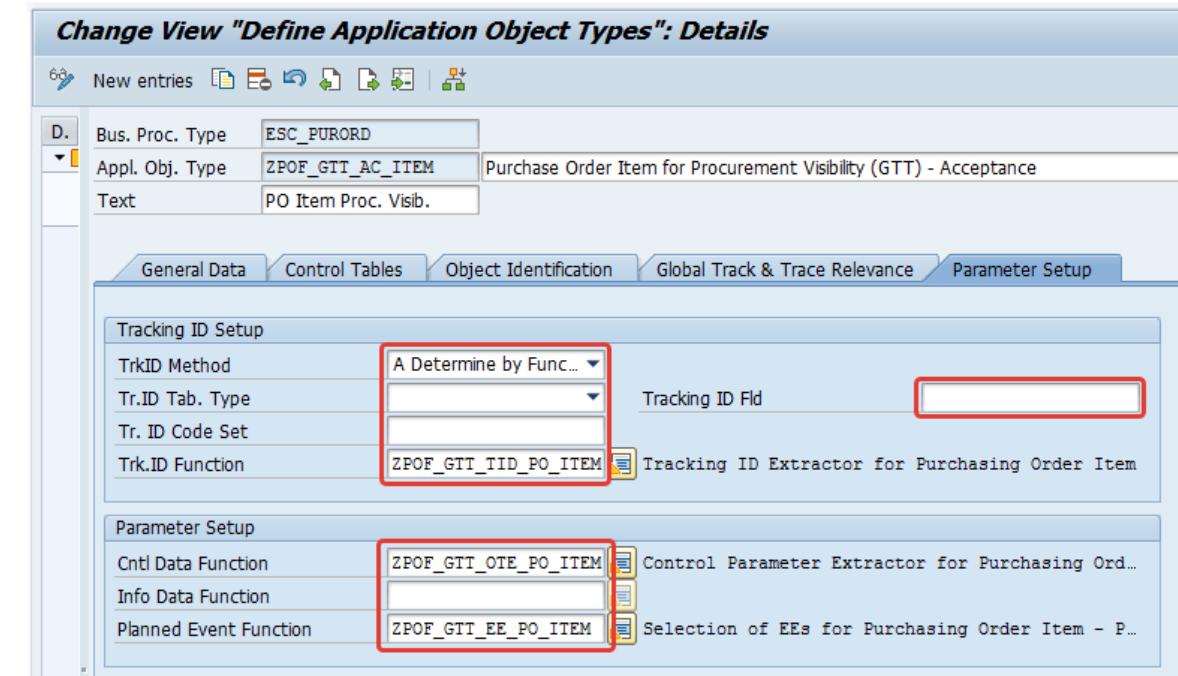
7-21: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

If you choose the **TrkID Method** as *Determine by Function*, then you need to define a tracking ID function according to STEP 6, and fill in the relevance function name here.

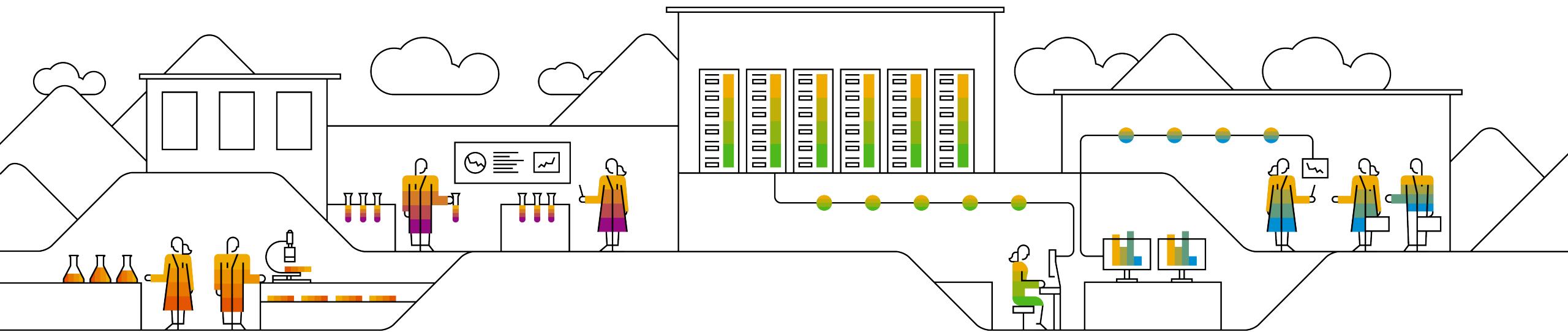
If no customized logic exists, for **TrkID Method** choose *Determine from Field*, then fill in the key field and name the Code Set for the AOT.

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

Click **Save**.



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 at <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 consists 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

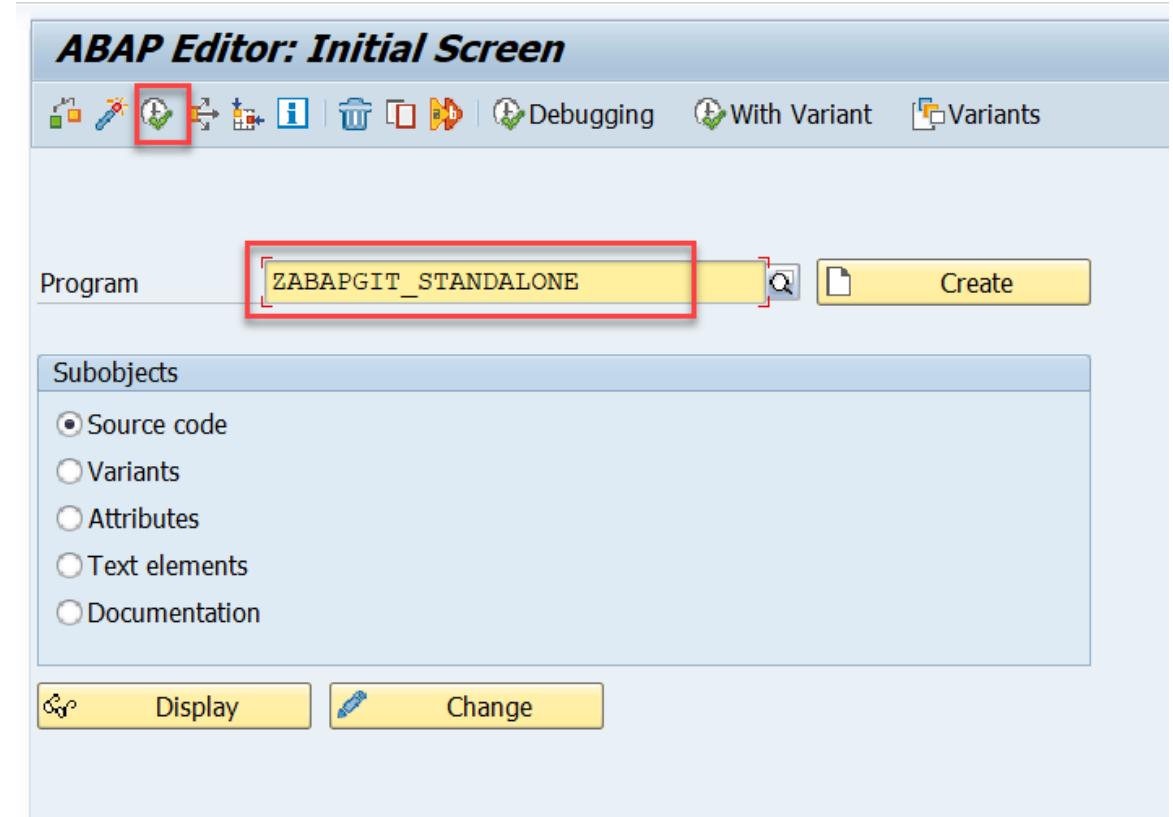
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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 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' application interface. At the top, there's a header bar with the title 'ABAP GIT for GTT'. Below it is a navigation bar with a logo, the text 'abapGit ► Repository List', and several buttons: 'New Online' (highlighted with a red box), 'New Offline', 'X', and '?'. There are also filter options: 'Filter:' with a search input field, 'Only Favorites' (checked), and 'Detail' (checked). The main area is a table with columns: 'Name' (sorted by name), 'Url' (repository URLs), 'Package' (empty), 'Branch' (empty), and 'Action' (empty). Two repository entries are listed, both with their URLs blurred. At the bottom center is the 'abapGit' logo and version '1.98.0'. On the right side, there's a message 'js: OK'.

STEP 2: Download ABAP Code

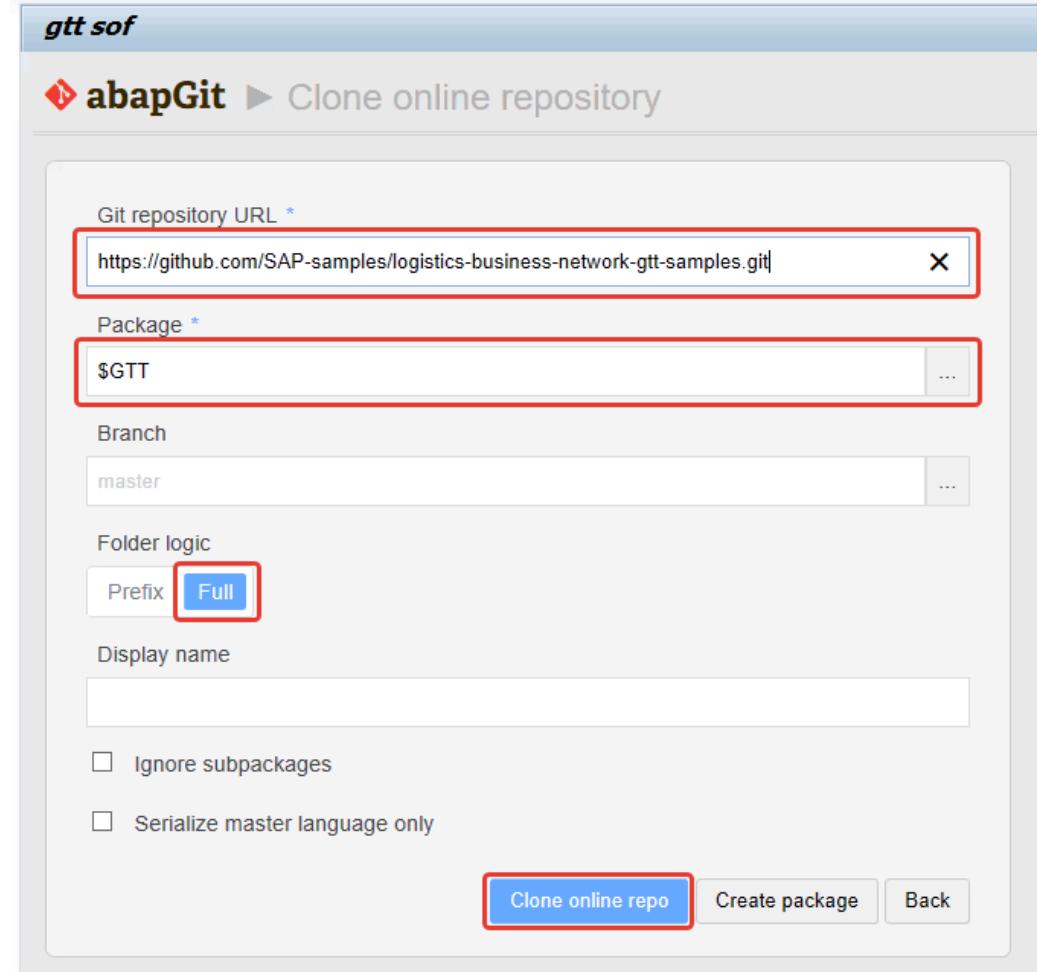
2-4: Fill in the **Git repository URL**:

