SAP Logistics Business Network, Global Track and Trace Option

Track Purchase Orders - Deep Dive with SAP ERP Integration

Logistics Business Network
December 2020

PUBLIC

THE BEST RUN SAP

Objectives

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

- Learn what prerequisite is necessary for 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 own 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 Product Version for GTT v2 shall be SAP EHP1 FOR SAP NETWEAVER 7.3 or higher.

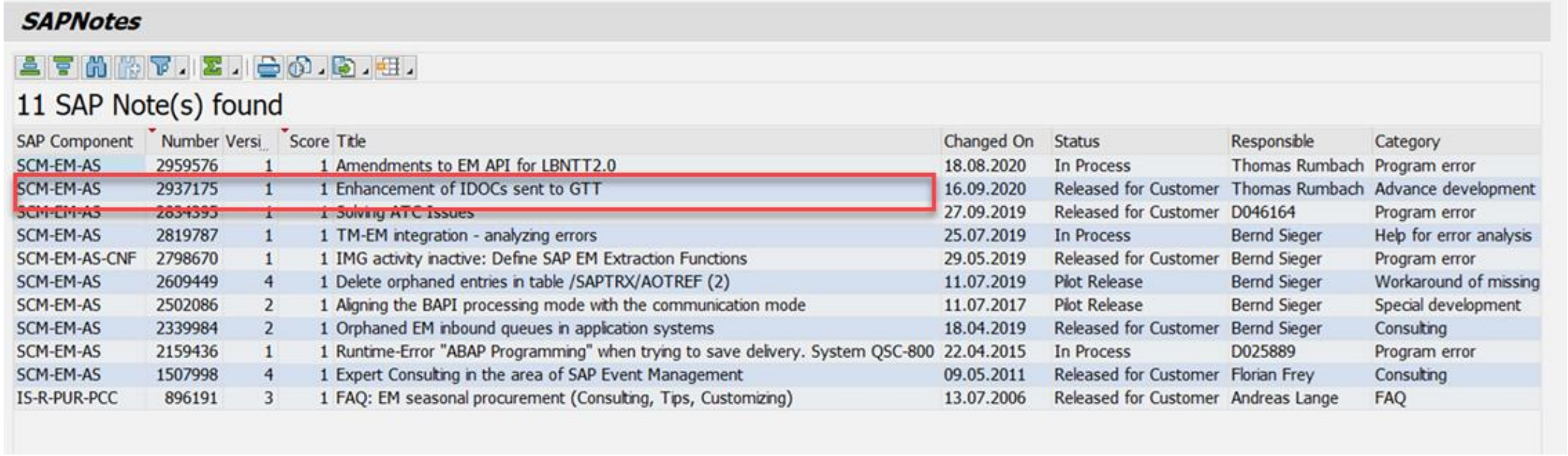
1-2: SAP NOTE 2937175 shall be implemented.

1-3: The ABAP codes to support sample apps for GTT v2 can be implemented in S4 HANA 2101 on premise, 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



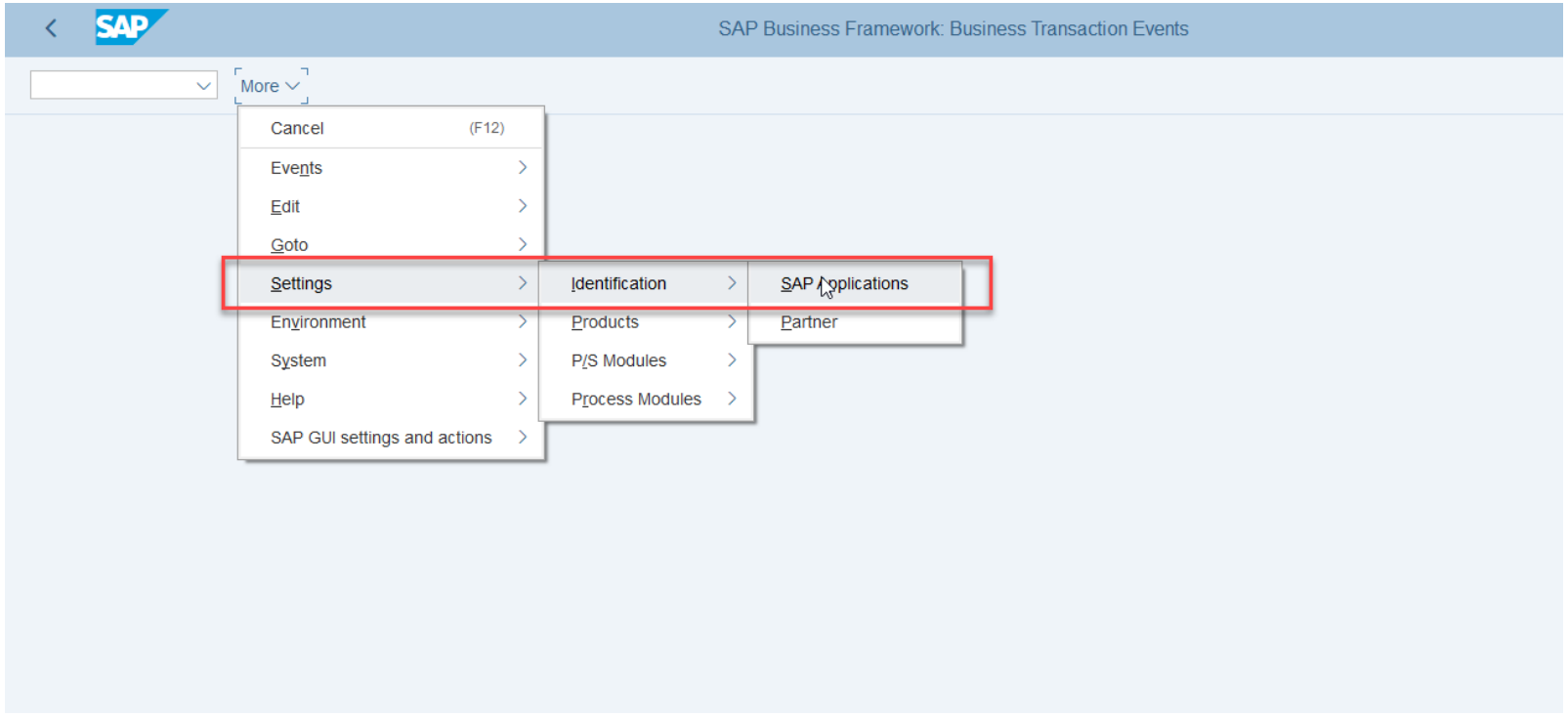
SAP Component	Number	Versi	Score	Title	Changed On	Status	Responsible	Category
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process	Thomas Rumbach	Program error
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer	Thomas Rumbach	Advance development
SCM-EM-AS	2834395	1	1	Solving ATC Issues	27.09.2019	Released for Customer	D046164	Program error
SCM-EM-AS	2819787	1	1	TM-EM integration - analyzing errors	25.07.2019	In Process	Bernd Sieger	Help for error analysis
SCM-EM-AS-CNF	2798670	1	1	IMG activity inactive: Define SAP EM Extraction Functions	29.05.2019	Released for Customer	Bernd Sieger	Program error
SCM-EM-AS	2609449	4	1	Delete orphaned entries in table /SAPTRX/AOTREF (2)	11.07.2019	Pilot Release	Bernd Sieger	Workaround of missing
SCM-EM-AS	2502086	2	1	Aligning the BAPI processing mode with the communication mode	11.07.2017	Pilot Release	Bernd Sieger	Special development
SCM-EM-AS	2339984	2	1	Orphaned EM inbound queues in application systems	18.04.2019	Released for Customer	Bernd Sieger	Consulting
SCM-EM-AS	2159436	1	1	Runtime-Error "ABAP Programming" when trying to save delivery. System QSC-800	22.04.2015	In Process	D025889	Program error
SCM-EM-AS	1507998	4	1	Expert Consulting in the area of SAP Event Management	09.05.2011	Released for Customer	Florian Frey	Consulting
IS-R-PUR-PCC	896191	3	1	FAQ: EM seasonal procurement (Consulting, Tips, Customizing)	13.07.2006	Released for Customer	Andreas Lange	FAQ

STEP 2: Log on the Development Client to Configure BTE

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**

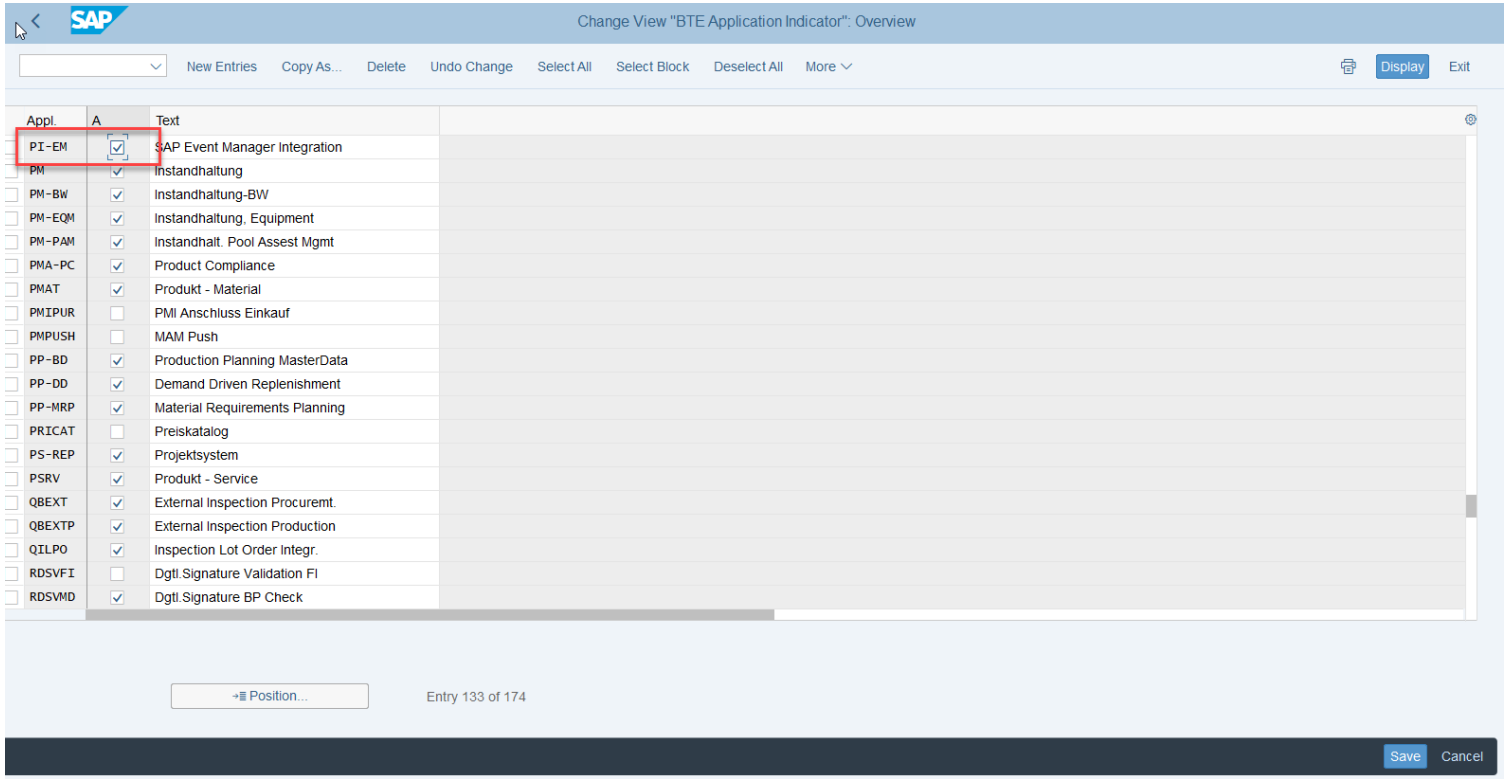


STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**



B) Configuration and Implementation
- Basic

B1. IDOC Configuration

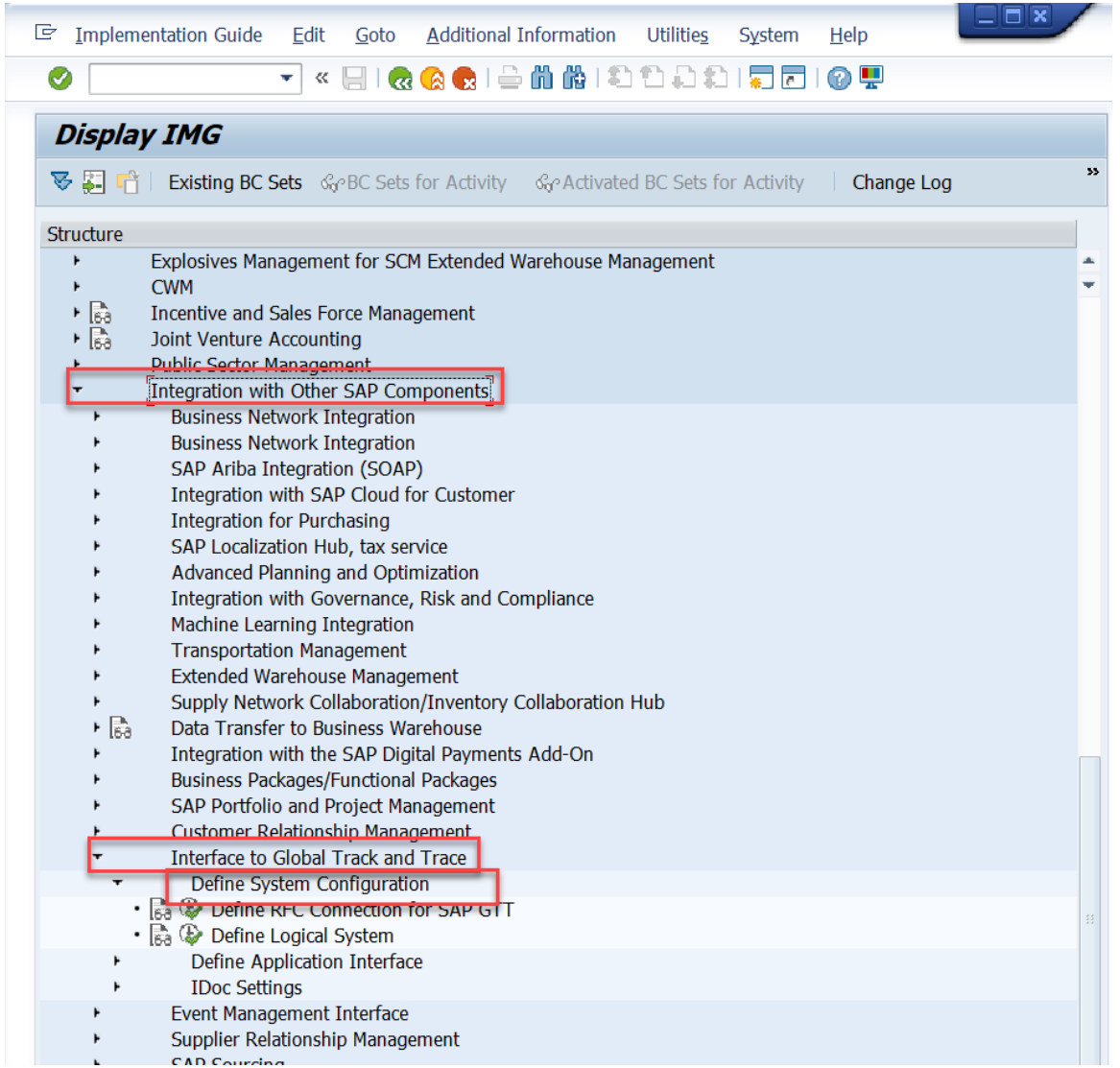
STEP 1: Define RFC Connection for GTT

1-1: Log on to the business client

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

1-3: Click **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **Define System Configuration**

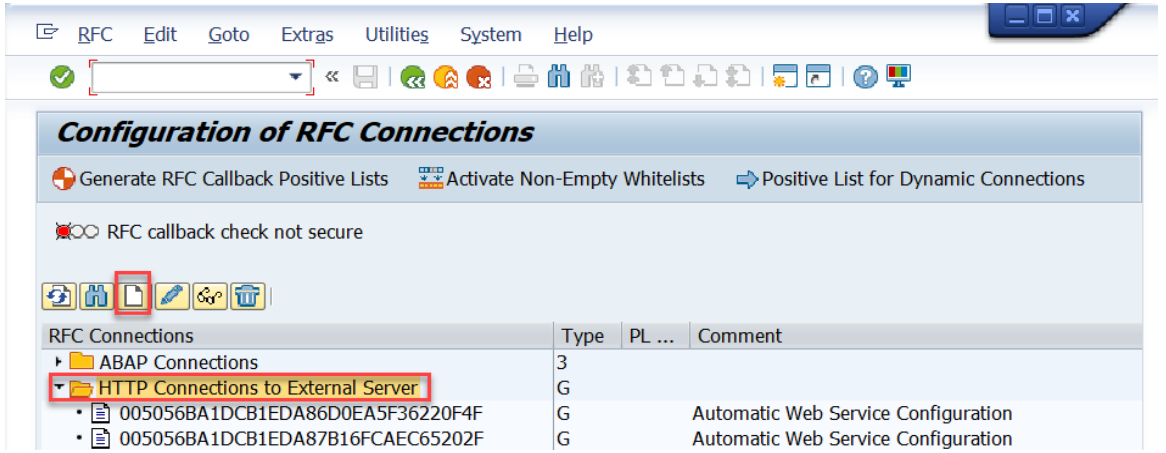
1-4: Choose activity: **Define RFC Connection for SAP GTT**

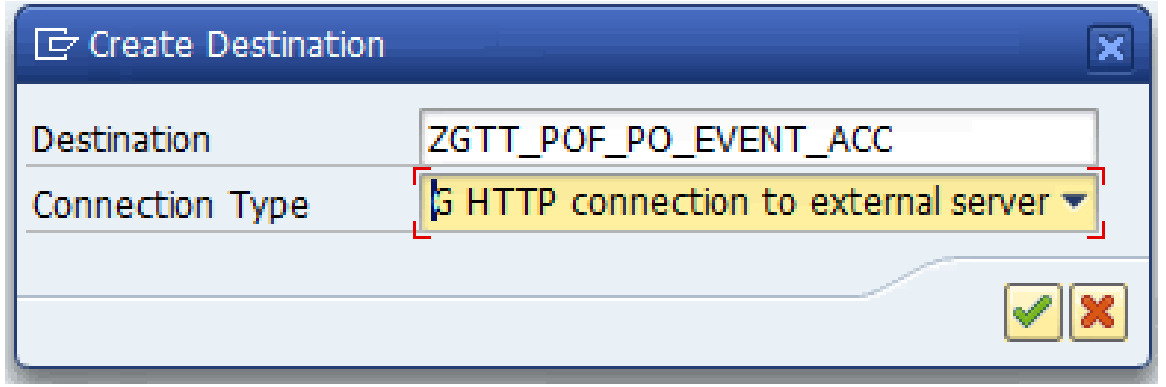


STEP 1: Define RFC Connection for GTT

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

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





STEP 1: Define RFC Connection for GTT

1-7: Enter a description

1-8: In the **Technical Settings** tab, fill in the **Host**, **Port** and **Path Prefix**

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

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

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

Port: *443*

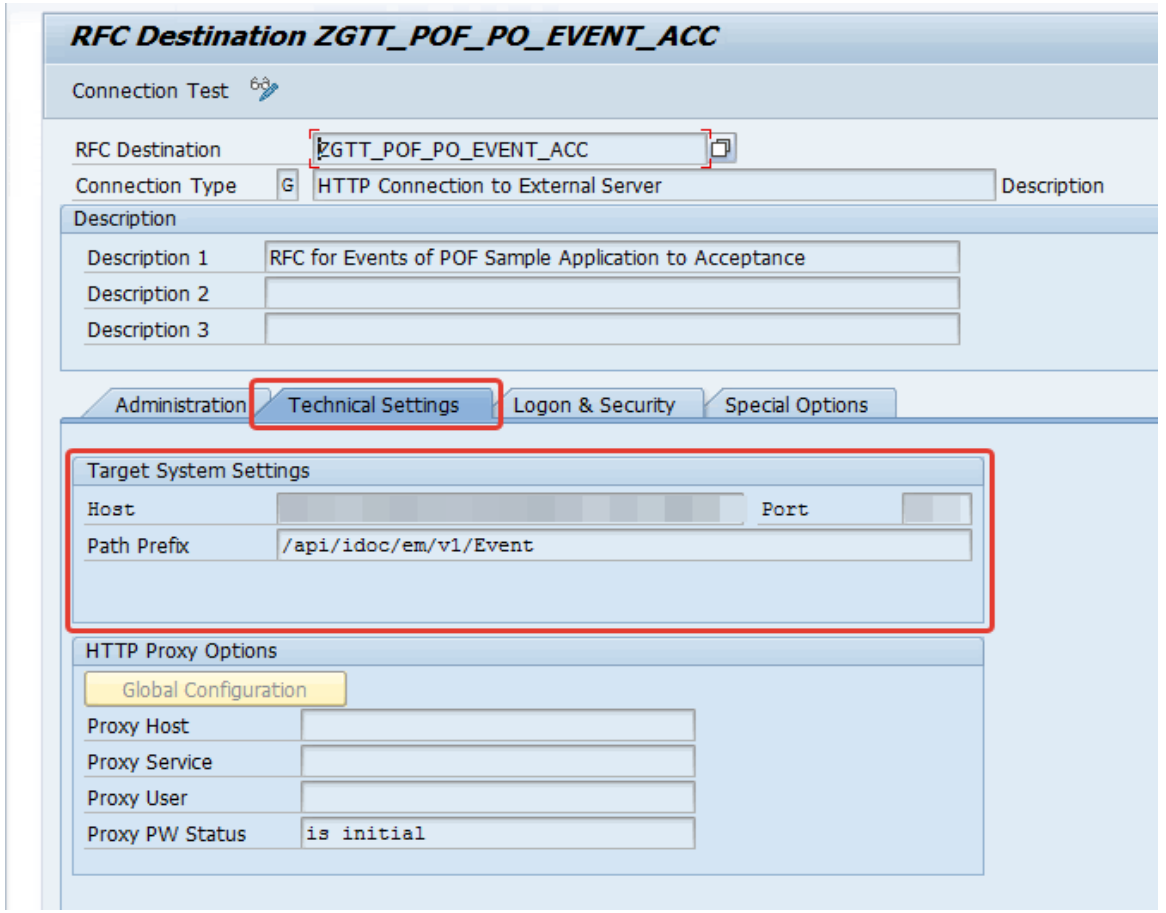
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefix**es.

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.

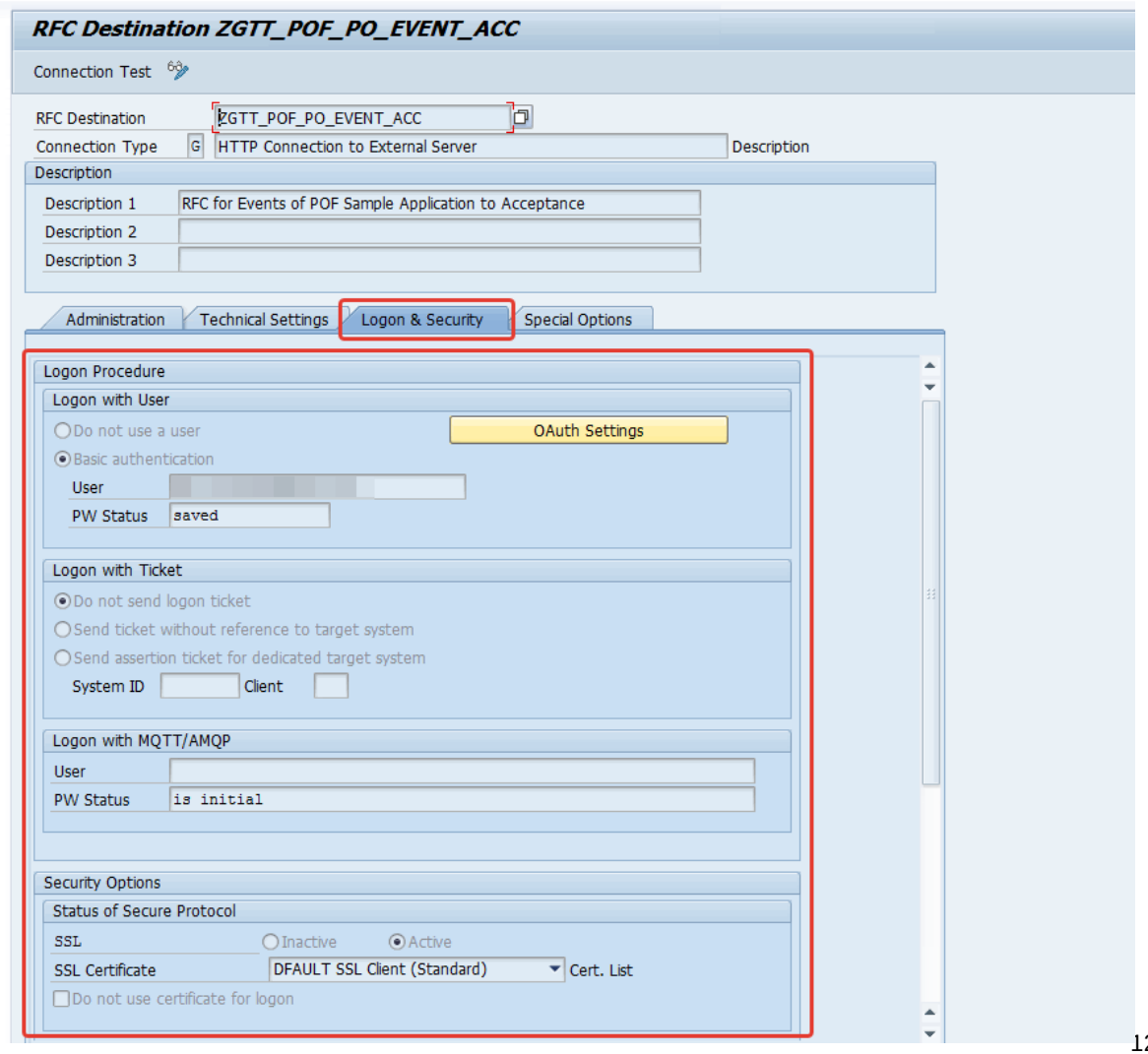
Also, SSL must be *Active.*
The recommended SSL Certificate is: *DFAULT SSL Client (Standard)*.

1-10: Save the configuration

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

Caution: You need to configure two RFC Connections:

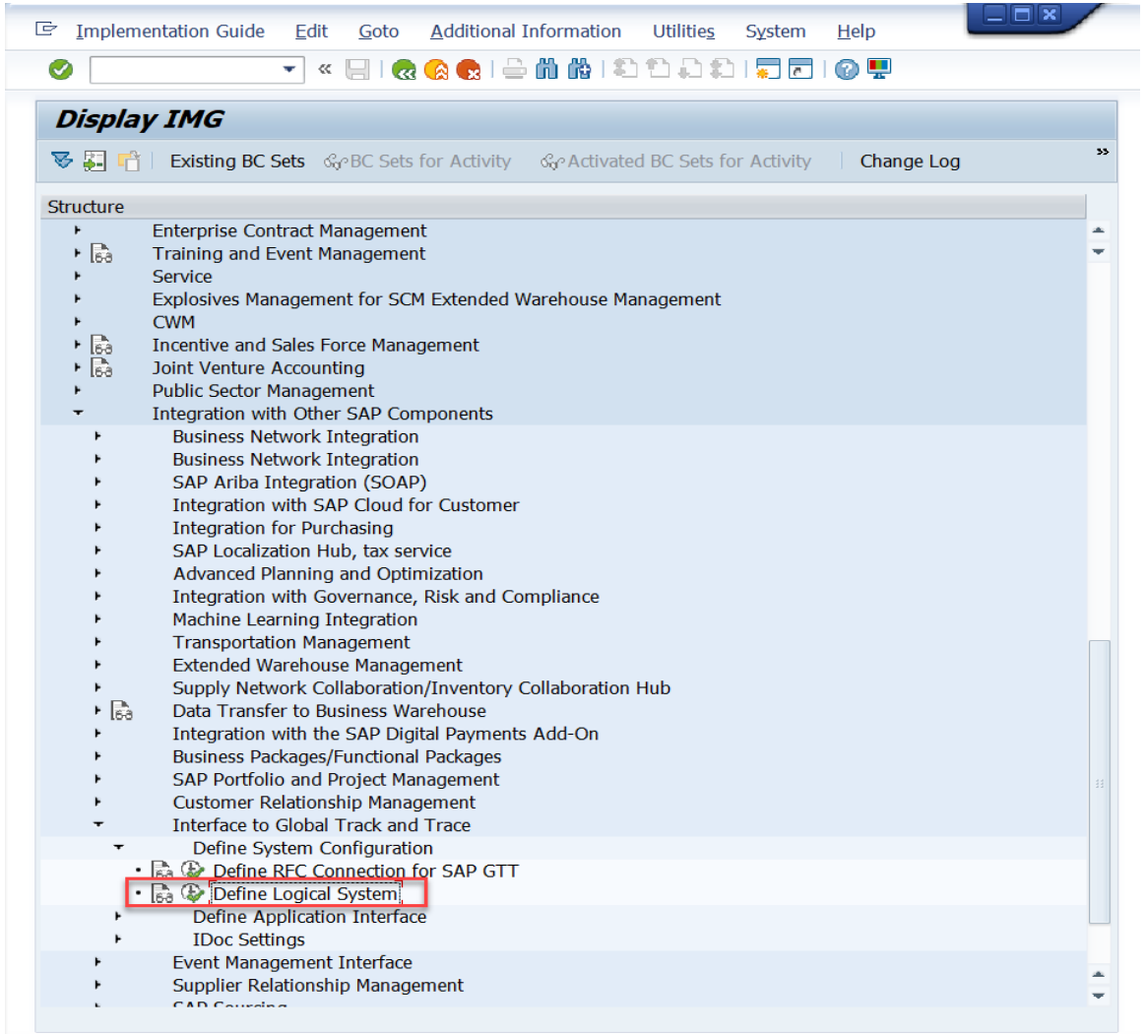
- one for event and
- the other for tracked process.