<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

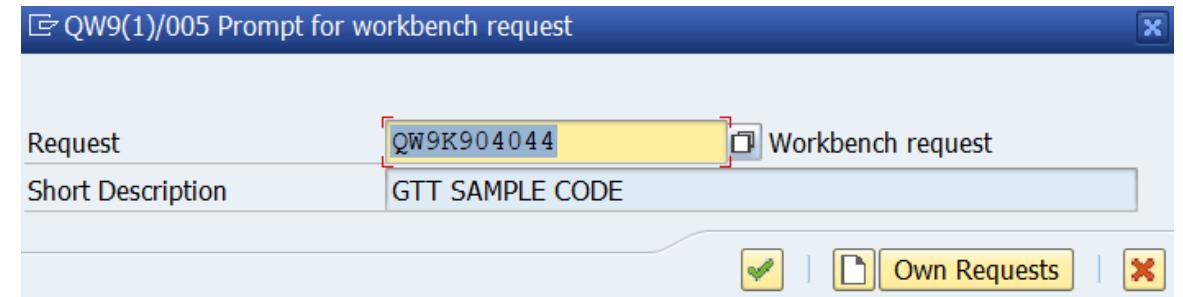
TIP: Clicking **Pull** action will download the whole package of the sample codes. If you want to download the specified folder's codes in Github, check details in Step 3-1 to 3-8.

The screenshot shows the abapGit interface. At the top, there is a navigation bar with 'abapGit' logo, 'Repository' link, 'Repository List' button, 'Settings' button, and other icons. Below the navigation bar, the URL 'POF https://github.wdf.sap.corp/TrackAndTrace/GTT-V2-Sample-Apps-Sandbox.git eacdfd4' is displayed, along with a star icon, a 'master' branch indicator, and a 'zgtt_sample_pof' tag. A toolbar below the URL contains buttons for 'Pull' (highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a settings gear icon. The main area displays a table with columns 'Type', 'Name', and 'Path'. The table lists several files and folders:

Type	Name	Path	diff
non-code and meta files		./abapgit.xml	[diff]
MSAG	ZPOF_GTT	/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.msag.xml	[diff]
TABL	ZPOF_GTT_EE_REL	/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt_ee_rel.tabl.xml	[diff]
DTEL	ZPOF_KOSTA	/lbn-gtt-template-tpo/abap/zsrc/zpof_kosta.dtel.xml	[diff]
DTEL	ZPOF_PDSTK	/lbn-gtt-template-tpo/abap/zsrc/zpof_pdstk.dtel.xml	[diff]
DTEL	ZPOF_PKSTA	/lbn-gtt-template-tpo/abap/zsrc/zpof_pksta.dtel.xml	[diff]
DTEL	ZPOF_WBSTA	/lbn-gtt-template-tpo/abap/zsrc/zpof_wbsta.dtel.xml	[diff]

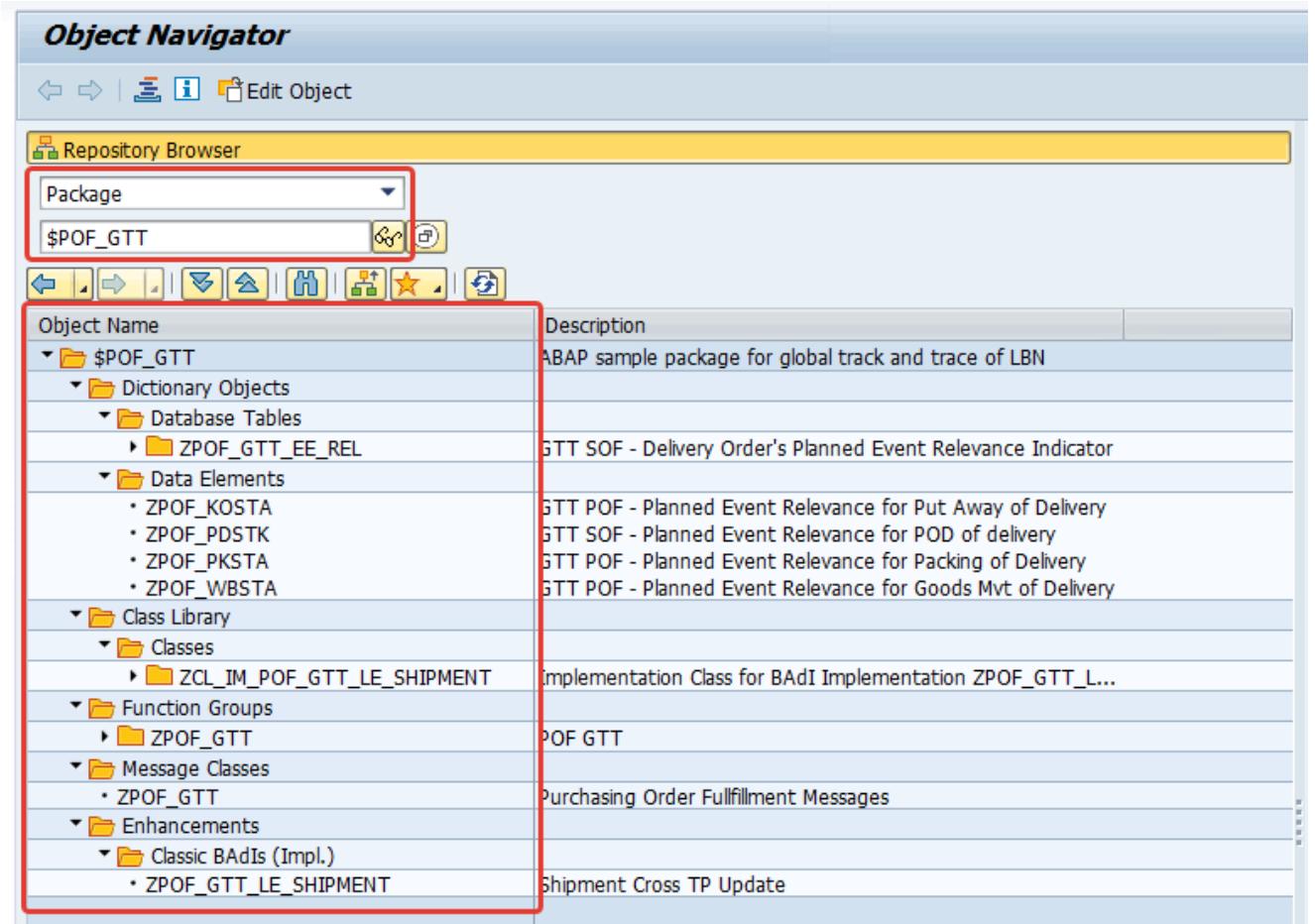
STEP 2: Download ABAP Code

2-9: Assign the change to a change request. If you do not have any available change requests, 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 you want to download only the sample codes in the folder of 'Ibn-gtt-template-tpo' from Github instead of downloading all, click **Fork** button to pop up a dialog window

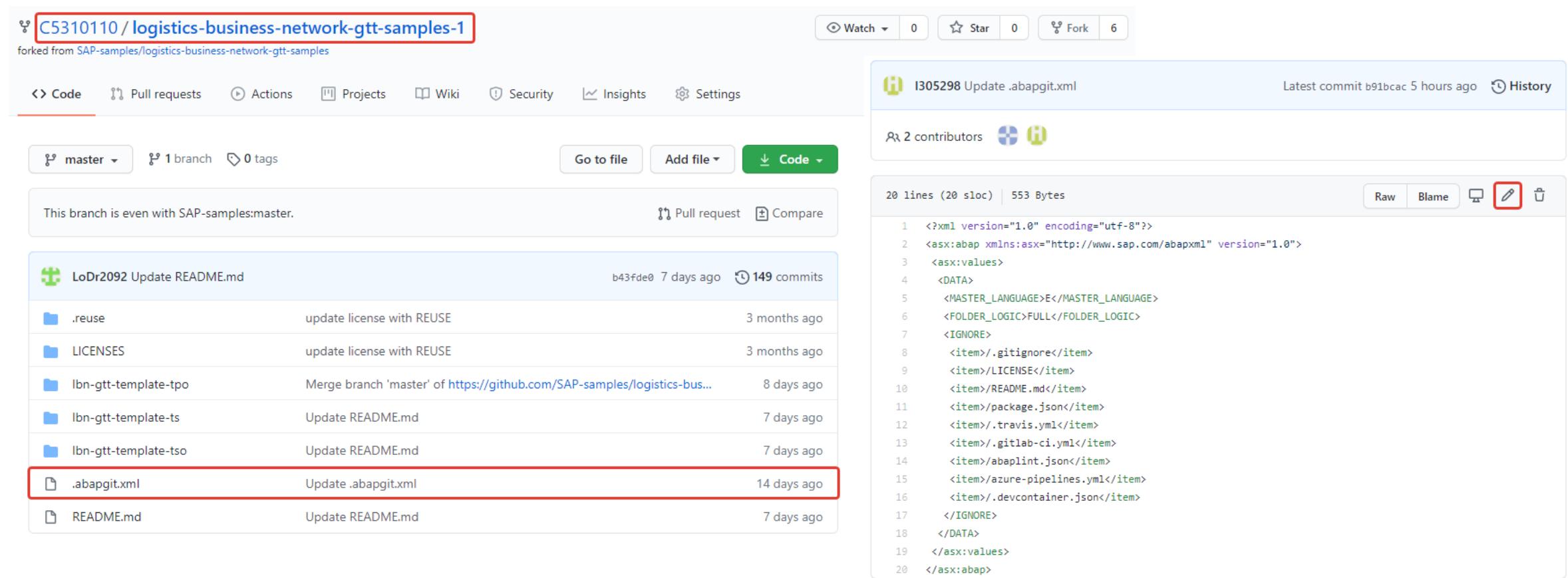
3-2: Click the user account and it will copy the newest version into the user's account