STEP 2: Define Logical System

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

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

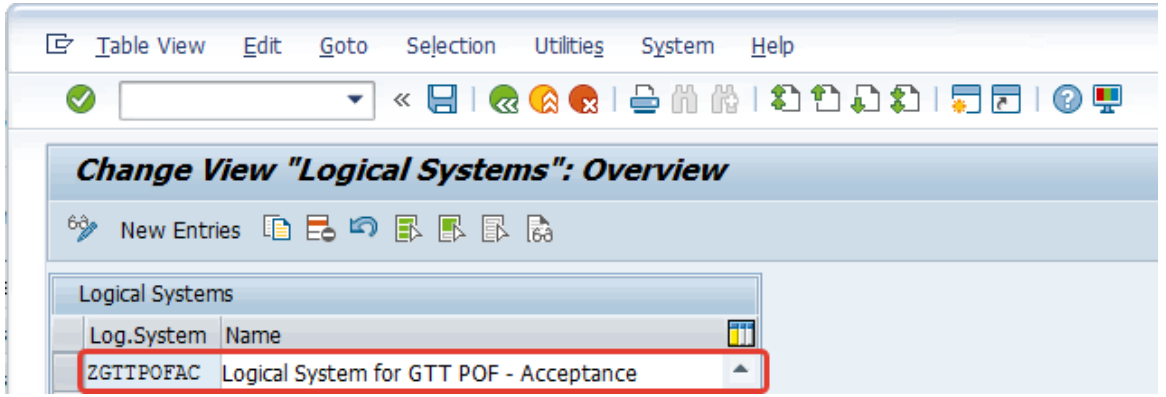


STEP 2: Define Logical System

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

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

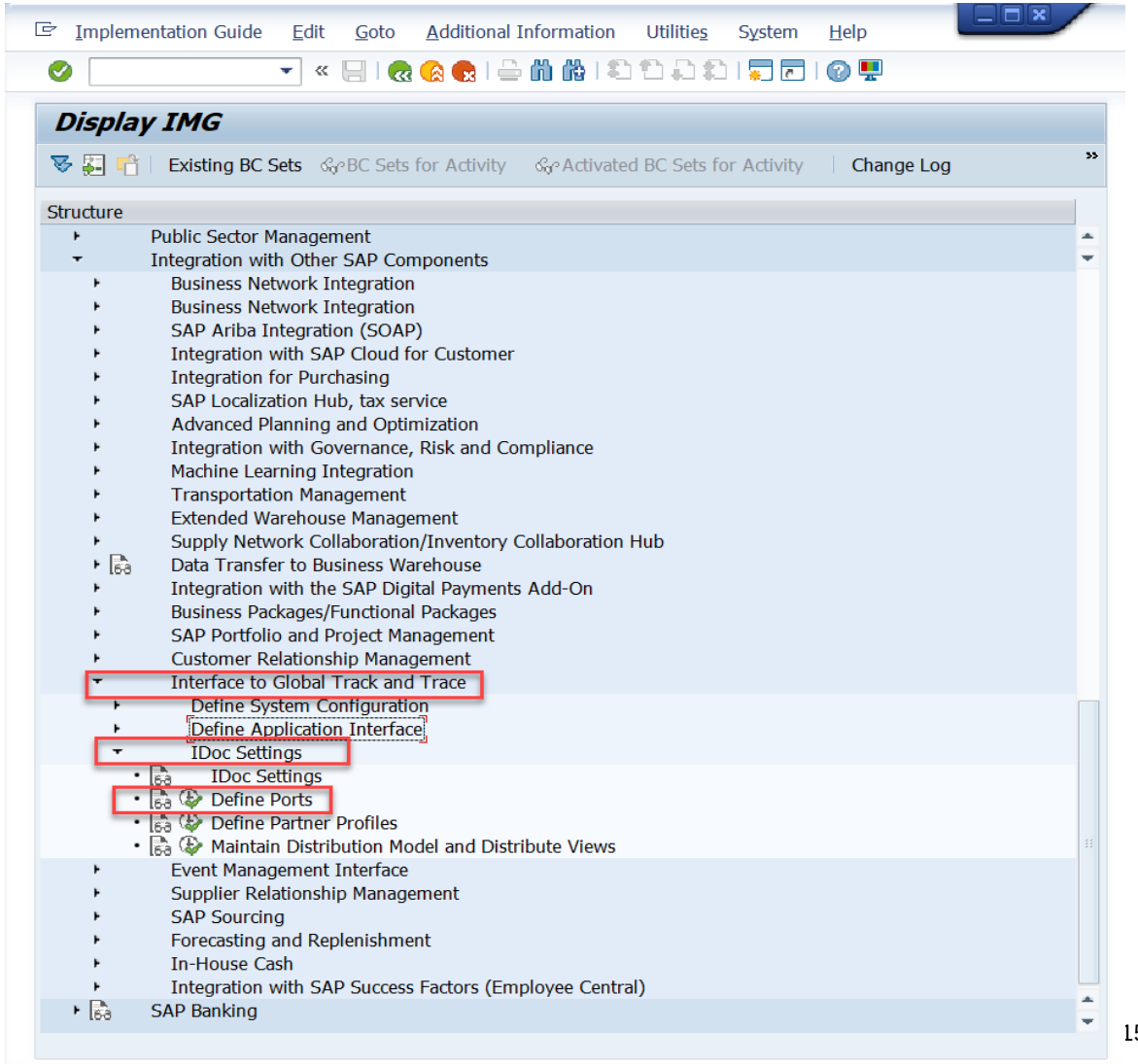
2-4: Save the configuration



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

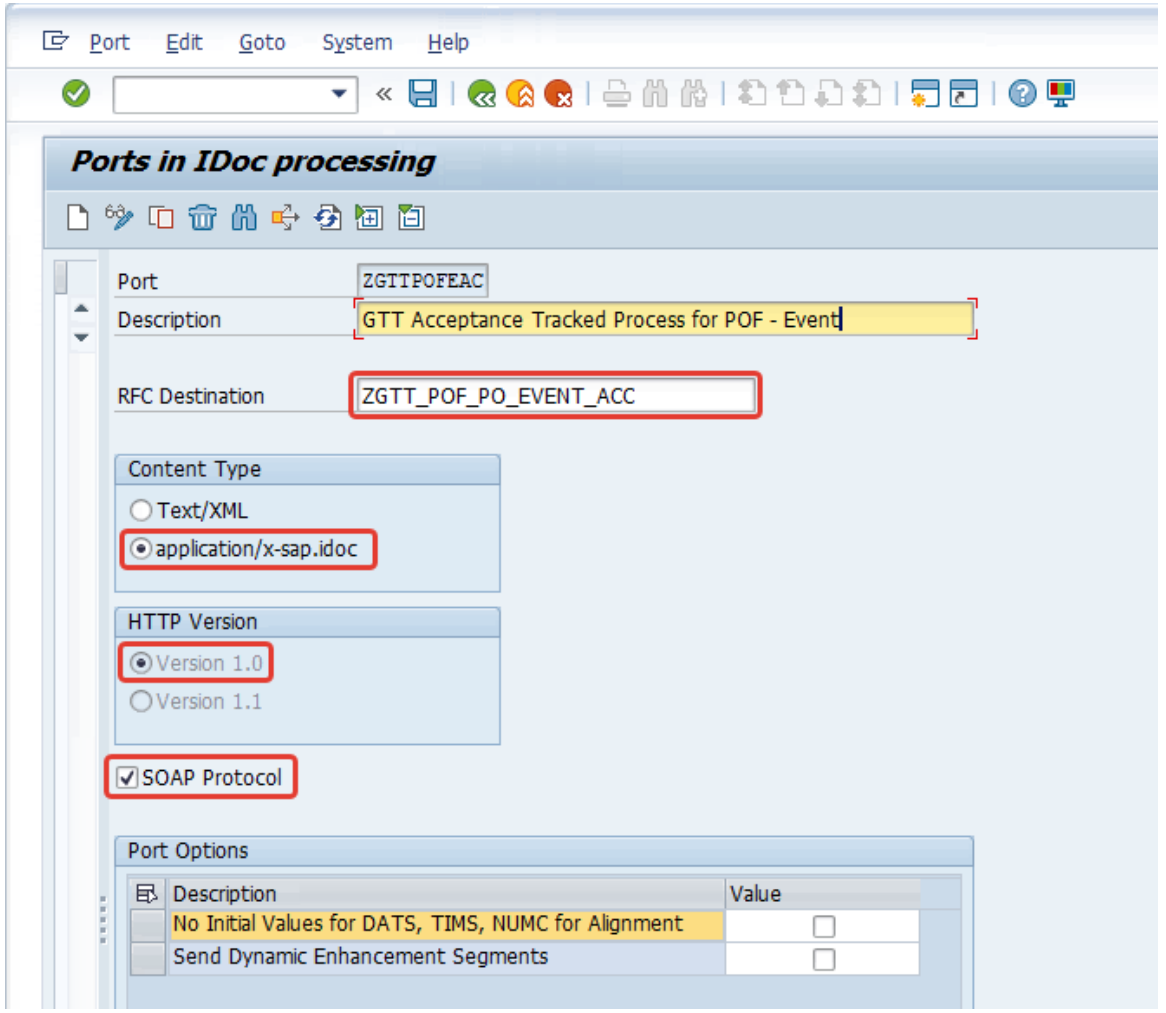
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

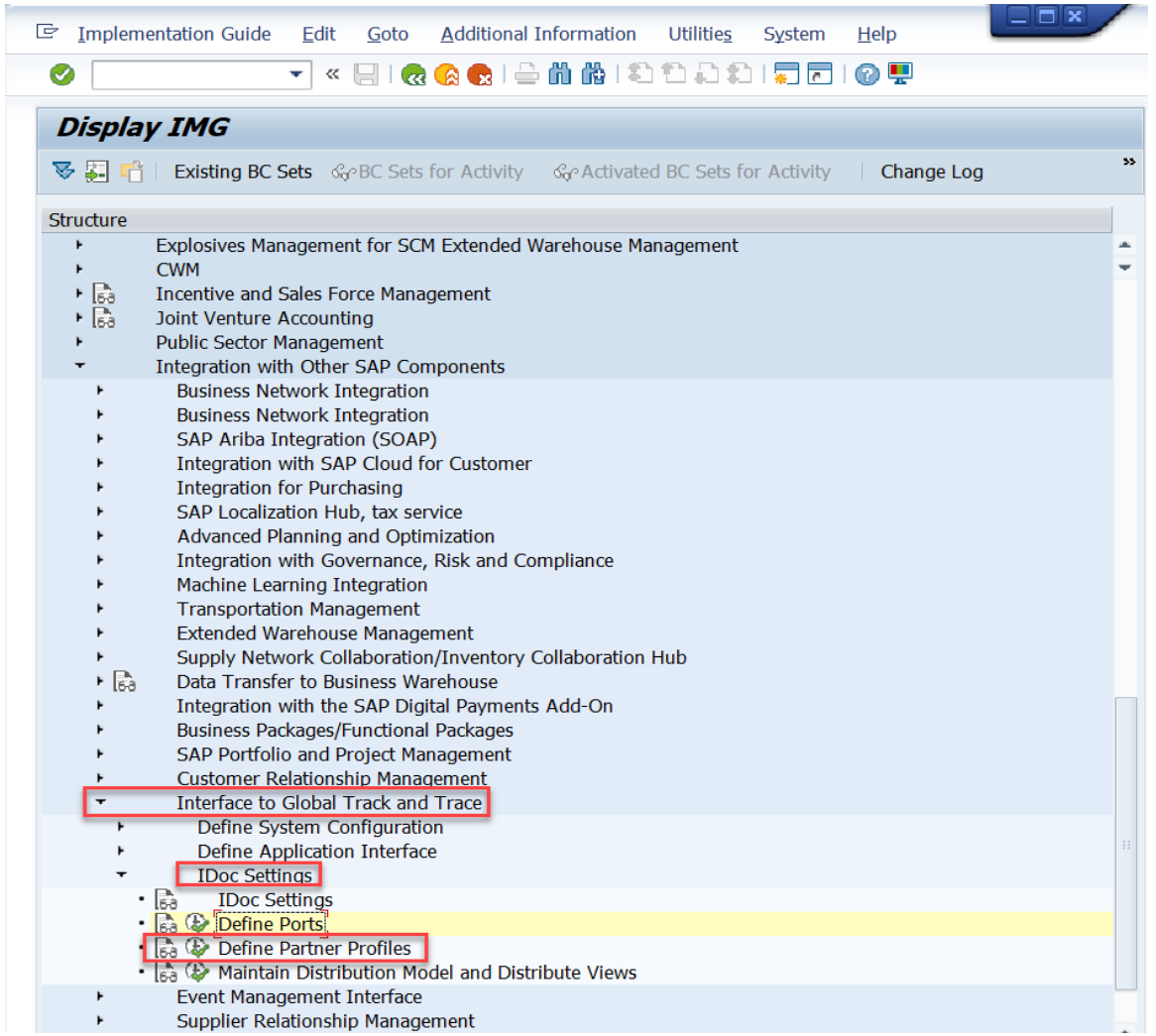
Caution: You need to define two ports, one for event and the other for tracked process.



STEP 4: Define Partner Profiles

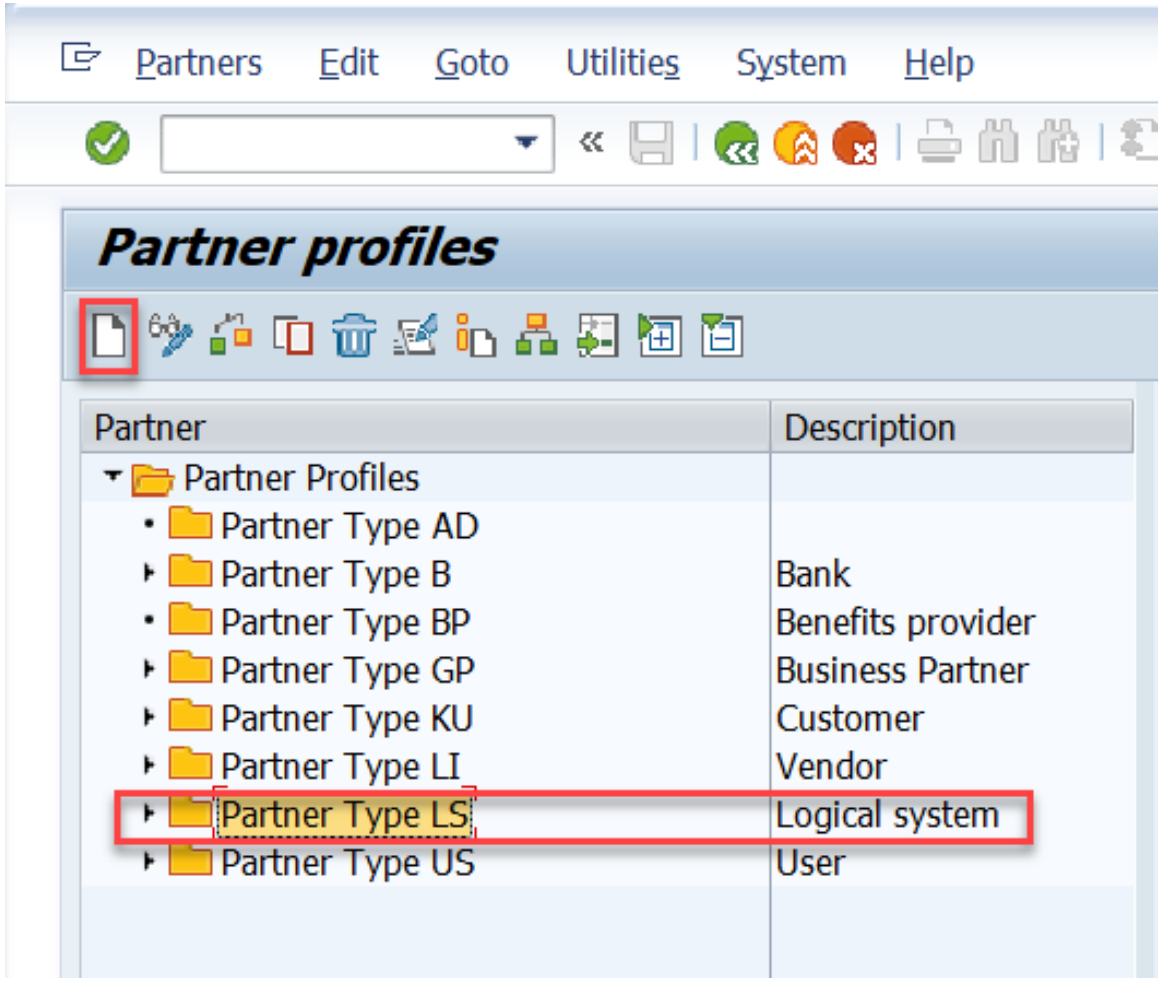
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

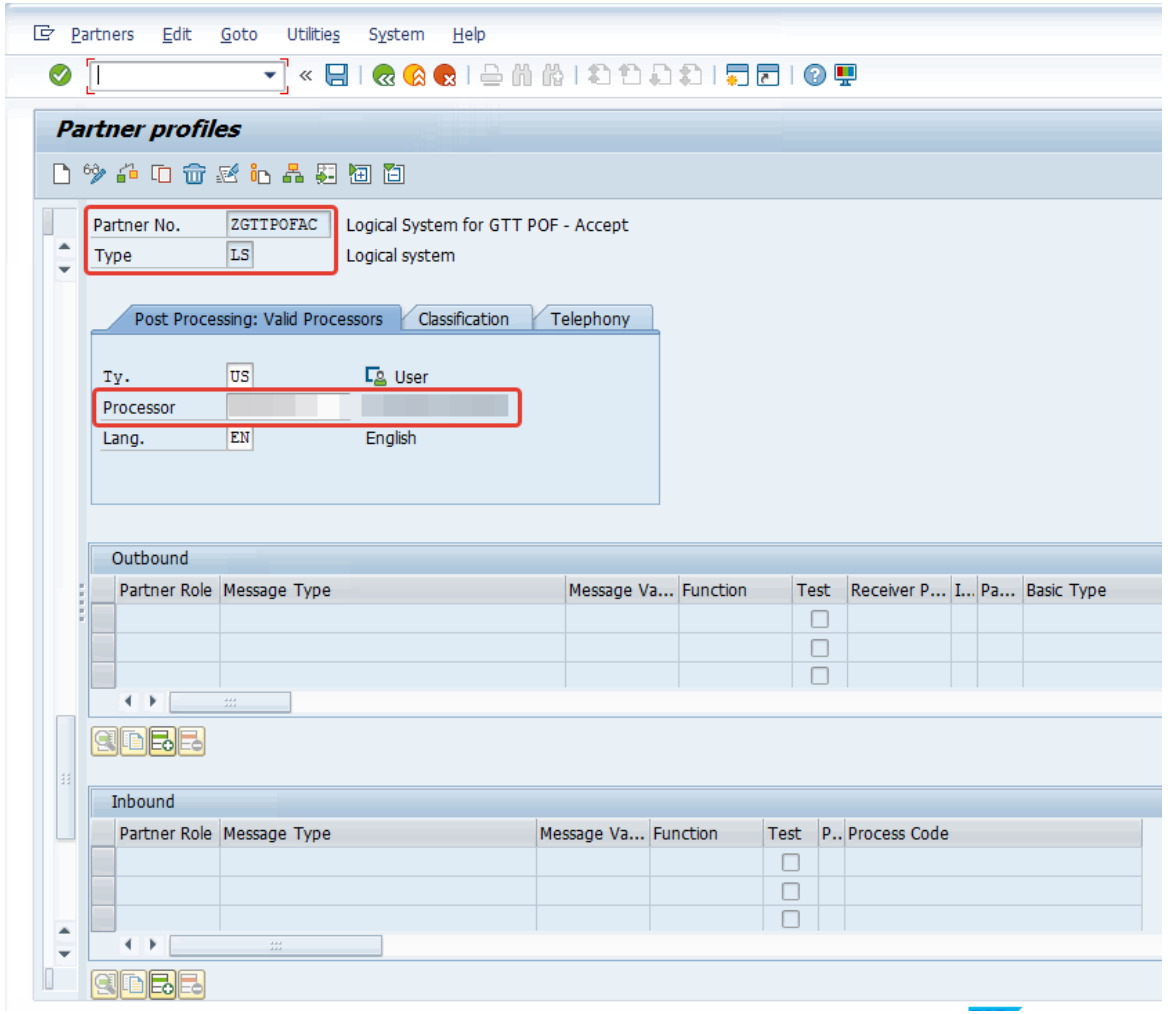
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

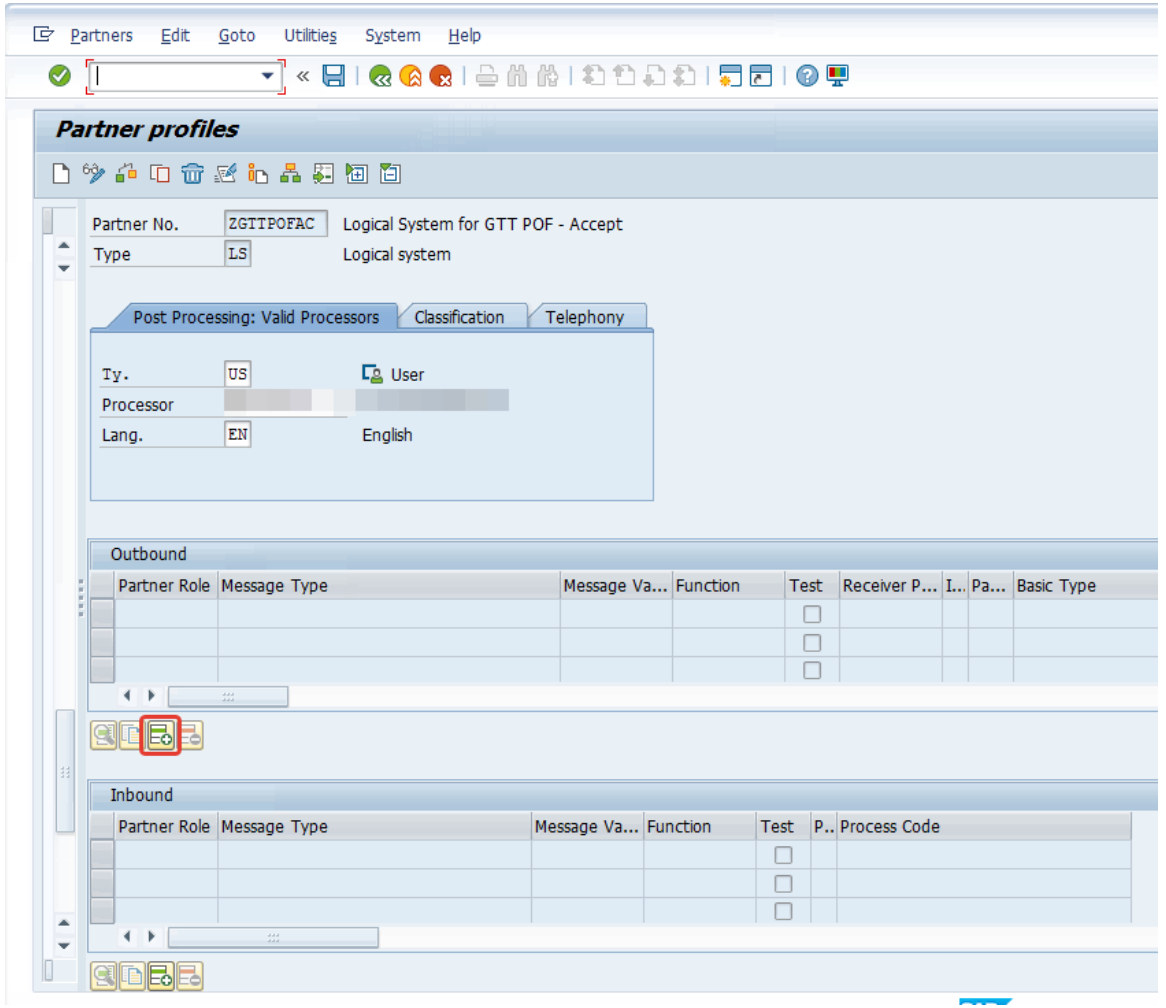
4-4: Fill in the **Partner No.** that you created in STEP 2

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*

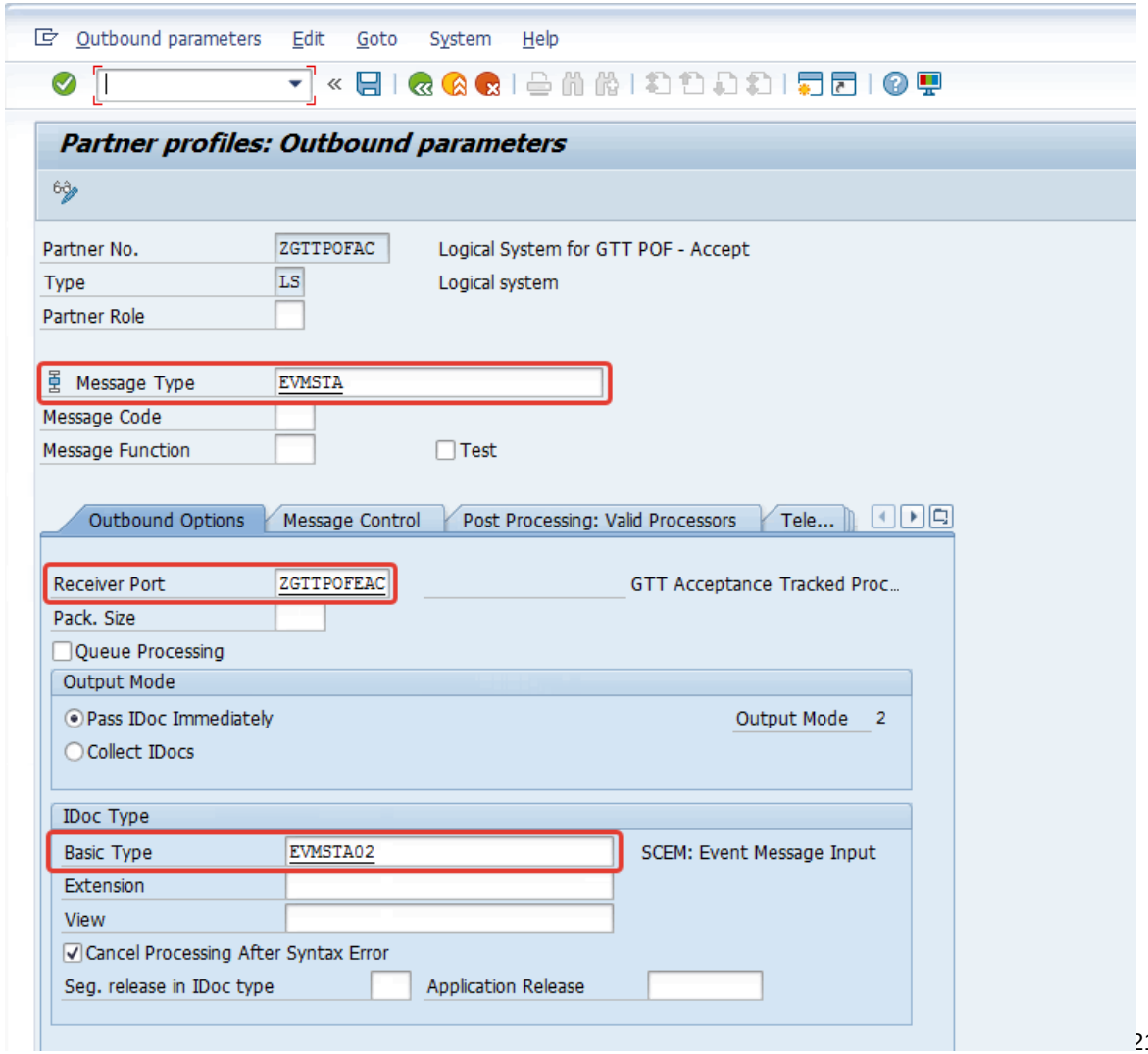
For the tracked Process:

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



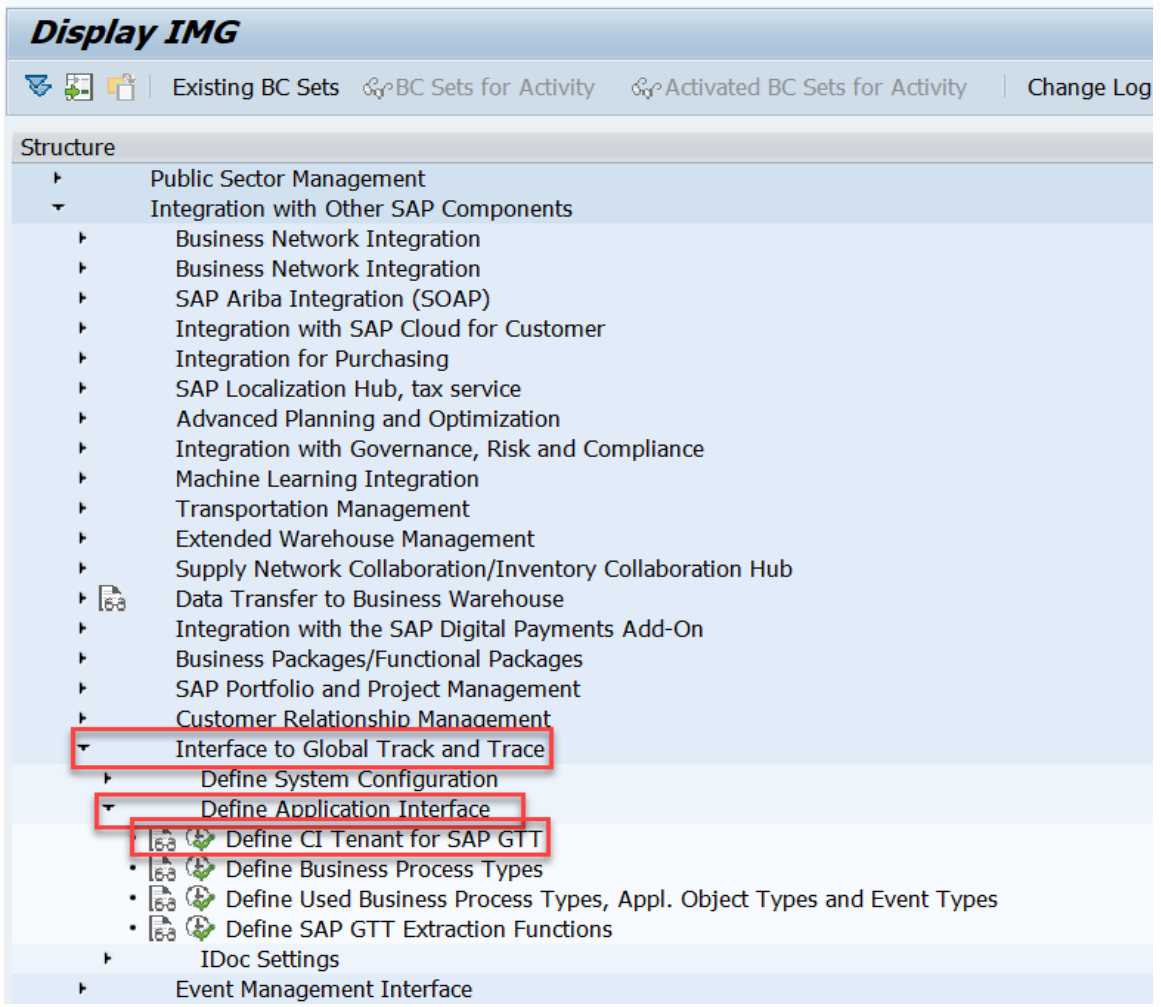
B) Configuration and Implementation - Basic

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

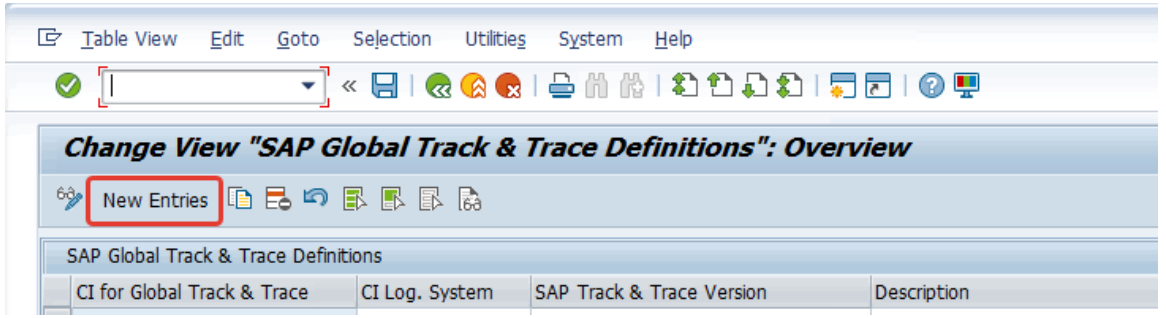
5-2: Choose activity
Define CI Tenant for SAP GTT

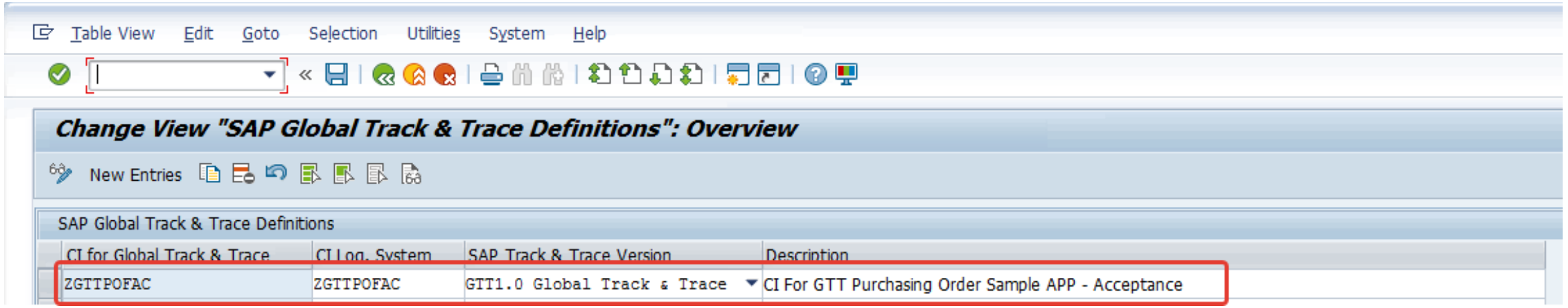


STEP 5: Define CI Tenant for GTT

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.

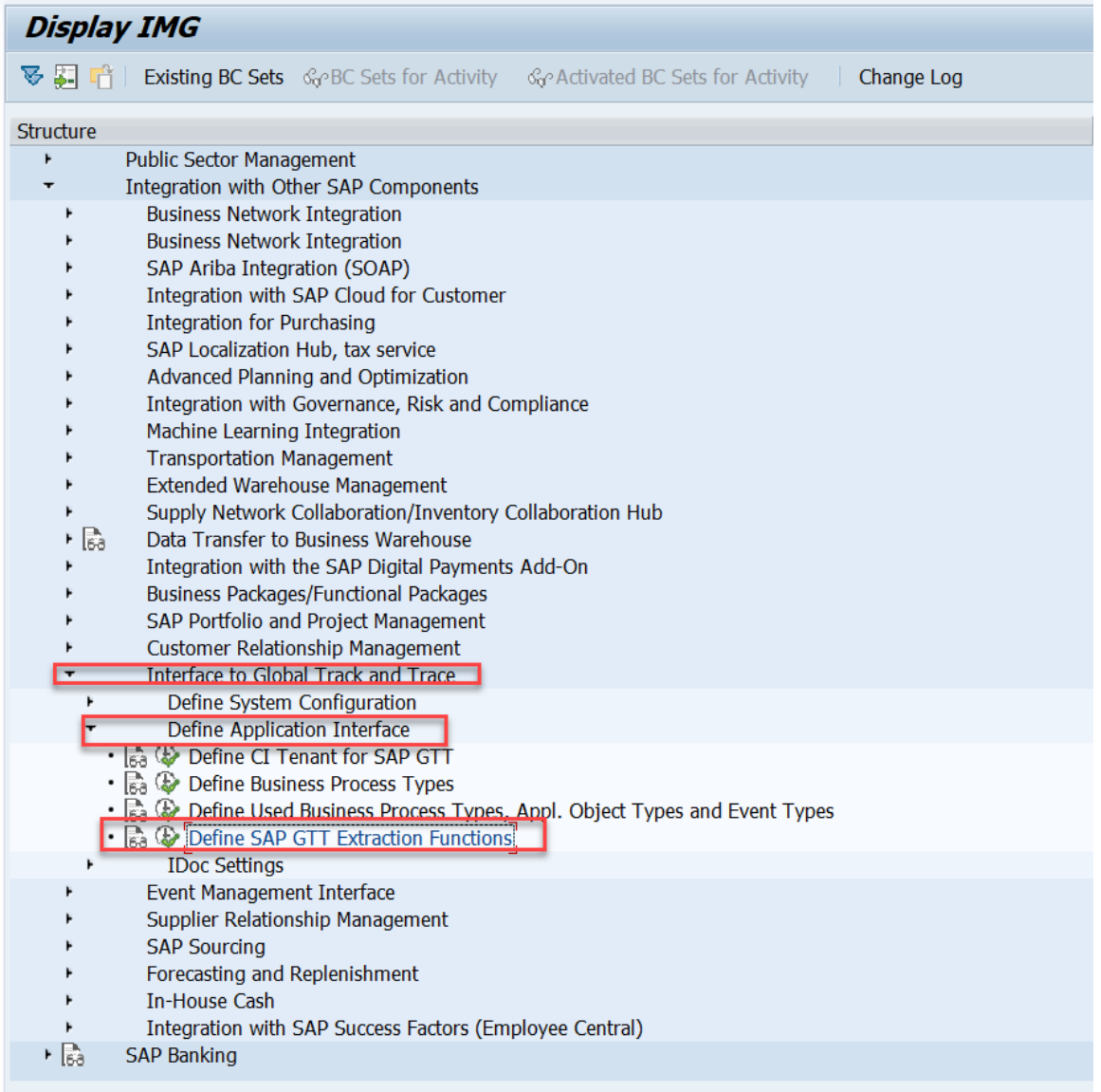




STEP 6: Define GTT Extraction Functions

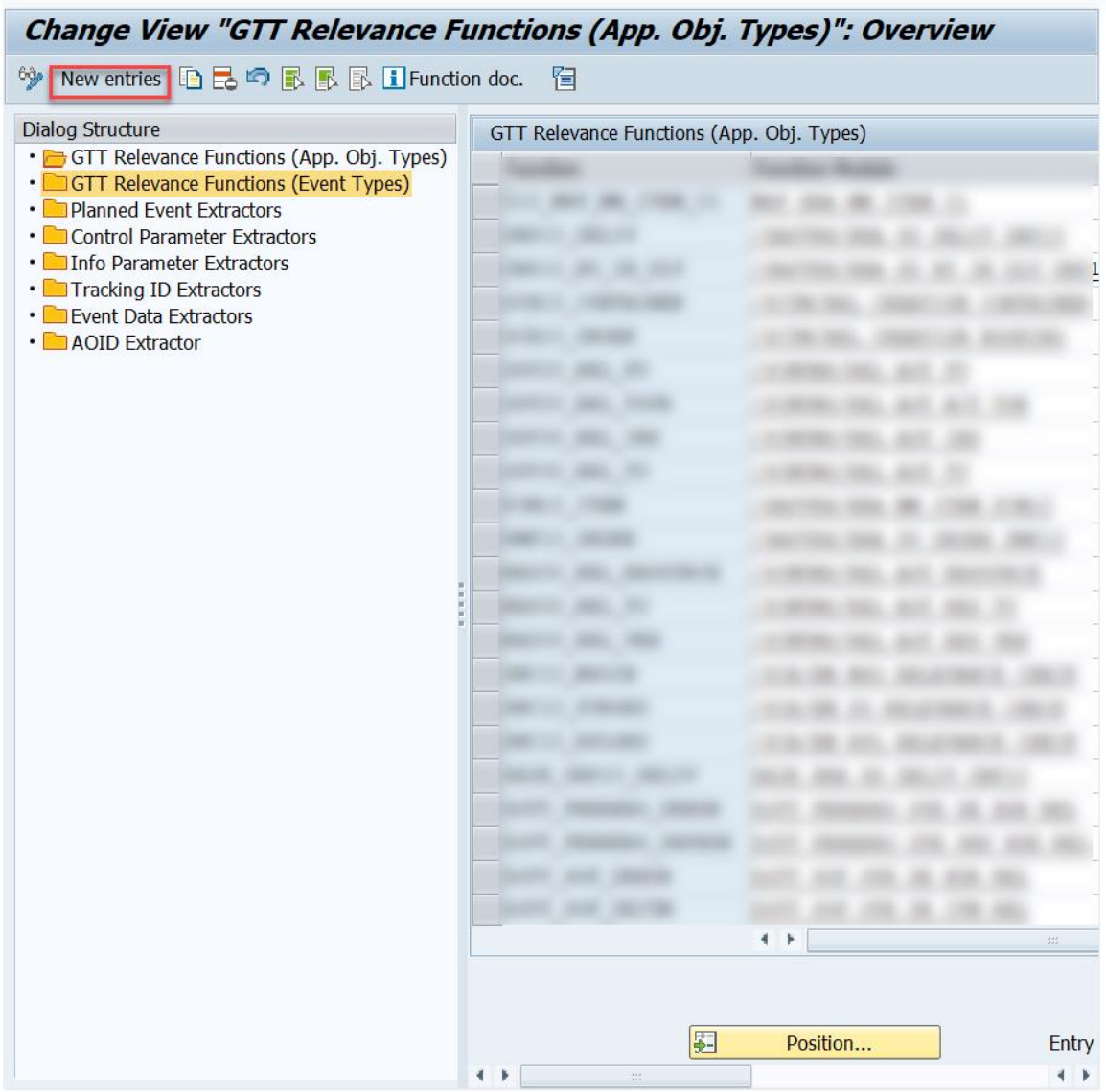
6-1: In **Display IMG** page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