The screenshot shows two parts of the GitHub interface. On the left, the main repository page for 'SAP-samples / logistics-business-network-gtt-samples' is displayed. The 'Code' tab is selected. A red box highlights the 'Fork' button in the top right corner of the header, which has a value of '6'. Below the header, there are navigation links for Code, Issues (1), Pull requests, Actions, Projects, Security, and Insights. Underneath these are buttons for master (selected), 1 branch, 0 tags, Go to file, and Code (dropdown). The repository's contents are listed in a table, with the row for 'Ibn-gtt-template-tpo' highlighted by a red box. This row contains a file icon, the folder name 'Ibn-gtt-template-tpo', a merge commit message 'Merge branch 'master' of https://github.com/SAP-sa...', and a timestamp '8 days ago'. To the right of the repository details is an 'About' section containing a description of the code example and a list of tags: sample, sample-code, sap-logistics-business-network, global-track-and-trace, Ibn, gtt, sales-order-fulfillment, and Readme. On the right, a modal dialog titled 'Fork logistics-business-network-gtt-samples' is shown. It asks 'Where should we fork logistics-business-network-gtt-samples?' and displays a dropdown menu with a single option 'I305298', which is also highlighted with a red box.

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 'C5310110 / logistics-business-network-gtt-samples-1'. The repository has 0 stars and 6 forks. The 'Code' tab is selected, showing a list of files. The '.abapgit.xml' file is highlighted with a red border. On the right, a detailed view of the '.abapgit.xml' file is shown, titled 'I305298 Update .abapgit.xml'. It was updated 5 hours ago by 'b91bcac'. The file contains 20 lines of XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx: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>
</asx:values>
</asx:abap>
```

Step 3: Download ABAP Code Within the Specified Folder

3-5: Add the sentence of '<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>' as below

3-6: Commit change

The screenshot shows a GitHub repository page for 'C5310110 / logistics-business-network-gtt-samples-1'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected, showing the file '.abapgit.xml' in the 'master' branch. The code editor displays the XML content of the file. A red box highlights the line containing the path '<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>'. To the right, a 'Commit changes' dialog is open, prompting for a commit message ('Update .abapgit.xml') and an optional extended description. It offers two options: committing directly to the master branch (selected) or creating a new branch. The 'Commit changes' button is highlighted with a red border.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <DATA>
      <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
      <STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>
      <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>
</asx:values>
</asx:abap>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

-o- Commit directly to the `master` branch.

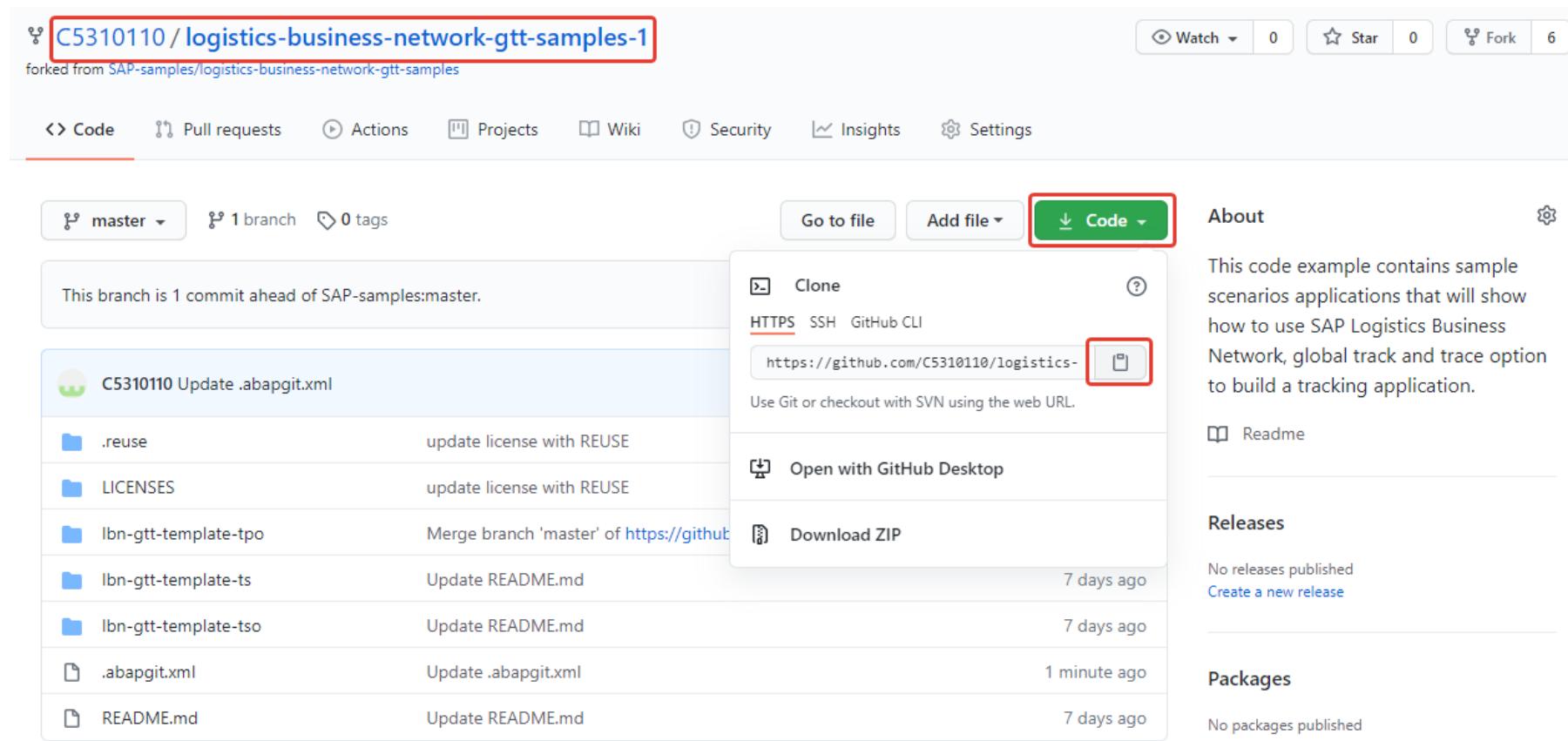
↗ Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes Cancel

Step 3: Download ABAP Code Within the Specified Folder

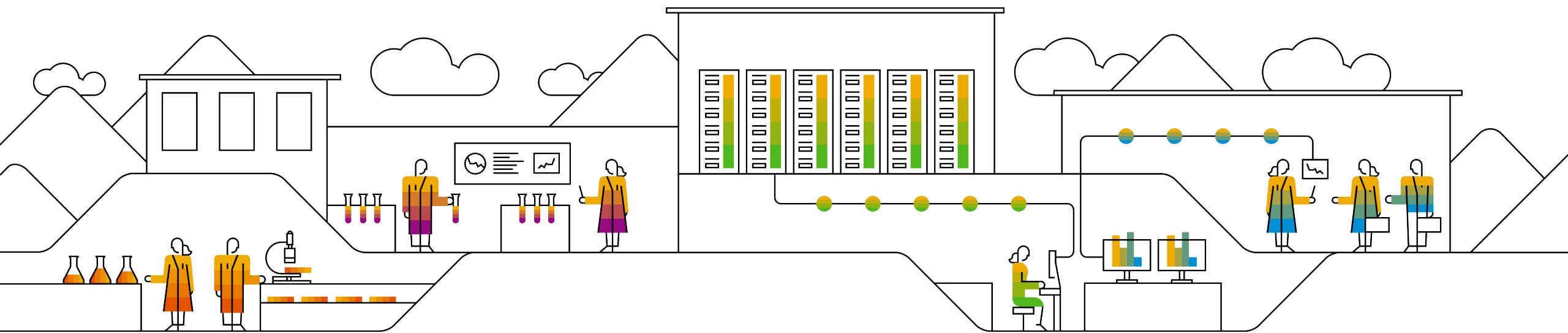
3-7: Go to the root and copy the repository URL by clicking  button

3-8: Repeat Step 2-4 to 2-10



D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

When you create Application Object Type for one Business Process Type, make sure the AOT name must be as same as the name defined in the corresponding model in the Manage Models app in GTT Version 2.

The image displays two SAP application screenshots side-by-side. On the left is the SAP AOT interface titled "Change View 'Define Application Object Types': Details". It shows a table with columns: Bus. Proc. Type (ESC_PURORD) and Appl. Obj. Type (ZPOF_GTT_AC_HD). The "Appl. Obj. Type" field is highlighted with a red box. Below the table are tabs: General Data (selected), Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. Under "General Data", there are sections for Sequencing / Destination (Seq. No. 10, CI for GTT ZGTTPOFAC) and Business Object Reference (Object Type BUS2012 PurchaseOrder, BO Setup Fnct.). On the right is the SAP "Model Details" screen from the Manage Models app. It shows a navigation bar with "SAP Model Details Internal - Test" and tabs: Tracked Process, Field Type Pool, Event Type Pool, Code List, IDOC Integration (highlighted with a red box), Visibility Provider Integration, Planned Event Extension, and Event to Action. The "Tracked Process" tab is selected, showing "PurchaseOrder" as the tracked process. The "IDOC Integration" tab shows a mapping between "PurchaseOrder" and "PurchaseOrderEvent" with IDOC code E1EHPAO. The "Fields" section lists fields like purchaseOrderNo, supplierId, plannedDeliveryDate, netValue, currency, incotermsVersion, and incoterms, each mapped to an IDOC segment and field. The "Application Object Type" field in the "Tracked Process Mapping" section is also highlighted with a red box and contains the value ZPOF_GTT_AC_HD.

2: Maintain Tracking ID Type

In the AOT you maintained, make sure the name of Tracking ID Type is as same as the name defined in the corresponding process type of the model in the Manage Models app in GTT Version 2.

If the Tracking ID Type is determined by Field, input the value source field in the Tracking ID field, and the Code Set which refers to the Tracking ID Type for the AOT as below.

Change View "Define Application Object Types": Details

D. Bus. Proc. Type: ESC_PURORD
Appl. Obj. Type: ZPOF_GTT_AC_HD [Purchase Order Head for Procurement Visibility (GTT) - Acceptance]
Text: PO Head Proc. Visib.

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

Tracking ID Setup
TrkID Method: B Determine from Field
Tr.ID Tab. Type: 1 Main Object Table
Tr. ID Code Set: PURCHASE_ORDER
Trk.ID Function:

Parameter Setup
Cntl Data Function: ZPOF_GTT_OTE_PO_HD
Info Data Function:
Planned Event Function: ZPOF_GTT_EE_PO_HD

SAP Model Details Internal - Test

pof Active Purchase Order Fulfillment Namespace: com.lbngttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List IDOC Integration Visibility Provider Integration Planned Event Extension Event to Action

Items (6) Create Edit Delete

PurchaseOrder
Description: Purchase Order
Tracking Id Type: PURCHASE_ORDER

PurchaseOrderItem
Description: PurchaseOrderItem
Tracking Id Type: PURCHASE_ORDER_ITEM

InboundDelivery
Description: Inbound Delivery
Tracking Id Type: INBOUND_DELIVERY

InboundDeliveryItem
Description: Inbound Delivery Item
Tracking Id Type: INBOUND_DELIVERY_IT

Create Tracked Process

Name: * PurchaseOrder
Description:
Tracking Id Type: * PURCHASE_ORDER

OK Cancel

3: 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 Tracking ID Type and AOT ID, you need to enable *Determine by Function* option.

For customization of GTT relevance, you need to enable *Check Function(Function Module)* option.

Extractor	Description
510_WRF_CONTR_01	Control Parameters for Purchase Order (Seasonal Procurement)
CONTR_PARAM_DELIV	Selection of Control parameters for Deliveries in Shipment
OBP10_DELIV	Selection of CPs for Delivery - Outbound Delivery Visibility Process
OBP10_HU_IN_DLV	Selection of CPs for HUs in Delivery - Outbound Delivery Visibility Process
OCB10_CONTAINER	Selection of CPs for Containers in Ocean Carrier Booking Process
OCB10_ORDER	Selection of CPs for Booking Orders in Ocean Carrier Booking Process
ODT20_TOR	Selection of Control Parameters - Transportation Execution Visib. Proc.
ODT30_INS	Selection of Cntrl Parameters - Instruction Execution Visibility Procoress
ODT40_TOR	Selection of Control Parameters - Transportation Execution Visib. Proc.
PCM10_ITEM	Selection of CPs for Purchase Order Item - Procurement Visibility Process
PMF10_NOTIF	Selection of CPs for Notification - Production Malfunction Visibility Process
PMF10_ORDER	Selection of CPs for Manuf. Order - Production Malfunction Visibility Process
RES30_CPARAM	Selection of Control Parameters - Resource Tracking Visibility Process
SNC10_MSGIN	Control Parameter Extractor for SNC Messages
SNC10_PURORD	Control Parameter Extractor for SNC Purchase Order
SNC10_RPLORD	Control Parameter Extractor for SNC Replenishment Order
TRA10_DELIV	Selection of CPs for Deliveries in Road Shipment - Transp. Visibility Process
TRA10_ROADSEA	Selection of CPs for Road/Sea Shipment - Transp. Visibility Process
ZGTT_OBP10_DELIV	Selection of CPs for Delivery - Outbound Delivery Visibility Process
ZGTT_OTE_DE_HDR	Control Parameter Extractor for Outbound Delivery Header
ZGTT_OTE_DE_ITEM	Control Parameter Extractor for Outbound Delivery Item
ZGTT_OTE_SHP_HDR	Control Parameter Extractor for Shipment Header
ZGTT_OTE_SO_HDR	Control Parameter Extractor for Sales Order Header
ZPOF_GTT_OTE_DL_HDR	Control Parameter Extractor for Inbound Delivery Header
ZPOF_GTT_OTE_DL_ITEM	Control Parameter Extractor for Inbound Delivery Item
ZPOF_GTT_OTE_PO_HDR	Control Parameter Extractor for Purchasing Order Header
ZPOF_GTT_OTE_PO_ITEM	Control Parameter Extractor for Purchasing Order Item
ZPOF_GTT_OTE_SH_HDR	Control Parameter Extractor for Shipment Header
ZSST_GTT_OTE_FO_HDR	Control Parameter Extractor for Freight Order

4: Sample Codes for Track PO Fulfillment Template App

4-1 To support the Track PO Fulfillment template app, the sample codes cover the following cases by function group ZPOF_GTT:

Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_HDR	Control Parameter Extractor for Inbound Delivery Header
Control Parameter Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM	Control Parameter Extractor for Inbound Delivery Item
Control Parameter Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_HDR	Control Parameter Extractor for Purchasing Order Header
Control Parameter Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM	Control Parameter Extractor for Purchasing Order Item
Control Parameter Extractors	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR	Control Parameter Extractor for Shipment Header
Event Data Extractors	ESC_MATDOC	ZPOF_GTT_EE_DL_HDR_GR	Actual event DLV Header Goods Receipt
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA	Actual event DLV Item Put Away
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG	Actual event DLV Item Packing
Event Data Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_CONF	Actual event PO Item Confirmation
Event Data Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_DEL	Actual event PO Item Deletion
Event Data Extractors	ESC_MATDOC	ZPOF_GTT_EE_PO_ITEM_GR	Actual event PO Item Goods Receipt
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_ARR	Actual event Shipment Header Arrival
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_CI	Actual event Shipment Header Check In
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_DEP	Actual event Shipment Header Departure
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LE	Actual event Shipment Header Load End
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LS	Actual event Shipment Header Load Start
Planned Event Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_HDR	Selection of EEs for Inbound Delivery Item - Procurement Visibility Process
Planned Event Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_HDR	Selection of EEs for Purchasing Order Header - Procurement Visibility Process
Planned Event Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM	Selection of EEs for Purchasing Order Item - Procurement Visibility Process
Planned Event Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR	Selection of EEs for Shipment Header - Procurement Visibility Process
Tracking ID Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM_TID	Tracking ID Extractor for Inbound Delivery Item
Tracking ID Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM_TID	Tracking ID Extractor for Purchasing Order Item
Tracking ID Extractors	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR_TID	Tracking ID Extractor for Shipment Header

4: Sample Codes for Track PO Fulfillment Template App

Continued from the previous table:

Category	Business Process Type	Function Module Name	Description
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_HDR_REL	Appl. Object Type Relevance for Inbound Delivery Header
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM_REL	Appl. Object Type Relevance for Inbound Delivery Item
GTT relevance function of AOT	ESC_PURORD	ZPOF_GTT_OTE_PO_HDR_REL	Appl. Object Type Relevance for Purchasing Order Header
GTT relevance function of AOT	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM_REL	Appl. Object Type Relevance for Purchasing Order Item
GTT relevance function of AOT	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR_REL	Appl. Object Type Relevance for Shipment Header
GTT relevance function of Event Type	ESC_MATDOC	ZPOF_GTT_EE_DL_HDR_GR_REL	Relevance function for Actual event DLV Header Goods Receipt
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA_REL	Relevance function for Actual event DLV Item Put Away
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG_REL	Relevance function for Actual event DLV Item Packing
GTT relevance function of Event Type	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_CONF_REL	Relevance function for Actual event PO Item Confirmation
GTT relevance function of Event Type	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_DEL_REL	Relevance function for Actual event PO Item Deletion
GTT relevance function of Event Type	ESC_MATDOC	ZPOF_GTT_EE_PO_ITEM_GR_REL	Relevance function for Actual event PO Item Goods Receipt
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_ARR_REL	Relevance function for Actual event Header Arrival
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_CI_REL	Relevance function for Actual event Header Check In
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_DEP_REL	Relevance function for Actual event Header Departure
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LE_REL	Relevance function for Actual event Header Load End