6-2: Choose activity
Define SAP GTT Extraction Functions



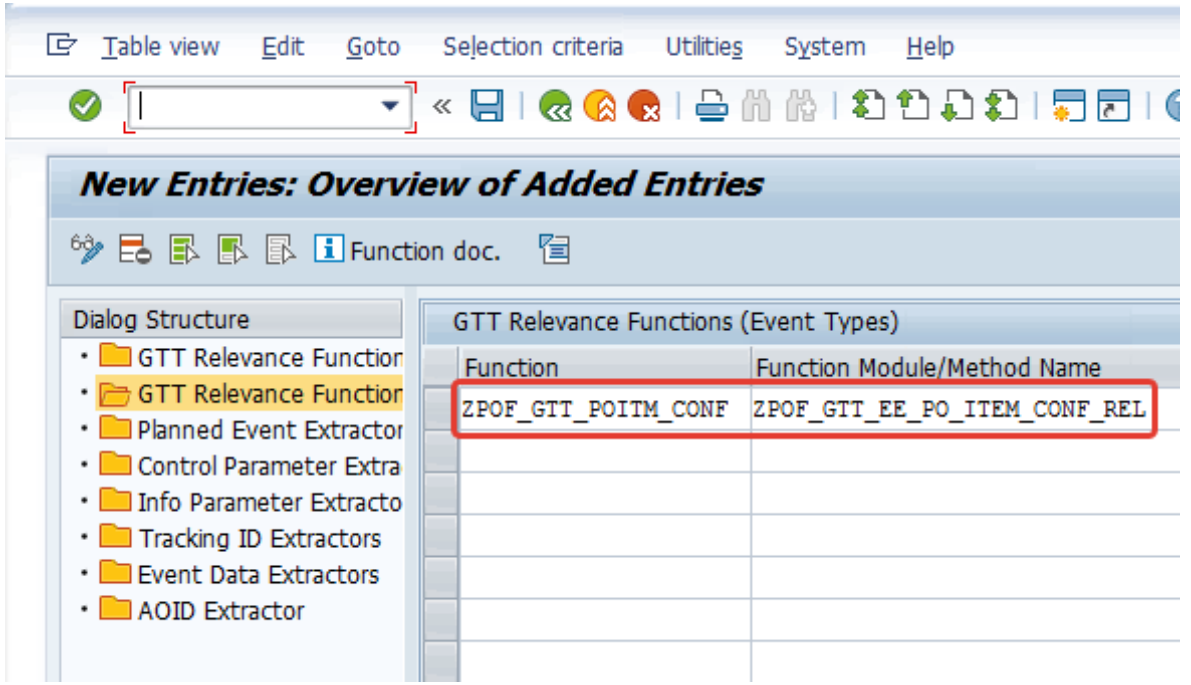
STEP 6: Define GTT Extraction Functions

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



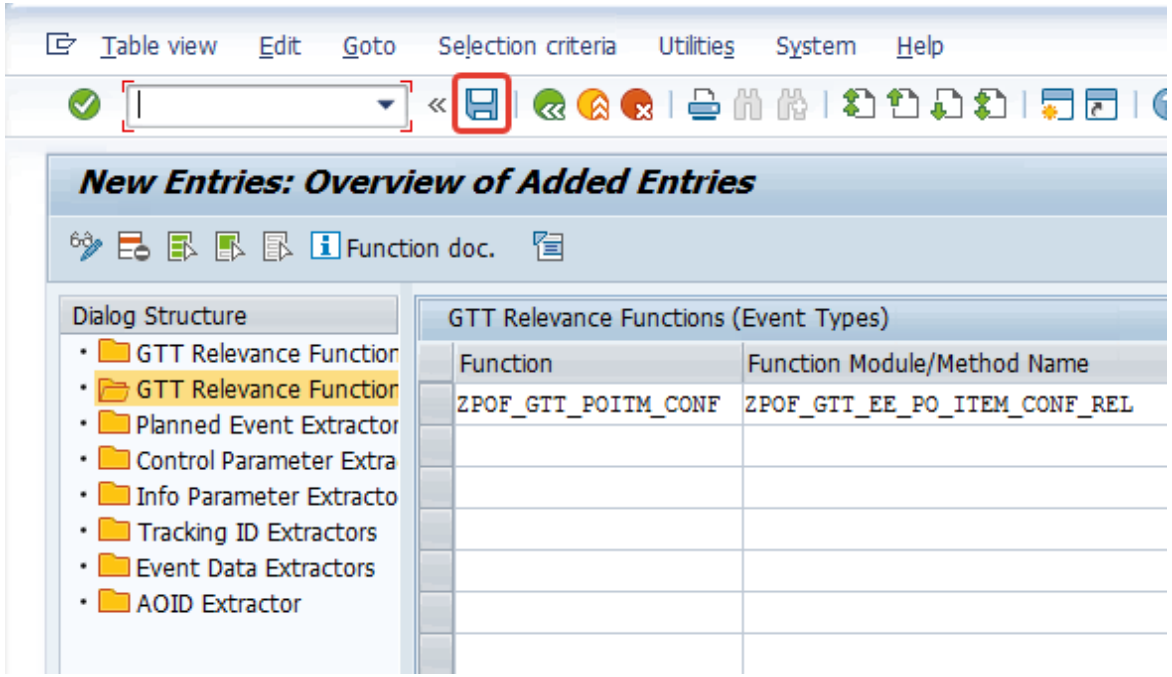
STEP 6: Define GTT Extraction Functions

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



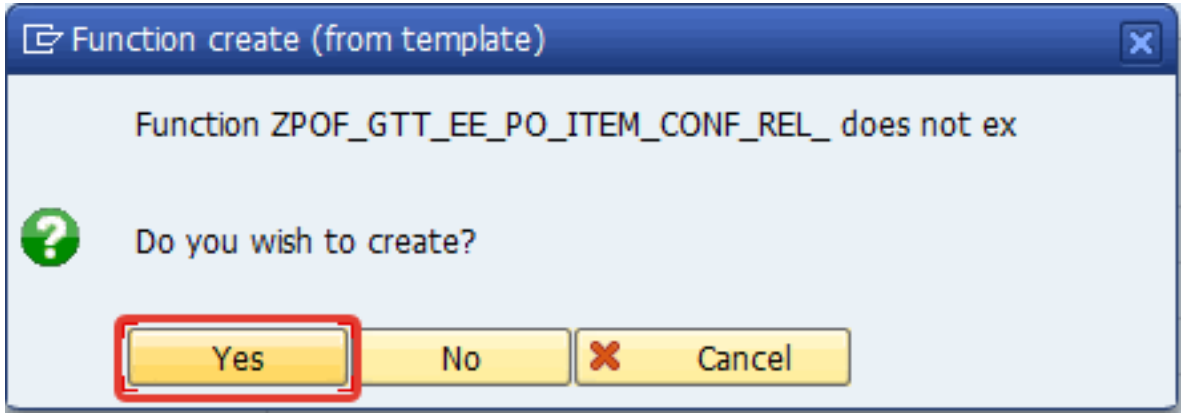
STEP 6: Define GTT Extraction Functions

6-5: Click **Save**



STEP 6: Define GTT Extraction Functions

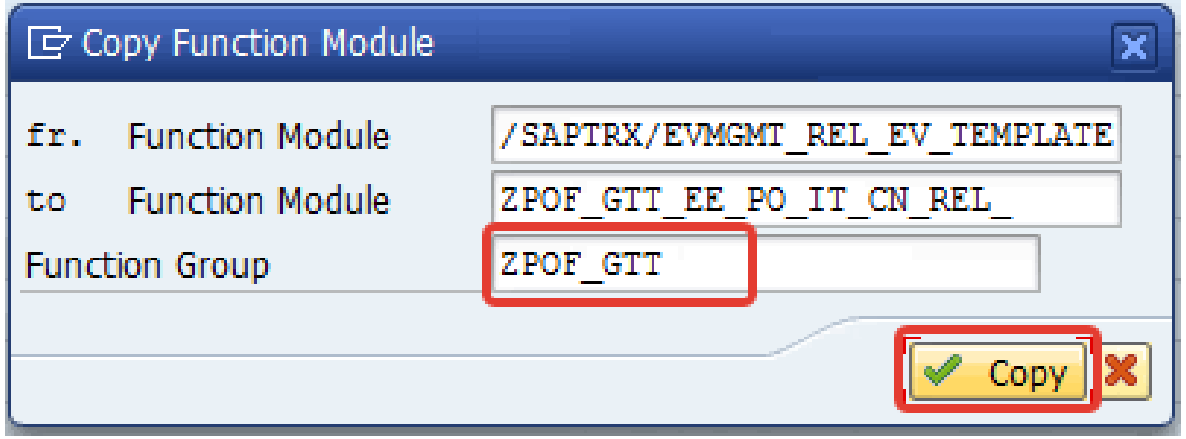
6-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



STEP 6: Define GTT Extraction Functions

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

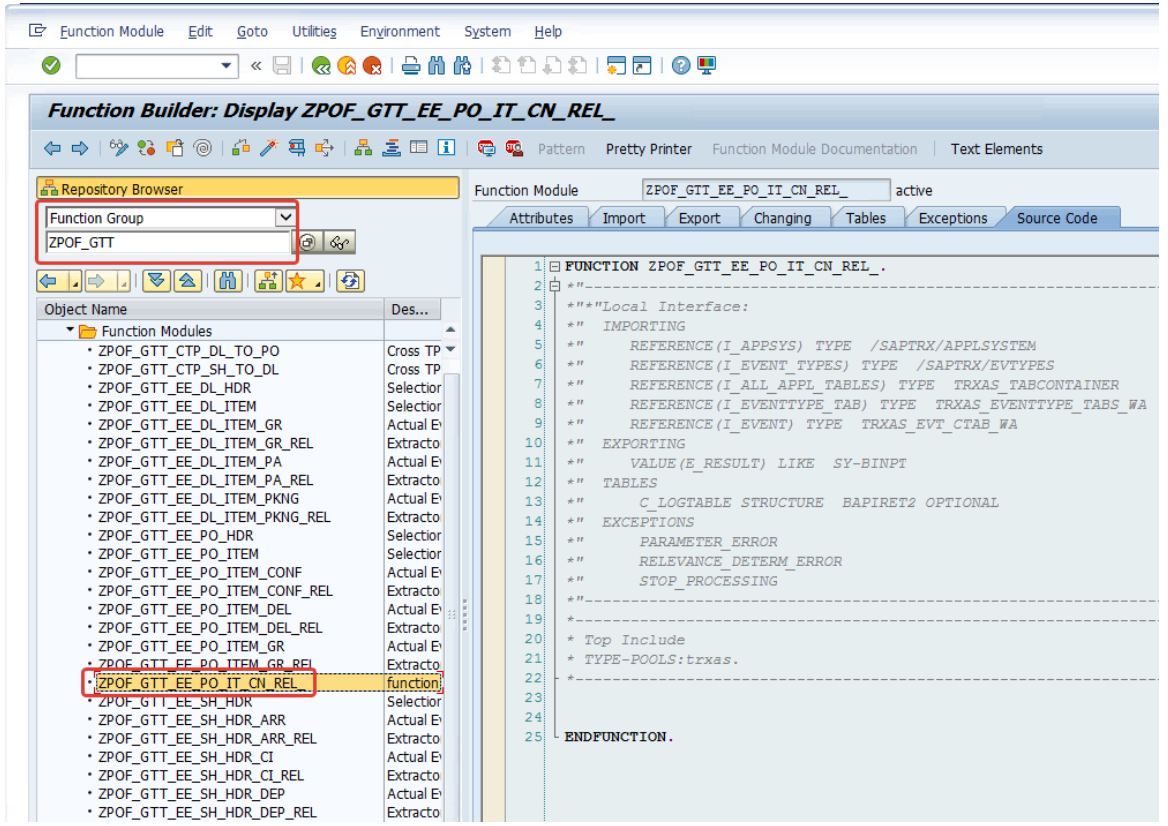
6-8: Click **Copy**



STEP 6: Define GTT Extraction Functions

6-9: Use T-Code SE80 to check the function module you just created

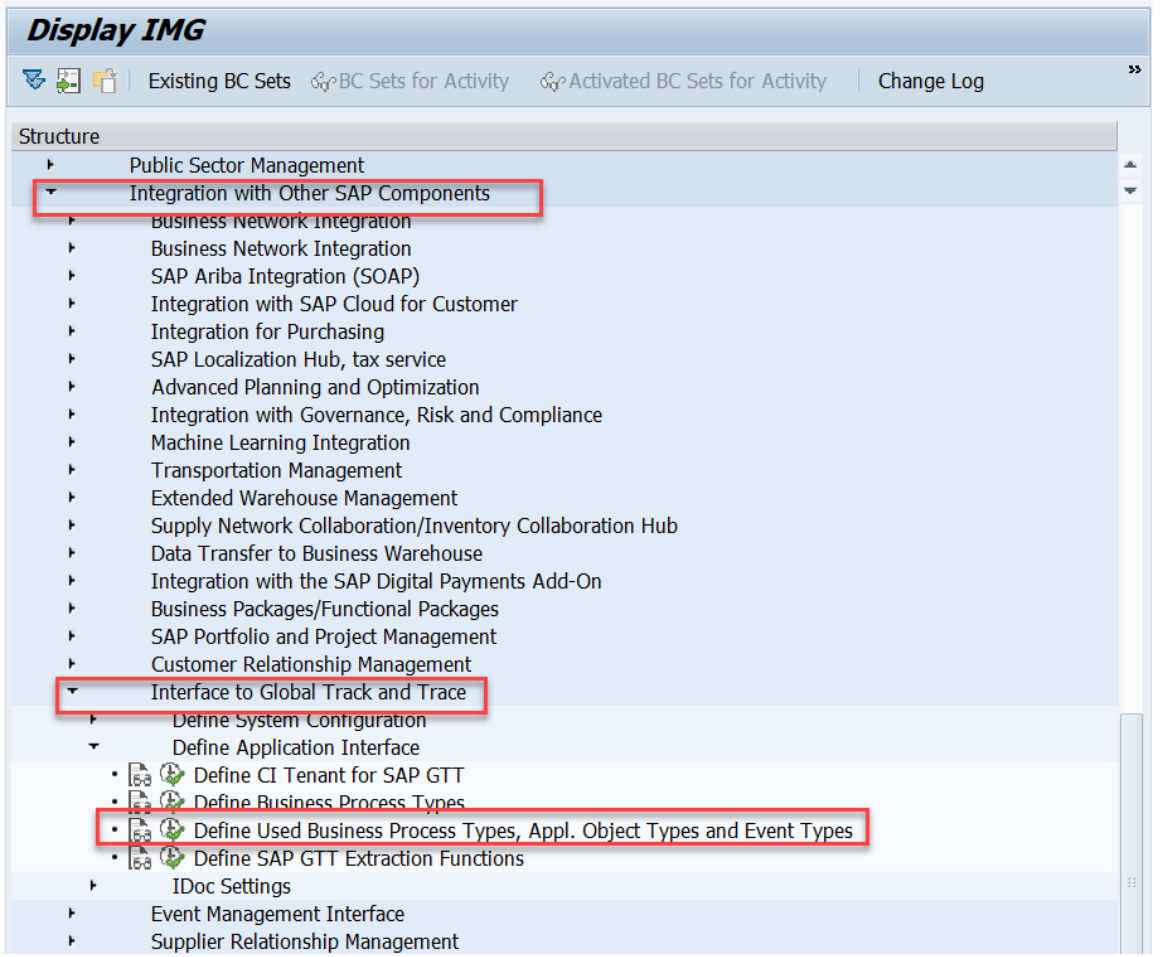
Caution: More information on how to implement extraction functions and the relevant sample code is introduced later.