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LS_REL	Relevance function for Actual event Header Load Start
Cross TP Update Function	ESC_DELIV	ZPOF_GTT_CTP_SH_TO_DL	Cross TP Update from Shipment to Delivery

4: Sample Codes for Track PO Fulfillment Template App

4-2 To support the Track PO Fulfillment template app, the following extractors should be set up:

Business Process Type	Extractor Type	Extraction Level	Description	Control Tables	Used Function Modules
ESC_PURORD	AOT Type	PO Header	Purchase Order Head for Procurement Visibility (GTT)	PURCHASE_ORDER_HEADER_NEW	ZPOF_GTT_OTE_PO_HDR ZPOF_GTT_OTE_PO_HDR_REL
ESC_PURORD	AOT Type	PO Item	Purchase Order Item for Procurement Visibility (GTT)	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD	ZPOF_GTT_OTE_PO_ITEM ZPOF_GTT_OTE_PO_ITEM_REL ZPOF_GTT_EE_PO_ITEM ZPOF_GTT_OTE_PO_ITEM_TID
ESC_PURORD	Event Type	PO Item	Purchase Order Item Confirmation Event	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD PURCHASE_ORDER_HEADER_OLD	ZPOF_GTT_EE_PO_ITEM_CONF ZPOF_GTT_EE_PO_ITEM_CONF_REL
ESC_PURORD	Event Type	PO Item	Purchase Order Item Deletion Event	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD PURCHASE_ORDER_HEADER_OLD	ZPOF_GTT_EE_PO_ITEM_DEL ZPOF_GTT_EE_PO_ITEM_DEL_REL
ESC_MATDOC	Event Type	PO Item	Purchase Order Item Goods Receipt Event	MATERIAL_SEGMENT MATERIAL_HEADER	ZPOF_GTT_EE_PO_ITEM_GR ZPOF_GTT_EE_PO_ITEM_GR_REL
ESC_DELIV	AOT Type	DLV Header	Inbound Delivery Head for Procurement Visibility (GTT)	DELIVERY_HEADER_NEW DELIVERY_HEADER_OLD	ZPOF_GTT_OTE_DL_HDR ZPOF_GTT_OTE_DL_HDR_REL ZPOF_GTT_EE_DL_HDR
ESC_DELIV	AOT Type	DLV Item	Inbound Delivery Item for Procurement Visibility (GTT)	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD	ZPOF_GTT_OTE_DL_ITEM ZPOF_GTT_OTE_DL_ITEM_REL ZPOF_GTT_EE_DL_ITEM ZPOF_GTT_OTE_DL_ITEM_TID
ESC_DELIV	Event Type	DLV Item	Inbound Delivery Item Put Away Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_ITEM_PA ZPOF_GTT_EE_DL_ITEM_PA_REL

4: Sample Codes for Track PO Fulfillment Template App

Continued from the previous table:

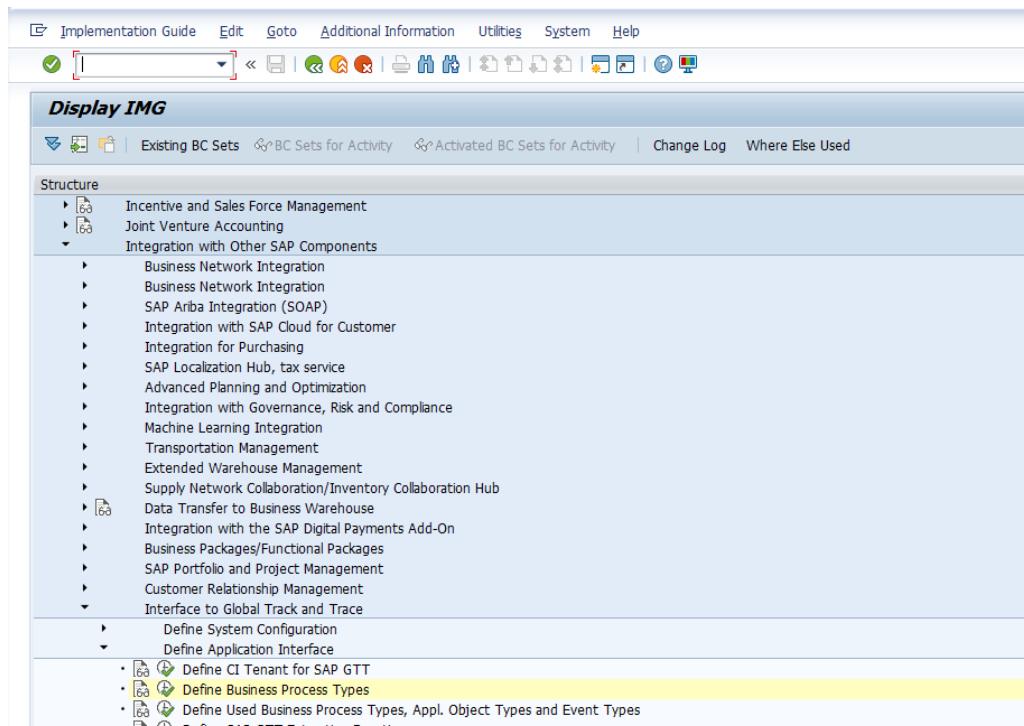
Business Process Type	Extractor Type	Extraction Level	Description	Control Tables	Used Function Modules
ESC_DELIV	Event Type	DLV Item	Inbound Delivery Item Packing Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_ITEM_PKNG ZPOF_GTT_EE_DL_ITEM_PKNG_REL
ESC_MATDOC	Event Type	DLV Header	Inbound Delivery Header Goods Receipt Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_HDR_GR ZPOF_GTT_EE_DL_HDR_GR_REL
ESC_SHIPMT	AOT Type	Shipment	Shipment for Procurement Visibility (GTT)	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_OTE_SH_HDR ZPOF_GTT_OTE_SH_HDR_REL ZPOF_GTT_EE_SH_HDR ZPOF_GTT_OTE_SH_HDR_TID
ESC_SHIPMT	Event Type	Shipment	Shipment Arrival Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_ARR ZPOF_GTT_EE_SH_HDR_ARR_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Check In Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_CI ZPOF_GTT_EE_SH_HDR_CI_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Departure Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_DEP ZPOF_GTT_EE_SH_HDR_DEP_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Load End Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_LE ZPOF_GTT_EE_SH_HDR_LE_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Load Start Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_LS ZPOF_GTT_EE_SH_HDR_LS_REL

5: Available Contexts for the Extractors' Modules

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

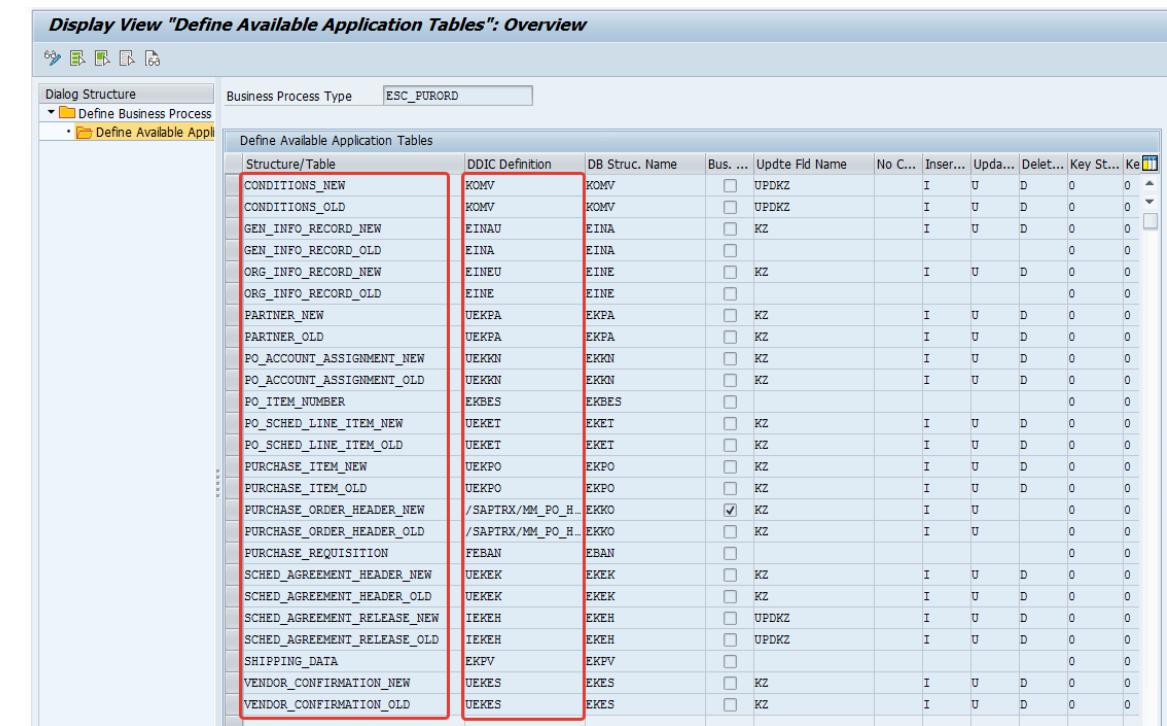
5-2: Choose activity **Define Business Process Types**

5-3: Select the **Business Process Types** to find all the context tables and their structure info.



The screenshot shows the SAP Display IMG interface. The top menu bar includes Implementation Guide, Edit, Goto, Additional Information, Utilities, System, and Help. Below the menu is a toolbar with various icons. The main area is titled "Display IMG" and contains a breadcrumb trail: Existing BC Sets > BC Sets for Activity > Activated BC Sets for Activity. It also has links for Change Log and Where Else Used. A tree view under "Structure" shows several categories, with "Integration with Other SAP Components" expanded to show sub-options like Business Network Integration, SAP Ariba Integration (SOAP), and Interface to Global Track and Trace. Under "Interface to Global Track and Trace", "Define Application Interface" is selected, highlighted with a yellow background. A legend at the bottom right indicates that yellow highlights mean "Defined".

Display View "Define Available Application Tables": Overview



This screenshot shows the "Display View 'Define Available Application Tables': Overview" dialog. The title bar includes "Display View", "ESC_PURORD", and a "Business Process Type" dropdown set to "Define Business Process". The main area is a grid table with the following columns:

Structure/Table	DDIC Definition	DB Struc. Name	Bus. ...	Updtc Fld Name	No C...	Inser...	Upda...	Delet...	Key St...	Ke...
CONDITIONS_NEW	KOMV		<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
CONDITIONS_OLD	KOMV		<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
GEN_INFO_RECORD_NEW	EINAU	EINA	<input type="checkbox"/>	KZ	I	U	D	0	0	
GEN_INFO_RECORD_OLD	EINA	EINA	<input type="checkbox"/>					0	0	
ORG_INFO_RECORD_NEW	EINEU	EINE	<input type="checkbox"/>	KZ	I	U	D	0	0	
ORG_INFO_RECORD_OLD	EINE	EINE	<input type="checkbox"/>					0	0	
PARTNER_NEW	UEKPA	EKPA	<input type="checkbox"/>	KZ	I	U	D	0	0	
PARTNER_OLD	UEKPA	EKPA	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ACCOUNT_ASSIGNMENT_NEW	UEKKN	EKKN	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ACCOUNT_ASSIGNMENT_OLD	UEKKN	EKKN	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ITEM_NUMBER	EKBES	EKBES	<input type="checkbox"/>					0	0	
PO_SCHED_LINE_ITEM_NEW	UEKET	EKET	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_SCHED_LINE_ITEM_OLD	UEKET	EKET	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ITEM_NEW	UEKPO	EKPO	<input type="checkbox"/>		I	U	D	0	0	
PURCHASE_ITEM_OLD	UEKPO	EKPO	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ORDER_HEADER_NEW	/SAPTRX/MM_PO_H	EKKO	<input checked="" type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ORDER_HEADER_OLD	/SAPTRX/MM_PO_H	EKKO	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_REQUISITION	FEBAN		<input type="checkbox"/>					0	0	
SCHED_AGREEMENT_HEADER_NEW	UEKEK	EKEK	<input type="checkbox"/>	KZ	I	U	D	0	0	
SCHED_AGREEMENT_HEADER_OLD	UEKEK	EKEK	<input type="checkbox"/>	KZ	I	U	D	0	0	
SCHED_AGREEMENT_RELEASE_NEW	IEKEH	EKEH	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
SCHED_AGREEMENT_RELEASE_OLD	IEKEH	EKEH	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
SHIPPING_DATA	EKPV		<input type="checkbox"/>					0	0	
VENDOR_CONFIRMATION_NEW	UEKES	EKES	<input type="checkbox"/>	KZ	I	U	D	0	0	
VENDOR_CONFIRMATION_OLD	UEKES	EKES	<input type="checkbox"/>	KZ	I	U	D	0	0	

6: Coding Tips in the GTT Relevance Function Modules

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

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: *ZPOF_GTT_OTE_PO_ITEM_REL*

The image shows two SAP ABAP development environments side-by-side. On the left is the 'Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_REL' window, and on the right is the 'ABAP Editor: Display Include LZPOF_GTTD20' window.

Function Builder (Left):

- Object Name:** ZPOF_GTT_OTE_PO_ITEM_REL
- Source Code:** Displays the ABAP code for the function module. The code handles various data types like *lt_app_objects*, *io_udm_message*, and *ls_bapiret*. It includes a TRY-CATCH block for *cx_udm_message* and uses *lcl_ef_performer* to check relevance. Error handling is done through *APPEND ls_bapiret TO c_logtable* and RAISING exceptions.

ABAP Editor (Right):

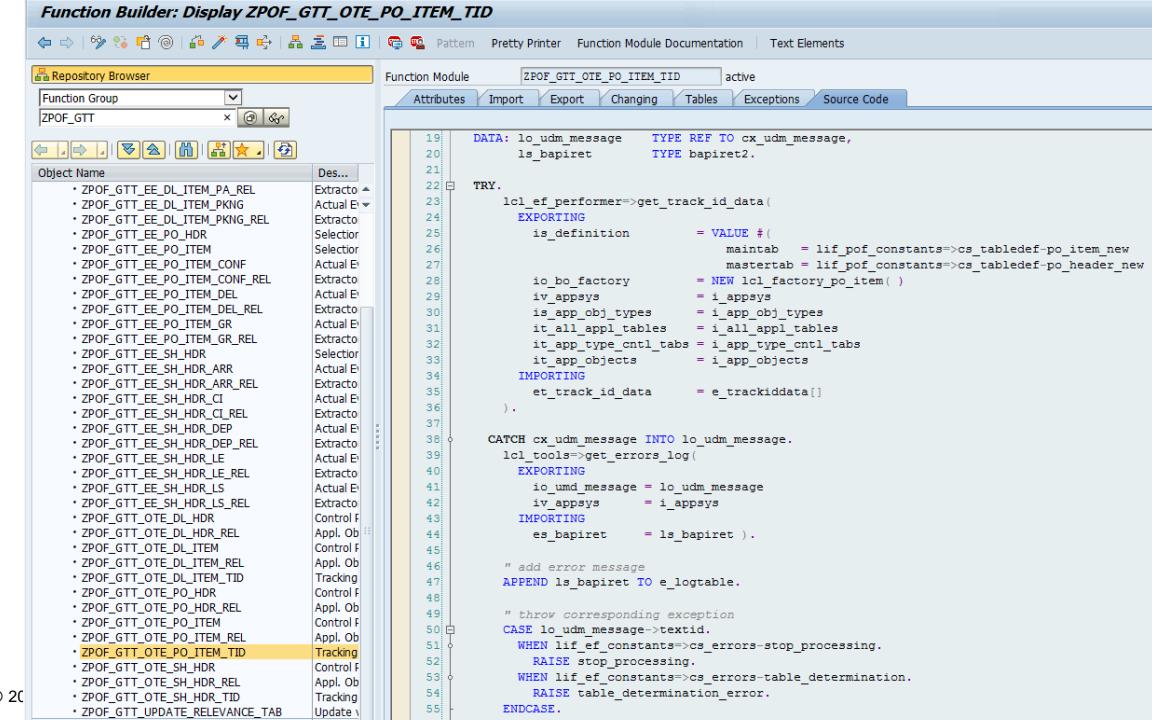
- Include:** LZPOF_GTTD20
- Object Name:** LCL_BO_READER~CHECK_R
- Code:** Shows the implementation of the *lif_bo_reader~check_relevance* method. It performs several checks:
 - Basic check of main table.
 - Check that only 1 PO type is relevance for GTT.
 - If it's CREATING PO, always flag TRUE.
 - If it's UPDATING PO, check whether there is any change for all the above fields or not, comparing their NEW / OLD value pairs.
 - Don't need to consider DELETING PO, which will be considered by standard logic of EM framework and extractors cannot impact this case.
- Method Implementations:** Includes implementations for *lif_bo_reader~get_data*, *lif_bo_reader~get_data_old*, and *lif_bo_reader~get_structures_different*.
- Method Definitions:** Includes definitions for *lif_ef_constants*, *lif_ef_performer*, *lif_bo_processor*, *lcl_bo_reader_di_header*, *lcl_bo_reader_po_header*, and *lcl_bo_reader_po_item*.
- Method Implementations:** Includes implementations for *lif_ef_reader~get_data*, *lif_ef_reader~get_data_old*, and *lif_ef_reader~get_structures_different*.

7: Coding Tips in the Tracking ID Function Modules

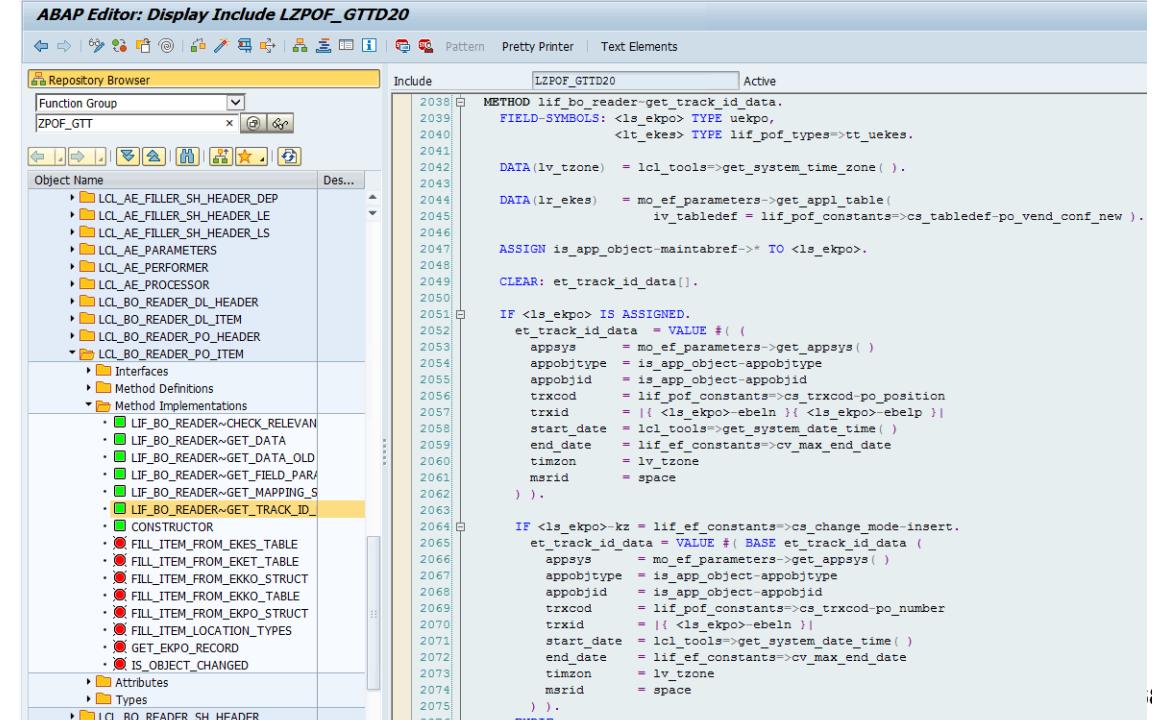
To customize the Tracking ID function modules, key points are as follows:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill in 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 app.
4. GTT v2 accepts delta transport for tracking IDs, which means that only the newly-created / changed / deleted tracking IDs shall be filled in, while the ones without changes need to be ignored in the logic.
5. The tracking ID for its own process type needs to be filled in for each process update.
6. In case of tracking ID deletion, the field ACTION shall be filled in with 'D'.

See sample code of function: *ZPOF_GTT_OTE_PO_ITEM_TID*



The screenshot shows the SAP ABAP Function Builder interface with the function module *ZPOF_GTT_OTE_PO_ITEM_TID* selected. The code implements a *TRY..CATCH..ENDTRY* block. Inside the *TRY* block, it calls *lcl_ef_performer->get_track_id_data* with *EXPORTING* parameters *is_definition* (set to *VALUE #1*) and *et_track_id_data* (set to *e_trackiddata[]*). It then handles errors by catching *cx_udm_message* and logging them. After the *TRY* block, there is a *CASE* statement based on *lo_udm_message->textid*. The *ENDCASE* block ends the function module.



The screenshot shows the SAP ABAP ABAP Editor interface with the include *LZPOF_GTTD20* selected. The code defines a method *lif_bo_reader->get_track_id_data* with *FIELD-SYMBOLS* *lt_ekpo* and *lt_ekes*. It uses *lif_pof_constants->cs_tableddef-po_vend_conf_new* to initialize *DATA* variables *lv_timezone* and *lr_ekes*. It then performs assignments and clears *et_track_id_data*. The method also handles *IS_EKPO* assignment and calls *lif_bo_reader->get_mapping_s* and *lif_bo_reader->get_track_id*. Finally, it checks for changes using *lif_ef_constants->cs_change_mode-insert* and updates *et_track_id_data* accordingly.

8: Coding Tips in the Control Parameter Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill in 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 in the composition (table) fields defined in Manage Models app, use the parameter field *PARAMINDEX* to specify the line number. If the field is empty, GTT regards it as a simple flat field.
5. To clear a composition, fill in the key field using invalid values, for which key attribute has been checked in Manage Models app. It's not recommended to fill in a code list type field to clear a composition even if it's a key field.
6. The fields with fixed names 'ACTUAL_BUSINESS_DATETIME' and 'ACTUAL_BUSINESS_TIMEZONE' are mandatory fields to be transported for event handling sequencing in GTT Version 2.
7. The fields with fixed names 'ACTUAL_TECHNICAL_TIMEZONE' and 'ACTUAL_TECHNICAL_DATETIME' are optional and recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in GTT via TP request, which leads to an error)
8. In Manage Models app, click tab *IDOC Integration* to map the parameter names and model field names.
9. 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.
10. For amount field which has reference currency, ensure to call BAPI 'BAPI_CURRENCY_CONV_TO_EXTERNAL' using the reference currency to make the amount tracked correctly by GTT Version 2. 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 less than 4 decimals.
11. In the shipment extractor, add the prefix LBN# into the fields SERVICE AGENT LBN ID for integration with Visibility Providers.

See sample code of function: *ZPOF_GTT_OTE_PO_ITEM*

8: Coding Tips in the Control Parameter Function Modules

Fields mapping is set up in the Manage Models app in the IDOC Integration section:

pof Active

Purchase Order Fulfillment

Namespace: com.lbngttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List **IDOC Integration** Visibility Provider Integration Planned Event Extension Event to Action

Tracked Process: PurchaseOrder Integration Switch:

Tracked Process Mapping

ERP Object Type: Others Application Object Type: ZPOF_GTT_AC_HD

Tracked Process / Events (1)

Name	IDOC	Event Code
Tracked Process		
PurchaseOrderEvent	E1EHPAO	

Fields

Field	IDOC Segment	IDOC Field
purchaseOrderNo	E1EHPCP	YN_PO_NUMBER
supplierId	E1EHPCP	YN_PO_SUPPLIER_ID
plannedDeliveryDate	E1EHPCP	YN_PO_DELIVERY_DATE
netValue	E1EHPCP	YN_PO_NET_VALUE
currency	E1EHPCP	YN_PO_CURRENCY
incotermsVersion	E1EHPCP	YN_PO_INCOTERMS_VERSION
incoterms	E1EHPCP	YN_PO_INCOTERMS
incotermsLocation	E1EHPCP	YN_PO_INCOTERMS_LOCATION

8: Coding Tips in the Control Parameter Function Modules

Main logic of Purchase Order Item is implemented in class LCL_BO_READER_PO_ITEM

The screenshot shows the SAP ABAP Editor interface with two main panes:

- Function Module:** This pane displays the source code for the function module ZPOF_GTT_OTE_PO_ITEM. The code implements logic for reading purchase order items, including handling control data, performing lookups, and catching errors.
- ABAP Editor: Display Include LZPOF_GTTD20:** This pane shows the implementation of the LCL_BO_READER_PO_ITEM class. It includes several methods for reading data from various sources like EKKO, EKPO, EKET, and EKES tables, as well as methods for getting field mappings and track IDs.