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**

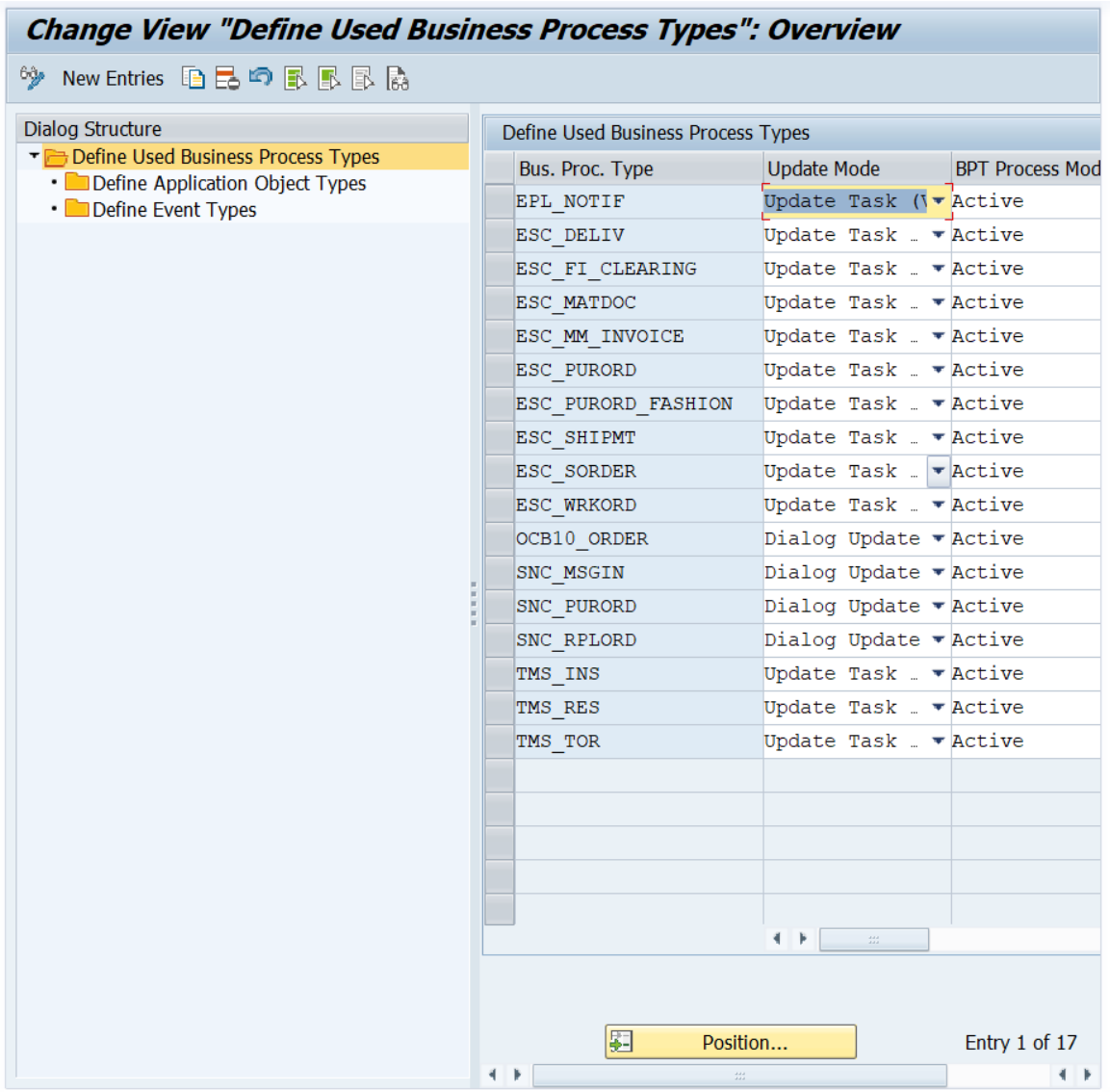


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

In the following:

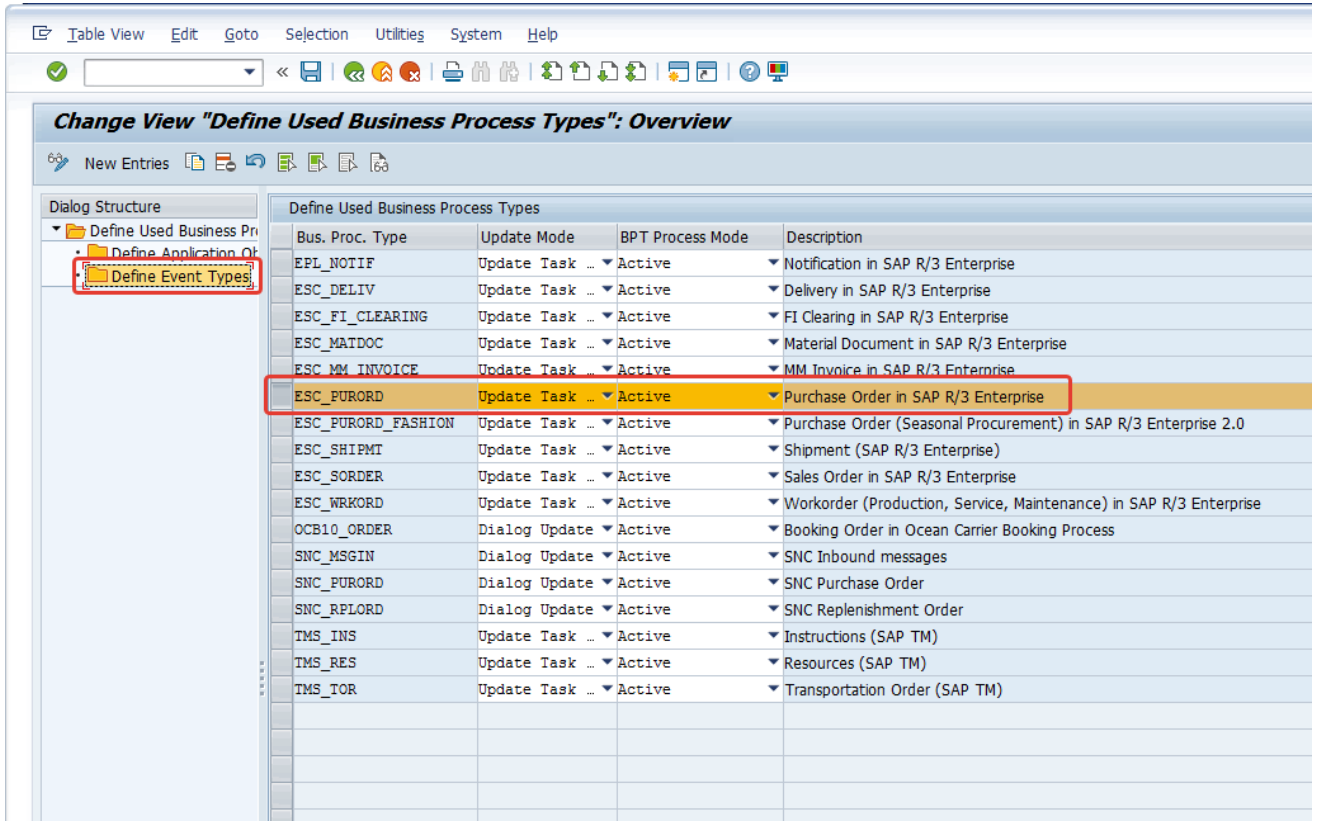
- Steps 7-3 to 7-10 demonstrate how to create an *Event Type* for a given business process type
- Steps 7-11 to 7-21 demonstrate how to create an *Application Object Type* for a given business process type



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

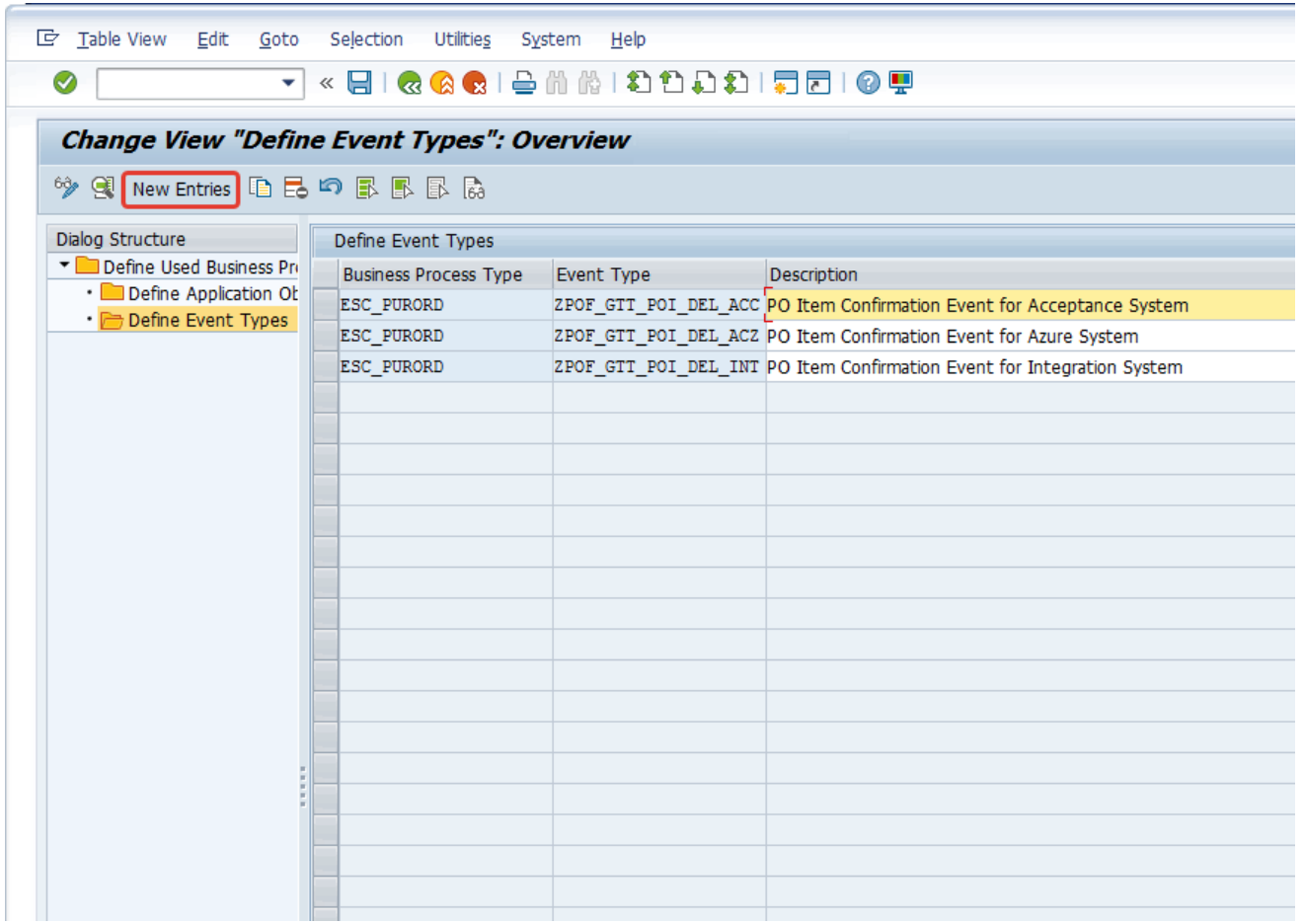
7-3: Choose the business process type from the **Define Used Business Process Types** on the right side

7-4: Double click **Define Event Types**



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

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

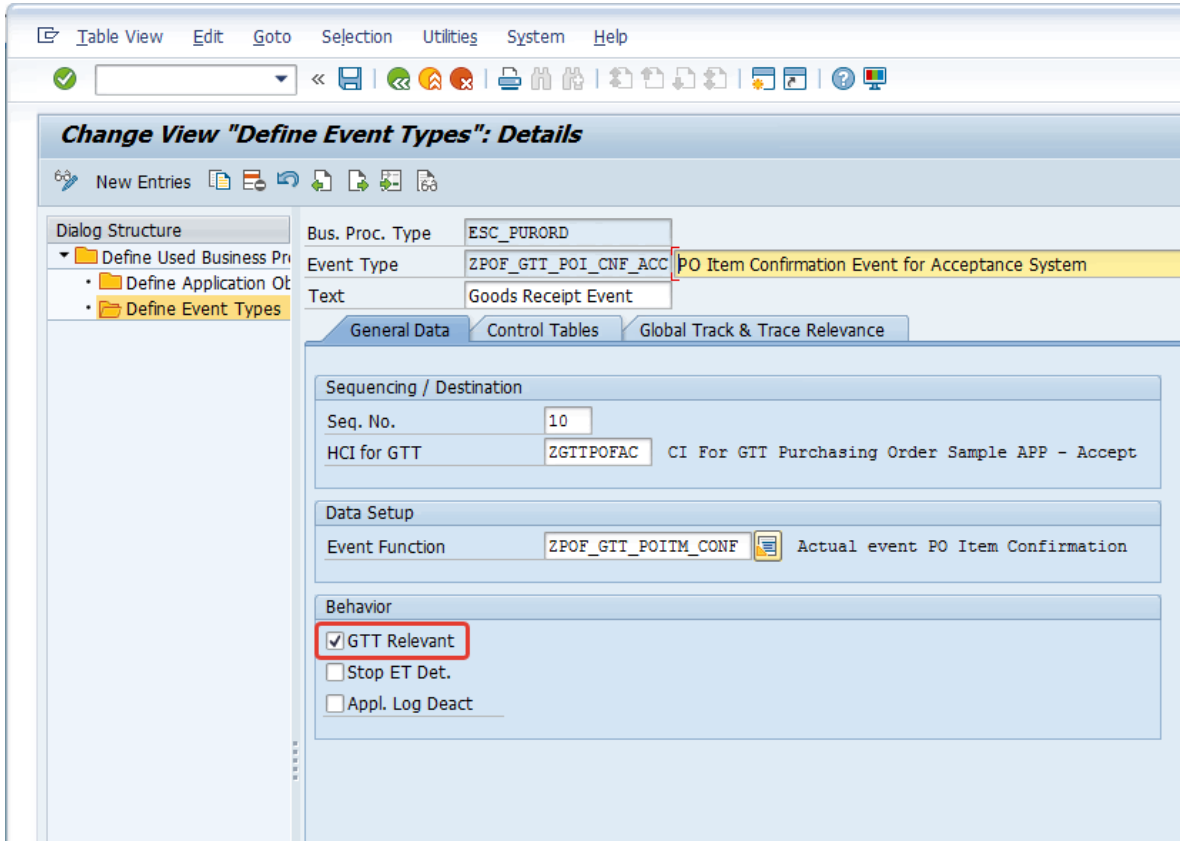


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

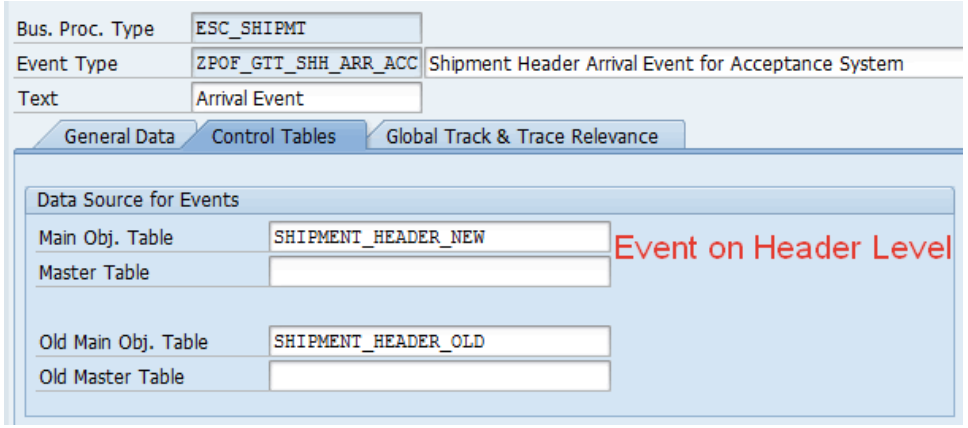
7-7: Fill in the information required in the **General Data** tab. **HCI for GTT** is the CI Tenant you created in STEP 5. **Event Function** is the extractor function you created in STEP 6.

7-8: Check **GTT Relevant**



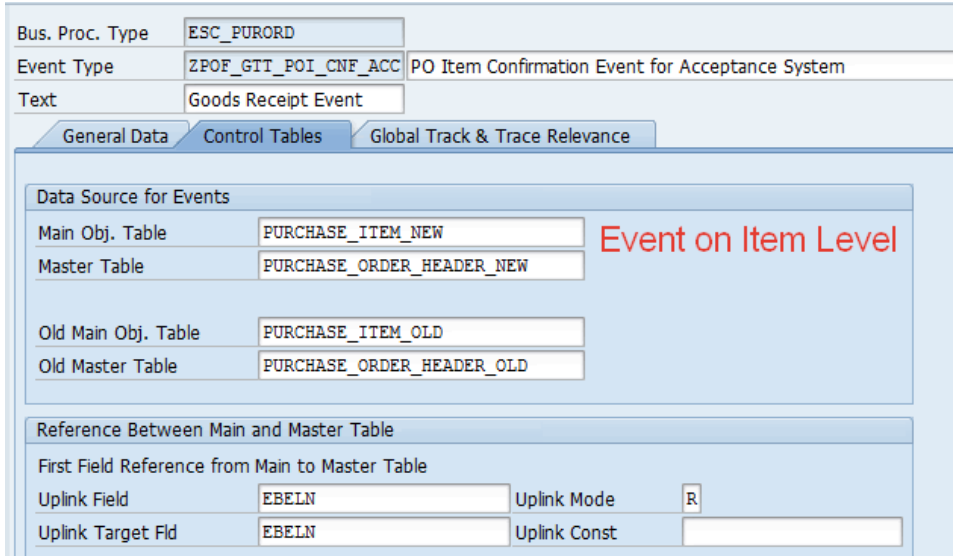
STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

7-9: Fill in the **Main Object Table** and **Master Table**.