```
Function Module ZPOF_GTT_OTE_PO_ITEM active
Attributes Import Export Changing Tables Exceptions Source Code

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

TRY.
  lcl_ef_performer->get_control_data(
    EXPORTING
      is_definition      = VALUE #(
        maintab          = lif_pof_constants->cs_tabledef-po_item_new
        mastertab         = lif_pof_constants->cs_tabledef-po_header_new )
    io_bo_factory        = NEW lcl_factory_po_item( )
    iv_appsps           = i_appsps
    is_app_obj_types    = i_app_obj_types
    it_all_appl_tables  = i_all_appl_tables
    it_app_type_cntl_tabs = i_app_type_cntl_tabs
    it_app_objects       = i_app_objects
  CHANGING
    ct_control_data     = e_control_data[] ).

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

    " add error message
    APPEND ls_bapiret TO e_logtable.

    " throw corresponding exception
    CASE lo_udm_message->textid.
      WHEN lif_ef_constants->cs_errors_stop_processing.
        RAISE stop_processing.
      WHEN lif_ef_constants->cs_errors_table_determination.
        RAISE table_determination_error.
    ENDCASE.
  ENDTRY.
ENDFUNCTION.

ABAP Editor: Display Include LZPOF_GTTD20
Repository Browser
Function Group ZPOF_GTT
Object Name LCL_BO_READER_PO_ITEM
Method Implementations
  - LIF_BO_READER~CHECK_RELEVANT
  - LIF_BO_READER~GET_DATA
  - LIF_BO_READER~GET_DATA_OLE
  - LIF_BO_READER~GET_FIELD_PAR
  - LIF_BO_READER~GET_MAPPING_
  - LIF_BO_READER~GET_TRACK_ID_
  - CONSTRUCTOR
  - FILL_ITEM_FROM_EKES_TABLE
  - FILL_ITEM_FROM_EKET_TABLE
  - FILL_ITEM_FROM_EKKO_STRUCT
  - FILL_ITEM_FROM_EKPO_TABLE
  - FILL_ITEM_LOCATION_TYPES
  - GET_EKPO_RECORD
  - OBJECT_CHANGED
  Attributes
  Types
  LCL_BO_READER_SH_HEADER
  LCL_CTP_SENDER
  LCL_CTP_SENDER_DL_TO_PO_ITEM
  LCL_CTP_SENDER_SH_TO_DL_HEAD
  LCL_CTP_SENDER_SH_TO_DL_ITEM
  LCL_CTP_SHIPMENT_DATA
Include LZPOF_GTTD20 Active
METHOD lif_bo_reader~get_data.
  FIELD-SYMBOLS: <ls_item>      TYPE ts_po_item.
  rr_data   = NEW ts_po_item( ).

  ASSIGN rr_data->* TO <ls_item>.

  fill_item_from_ekko_struct(
    EXPORTING
      ir_ekko      = is_app_object-mastertabref
    CHANGING
      cs_po_item   = <ls_item> ).

  fill_item_from_ekpo_struct(
    EXPORTING
      ir_ekpo      = is_app_object-maintabref
    CHANGING
      cs_po_item   = <ls_item> ).

  fill_item_from_eket_table(
    EXPORTING
      ir_ekpo      = is_app_object-maintabref
      ir_eket      = mo_ef_parameters->get_appl_table(
        iv_tabledef = lif_pof_constants->cs_tabledef-po_sched_new )
    CHANGING
      cs_po_item   = <ls_item> ).

  fill_item_from_ekes_table(
    EXPORTING
      ir_ekpo      = is_app_object-maintabref
      ir_ekes      = mo_ef_parameters->get_appl_table(
        iv_tabledef = lif_pof_constants->cs_tabledef-po_vend_conf_new )
    CHANGING
      cs_po_item   = <ls_item> ).

  fill_item_location_types(
    CHANGING
      cs_po_item   = <ls_item> ).

ENDMETHOD.
```

9: Coding Tips in the Planned Event Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill in the output table *E_EXPEVENTDATA*.
3. As default except no change made on the model configuration, GTT Version 2 will ask 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. If nothing is transported, the planned events will be removed in GTT Version 2.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT_EXP_DATETIME* is optional, but needs to be filled in 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 in with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* app in GTT Version 2.
7. The field *LOC_ID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function: *ZPOF_GTT_EE_PO_ITEM*

pof Active

Purchase Order Fulfillment

Namespace: com.lbnrgttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List **IDOC Integration** Visibility Provider

Tracked Process: PurchaseOrderItem

Tracked Process Mapping

ERP Object Type: Others

Name	IDOC	Event Code
Tracked Process		
PurchaseOrderItemEvent	E1EHPAO	

Event Types		
ConfirmationEvent	E1EVMHDR02	CONFIRMATION
GoodsReceipt	E1EVMHDR02	GOODS_RECEIPT
DeletionEvent	E1EVMHDR02	DELETION
UndeletionEvent	E1EVMHDR02	UNDELETION

9: Coding Tips in the Planned Event Function Modules

Main logic of Purchase Order Item Planned Events is implemented in class LCL_PE_FILLER_PO_ITEM

```
Function Builder: Display ZPOF_GTT_EE_PO_ITEM
Function Module ZPOF_GTT_EE_PO_ITEM active
Attributes Import Export Changing Tables Exceptions Source Code

21 DATA: lo_udm_message    TYPE REF TO cx_udm_message,
22      ls_bapiret     TYPE bapiret2.
23
24 CLEAR e_logtable[].
25
26 TRY.
27   lcl_ef_performer->get_planned_events(
28     EXPORTING
29       is_definition      = VALUE #(
30         maintab           = lif_pof_constants->cs_tabledef-po_item_new
31         mastertab          = lif_pof_constants->cs_tabledef-po_header_new )
32     io_factory          = NEW lcl_factory_po_item( )
33     iv_appsyst         = i_appsyst
34     is_app_obj_types   = i_app_obj_types
35     it_all_appl_tables = i_all_appl_tables
36     it_app_type_cntl_tabs = i_app_type_cntl_tabs
37     it_app_objects     = i_app_objects
38   CHANGING
39     ct_expeventdata    = e_expeventdata[]
40     ct_measrmntdata   = e_measrmntdata[]
41     ct_infodata        = e_infodata[]
42   ).
43   CATCH cx_udm_message INTO lo_udm_message.
44   lcl_tools->get_errors_log(
45     EXPORTING
46       io_udm_message = lo_udm_message
47       iv_appsyst    = i_appsyst
48     IMPORTING
49       es_bapiret    = ls_bapiret .
50
51   " add error message
52   APPEND ls_bapiret TO e_logtable.
53
54   " throw corresponding exception
55 CASE lo_udm_message->txid.
56   WHEN lif_ef_constants=>cs_errors-stop_processing.
57     RAISE stop_processing.
58   WHEN lif_ef_constants=>cs_errors-table_determination.
59     RAISE table_determination_error.
60 ENDCASE.
61
62 ENTRY.
63 ENDFUNCTION.
```



```
ABAP Editor: Display Include LZPOF_GTTD30
Include LZPOF_GTTD30 Active

425 ENDIF.
426 ENDMETHOD.
427
428 METHOD lif_pe_filler~get_planed_events.
429   add_confirmation_event(
430     EXPORTING
431       is_app_objects = is_app_objects
432     CHANGING
433       ct_expeventdata = ct_expeventdata .
434
435   add_goods_receipt_event(
436     EXPORTING
437       is_app_objects = is_app_objects
438     CHANGING
439       ct_expeventdata = ct_expeventdata .
440
441 ENDMETHOD.
442
443 ENDCCLASS.
```

10: Coding Tips in the Event Data Function Modules

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

1. Make sure that the Main / Master tables follow the configuration of corresponding Event Type.
2. Add customization logic to fill in the output table *CT_TRACKINGHEADER*, *CT_TRACKLOCATION*, *C_EVENTID_MAP*.
3. If the event has user-defined fields in the *Manage Models* app, fill in the table *CT_TRACKPARAMETERS*.
4. Add two technical parameters with fixed names ‘*ACTUAL_TECHNICAL_TIMEZONE*’ and ‘*ACTUAL_TECHNICAL_DATETIME*’ which are recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in GTT via TP request, which leads to an error)
5. If the event has reference table information, fill in the table *CT_TRACKREFERENCES*.
6. The field *CT_TRACKINGHEADER-SRCCOD*, *SRCID*, *SRCTX* is used for event reason transport.
7. In *Manage Models* app, click tab *IDOC Integration* to map the user-defined parameter names and model field names.

See sample code of function: *ZPOF_GTT_EE_PO_ITEM_CONF*

10: Coding Tips in the Event Data Function Modules

To set up mapping of event type user-defined parameters, go to the *IDOC Integration* section of *Manage Models* app, select corresponding event type and set values of IDOC Field:

pof Active

Purchase Order Fulfillment

Namespace: com.lbngttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List **IDOC Integration** Visibility Provider Integration Planned Event Extension Event to Action

Tracked Process: PurchaseOrderItem ▼ Integration Switch: ON

Tracked Process Mapping

ERP Object Type: Others Application Object Type: ZPOF_GTT_AC_ITEM

Tracked Process / Events (5)

Name	IDOC	Event Code
Tracked Process		
PurchaseOrderItemEvent	E1EHPAO	
Event Types		
ConfirmationEvent	E1EVMDR02	CONFIRMATION
GoodsReceipt	E1EVMDR02	GOODS_RECEIPT
DeletionEvent	E1EVMDR02	DELETION
UndeletionEvent	E1EVMDR02	UNDELETION

Fields

Field	IDOC Segment	IDOC Field
quantity	E1EVMPAR	QUANTITY
confirmType	E1EVMPAR	CONFIRM_TYPE

10: Coding Tips in the Event Data Function Modules

Main logic of Purchase Order Item Confirmation event is implemented in class LCL_AE_FILLER_PO_ITEM_CONF

Function Module ZPOF_GTT_EE_PO_ITEM_CONF active

Attributes Import Export Changing Tables Exceptions Source Code

```
59: DATA: lo_udm_message      TYPE REF TO cx_udm_message,
60:       ls_bapiret        TYPE bapiret2.
61:
62: TRY.
63:   lcl_ae_performer->get_event_data(
64:     EXPORTING
65:       is_definition      = VALUE #((
66:         maintab          = lif_pof_constants->cs_tabledef-po_item_new
67:         masterstab        = lif_pof_constants->cs_tabledef-po_header_new )
68:       io_ae_factory      = NEW lcl_ae_factory_po_item_conf( )
69:       iv_appsyst         = i_appsyst
70:       is_event_type      = i_event_type
71:       it_all_appl_tables = i_all_appl_tables
72:       it_event_type_cntl_tabs = i_event_type_cntl_tabs
73:       it_events          = i_events
74:     CHANGING
75:       ct_eventid_map    = c_eventid_map[]
76:       ct_trackingheader = ct_trackingheader[]
77:       ct_tracklocation  = ct_tracklocation[]
78:       ct_trackreferences = ct_trackreferences[]
79:       ct_trackparameters = ct_trackparameters[]
80:   .
81: CATCH cx_udm_message INTO lo_udm_message.
82:   lcl_tools->get_errors_log(
83:     EXPORTING
84:       io_udm_message = lo_udm_message
85:       iv_appsyst     = i_appsyst
86:     IMPORTING
87:       es_bapiret     = ls_bapiret .
88:
89:   " add error message
90:   APPEND ls_bapiret TO ct_logtable.
91:
92:   " throw corresponding exception
93:   CASE lo_udm_message->textid.
94:     WHEN lif_ef_constants->cs_errors-stop_processing.
95:       RAISE stop_processing.
96:     WHEN lif_ef_constants->cs_errors-table_determination.
97:       RAISE event_data_error.
98:   ENDCASE.
99: ENDTRY.
100: ENDFUNCTION.
```

ABAP Editor: Display Include LZPOF_GTTD40

Repository Browser

Function Group ZPOF_GTT

Object Name • ZPOF_GTT_UPDATE_RELEVANCE_TAB

Includes

Classes

- LCL_AE_FACTORY
- LCL_AE_FACTORY_DL_ITEM_GR
- LCL_AE_FACTORY_DL_ITEM_PA
- LCL_AE_FACTORY_DL_ITEM_PKNG
- LCL_AE_FACTORY_PO_ITEM_CONF
- LCL_AE_FACTORY_PO_ITEM_DEL
- LCL_AE_FACTORY_PO_ITEM_GR
- LCL_AE_FACTORY_SH_HEADER_ARR
- LCL_AE_FACTORY_SH_HEADER_CI
- LCL_AE_FACTORY_SH_HEADER_DEP
- LCL_AE_FACTORY_SH_HEADER_LE
- LCL_AE_FACTORY_SH_HEADER_LS
- LCL_AE_FILLER_DL_ITEM_GR
- LCL_AE_FILLER_DL_ITEM_PA
- LCL_AE_FILLER_DL_ITEM_PKNG
- LCL_AE_FILLER_PO_ITEM_CONF

Method Implementations

- LIF_AE_FILLER~CHECK_RELEVANCY
- LIF_AE_FILLER~GET_EVENT_DATA
- CONSTRUCTOR
- GET_CONFIRMATION_QUANTITY
- GET_CONFIRMATION_QUANTITY_
- HAS_CHANGES
- IS_APPROPRIATE_CONF_CONTRO
- IS_APPROPRIATE_CONF_TYPE

Attributes

- LCL_AE_FILLER_PO_ITEM_DEL
- LCL_AE_FILLER_PO_ITEM_GR
- LCL_AE_FILLER_SH_HEADER_ARR
- LCL_AE_FILLER_SH_HEADER_BH

Include LZPOF_GTTD40 Active

```
98: METHOD lif_ae_filler~get_event_data.
99:   DATA(lv_difference) = get_confirmation_quantity_diff(
100:     is_events = is_events ).
101:
102:   ct_trackingheader = VALUE #( BASE ct_trackingheader (
103:     language = sy-langu
104:     trxid = lcl_po_tools->get_tracking_id_po_item(
105:       ir_ekpo = is_events-maintabref )
106:     trxcod = lif_pof_constants->cs_trxcod-po_position
107:     evtcnt = is_events-eventid
108:     evtid = lif_pof_constants->cs_milestone-po_confirmation
109:     evtdat = sy-datum
110:     evttim = sy-uzzeit
111:     evtzon = lcl_tools->get_system_time_zone( )
112:   )).
113:
114:   ct_eventid_map = VALUE #( BASE ct_eventid_map (
115:     eventid = is_events-eventid
116:     evtcnt = is_events-eventid
117:   )).
118:
119:   ct_tracklocation = VALUE #( BASE ct_tracklocation (
120:     evtcnt = is_events-eventid
121:     loccod = lif_ef_constants->cs_loc_types-plant
122:     locidl = lcl_tools->get_field_of_structure(
123:       ir_struct_data = is_events-maintabref
124:       iv_field_name = 'WERKS' )
125:   )).
126:
127:
128:   " QUANTITY
129:   ct_trackparameters = VALUE #( BASE ct_trackparameters (
130:     evtcnt = is_events-eventid
131:     param_name = lif_pof_constants->cs_event_param-quantity
132:     param_value = lcl_tools->get_pretty_value( iv_value = lv_difference )
133:   )).
134:
135:   " CONFIRMATION TYPE
136:   ct_trackparameters = VALUE #( BASE ct_trackparameters (
137:     evtcnt = is_events-eventid
138:     param_name = lif_pof_constants->cs_event_param-confirm_type
139:     param_value = lif_pof_constants->cs_relevance-ebtyp
140:   )).
141:
142: ENDMETHOD.
```

11: Enhancement Codes for Cross-process Tracking

The Track PO Fulfillment template app asks for cross-processes tracking, which is used in cases below:

- When the shipment process is updated and transported to GTT, the preceding inbound delivery and item process, and their planned events needs to be updated and transported to GTT.

IMPORTANT: To enable cross-process tracking, update the sample codes below after downloading:

- Update Inbound Delivery Header and Item AOT type Mask in Method GET_AOTYPE_RESTRICTIONS of LCL_CTP_SENDER_SH_TO_DL_HEAD and LCL_CTP_SENDER_SH_TO_DL_ITEM

```
ABAP Editor: Display Include LZPOF_GTTD80

Include LZPOF_GTTD80 Active

1551 CLASS lcl_ctp_sender_sh_to_dl_head IMPLEMENTATION.
1552   METHOD get_aotype_restrictions.
1553     et_aotype = VALUE #( (
1554       low   = 'ZPOF_GTT_*_DL_HD'
1555       option = 'CP'
1556       sign  = 'I'
1557     ) )
1558   ENDMETHOD.
1559
1560   METHOD get_instance.
1561     DATA(lt_trk_obj_type) = VALUE tt_trk_obj_type(
1562       ( lif_ef_constants=>cs_trk_obj_type-esc_shipmt )
1563       ( lif_ef_constants=>cs_trk_obj_type-esc_deliv )
1564     ).
```

11: Enhancement Codes for Cross-process Tracking

The cross-process tracking scenarios cover the following:

Shipment -> Inbound Delivery and Inbound Delivery Item:

- 1\ Tracking ID (Delta Transport)
 - Case: Shipment Create / Delete with Delivery
 - Case: Shipment Assign / Unassign Delivery
- 2\ Shipment Composition (Full Transport)
 - Case: Shipment Create / Delete with Delivery
 - Case: Shipment Assign / Unassign Delivery
- 3\ Planned Event in Delivery (Full Transport)
 - Case: Shipment Create / Delete with Delivery / with stage
 - Case: Shipment Assign / Unassign Delivery / with stage
 - Case: Stage Assign / Unassign Delivery
 - Case: Stage Insert / Delete
 - Case: Stage Location Update
 - Case: Stage Planned Datetime Update
- 4\ Planned Event in Delivery Item (Full Transport)
 - Case: Shipment Create / Delete with Delivery / with stage
 - Case: Shipment Assign / Unassign Delivery / with stage
 - Case: Stage Assign / Unassign Delivery
 - Case: Stage Insert / Delete
 - Case: Stage Location Update
 - Case: Stage Planned Datetime Update

12: Known Issues

1. Planned Event Extension not enabled

Currently, on the ERP side, the EXTENSION segment of process IDOC is not enabled for the planned event part, which means that you cannot make the user-defined fields for planned events in the Manage Models app.

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

2. IDOC sequencing issue

Currently, on the ERP side, when you report actual events while creating the process, the IDOCs might be sent in an incorrect order. For example, entering a PICK quantity and saving the new delivery in ERP will generate a PICK event IDOC and a delivery order IDOC. If the event IDOC approaches GTT prior to the order IDOC, it will lead to processing failure.

This issue is covered now, see the solution provided in these topics:

- [8: Coding Tips in the Control Parameter Function Modules](#)
- [10: Coding Tips in the Event Data Function Modules](#)
- [13: Solution of IDOC Sequencing Issue](#)

13: Solution of IDOC Sequencing Issue

1. Implement corrections provided in the note <https://launchpad.support.sap.com/#/notes/2959576>

2. Create CI tenant.

Select “**GTT2.0 Logistics Business Network - Track and Trace**” for SAP Track & Trace Version

SAP Global Track & Trace Definitions			
CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTPOFAC2	ZGTTPOFAC	GTT2.0 Logistics Business N...	CI For GTT Purchasing Order Sample APP - Acceptance

3. Create RFC destination

You need to configure only one RFC connection for both event and tracked process.

They have the same **Path Prefix**:
`/api/idoc/em/v1/TrackedProcessAndEvent`

RFC Destination ZGTT_POF_PO_TP_ACC2

Connection Test

RFC Destination	ZGTT_POF_PO_TP_ACC2
Connection Type	G HTTP Connection to External Server
Description	
Description 1	RFC for Tracted Process of POF Sample Application to Acceptance
Description 2	
Description 3	

Administration Technical Settings Logon & Security Special Options

Target System Settings

Host		Port	443
Path Prefix	/api/idoc/em/v1/TrackedProcessAndEvent		

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.