Caution:

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

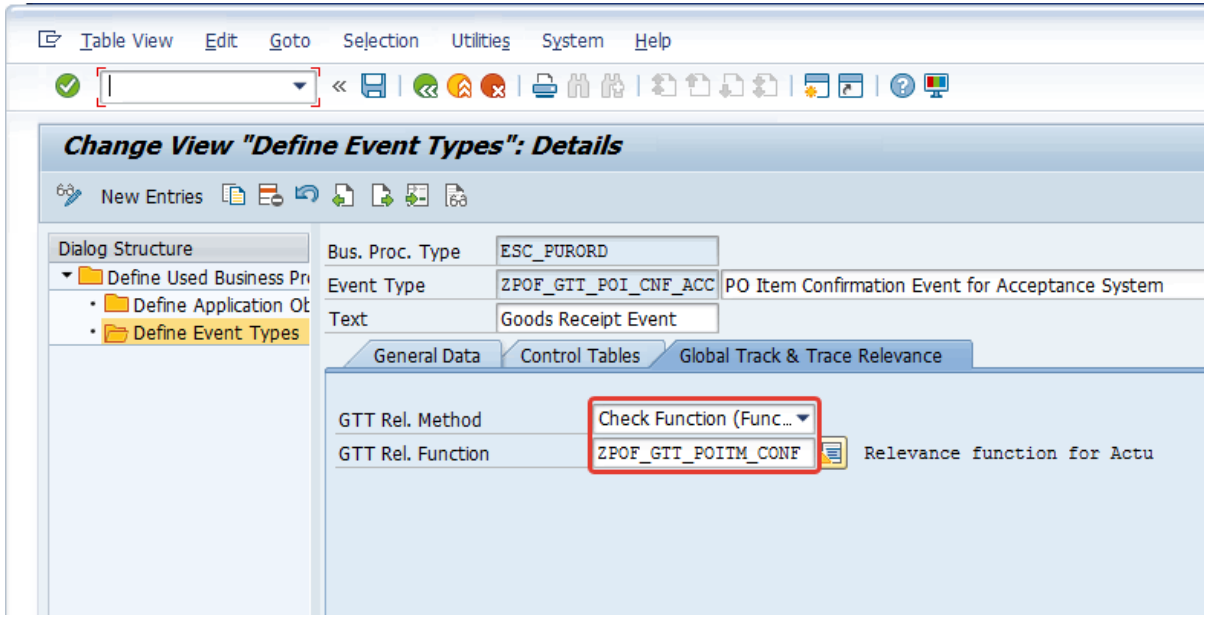


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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.

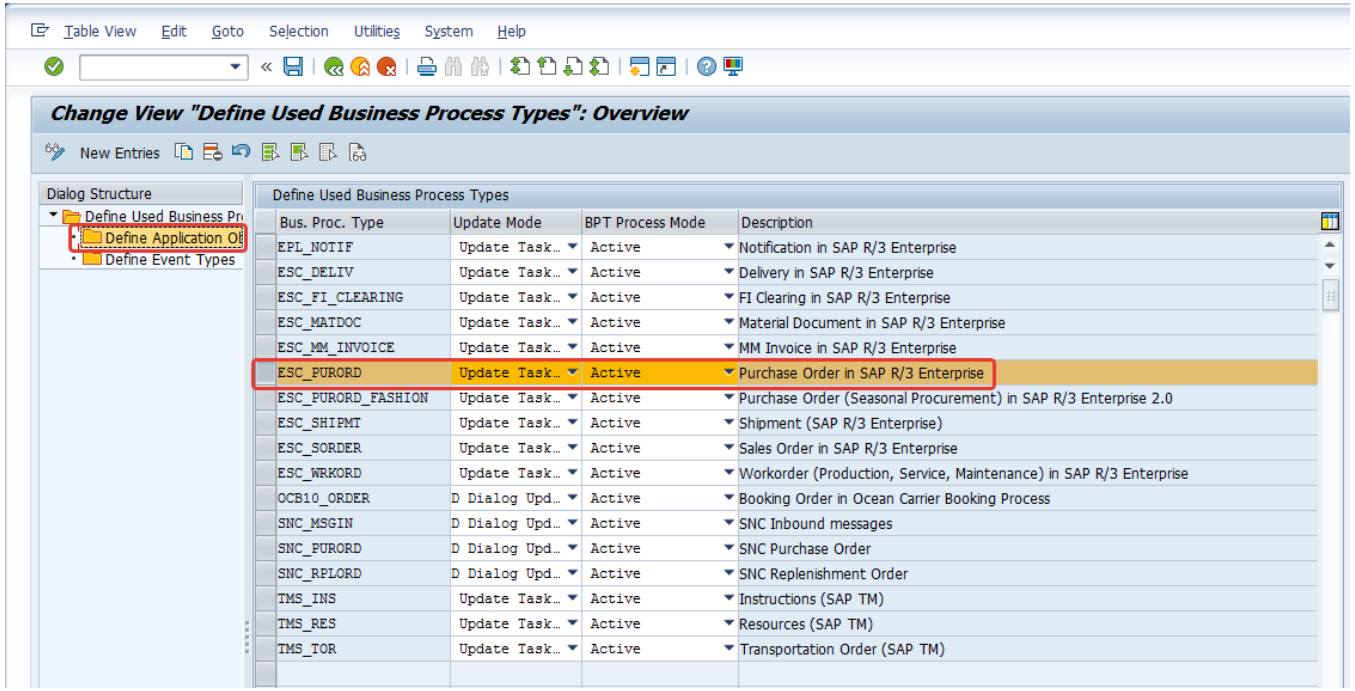
Click **Save**.



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

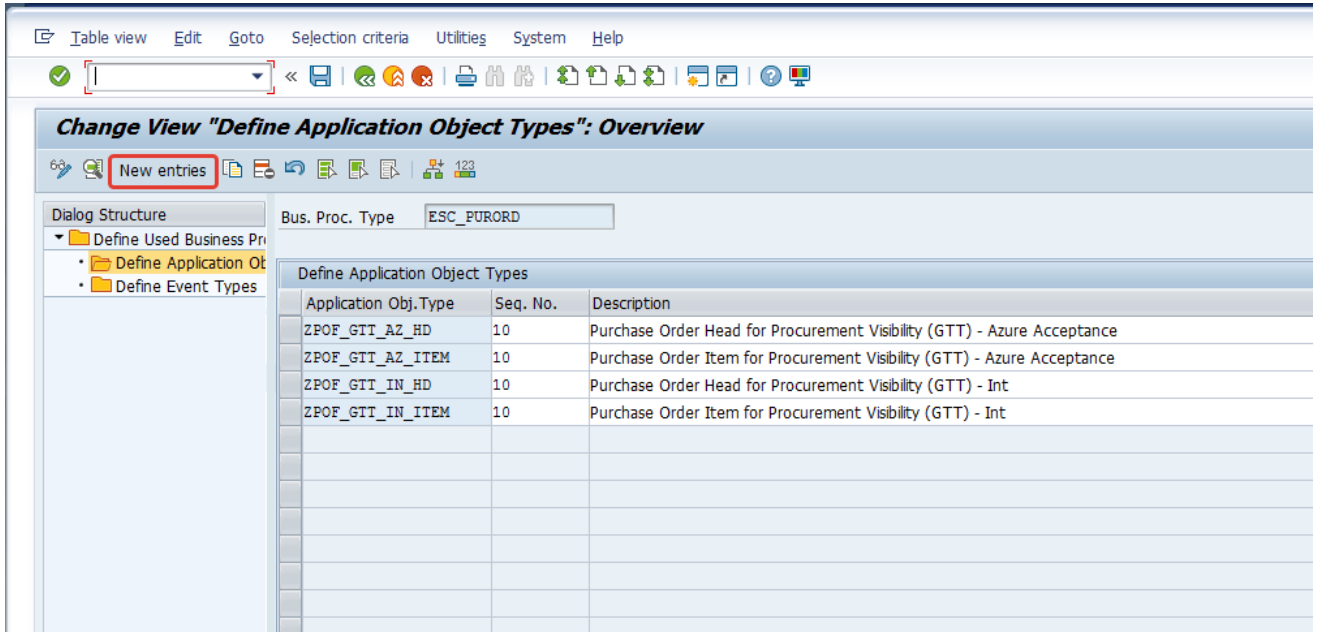
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

7-12: Double click **Define Application Object Types**



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

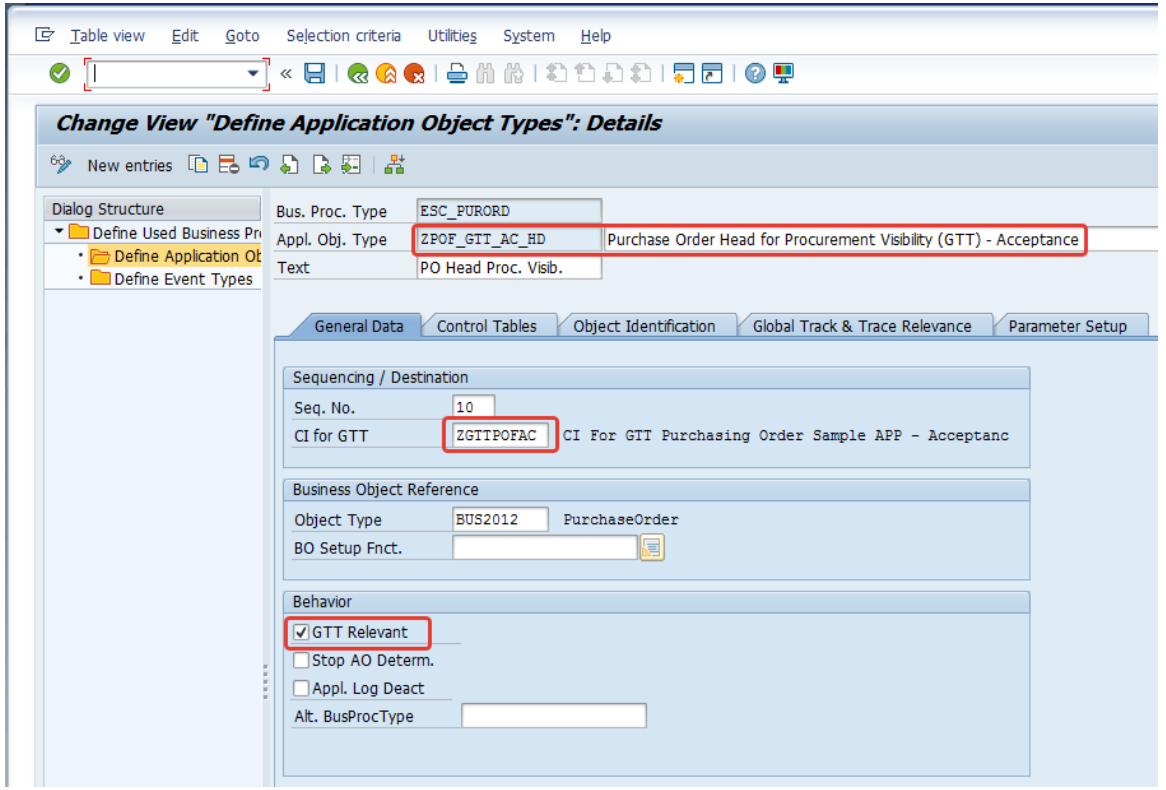


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

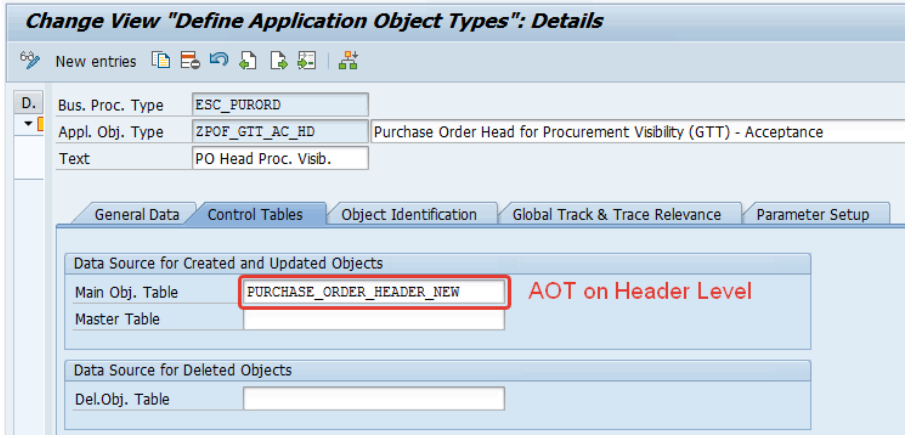
7-15: Fill in the information required in the **General Data** tab. **CI for GTT** is the CI Tenant you created in STEP 5.

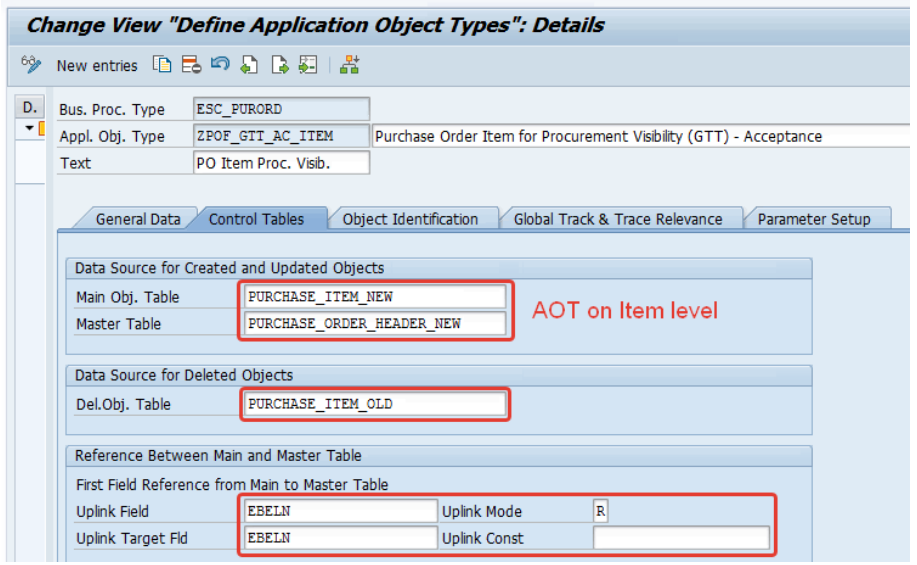
7-16: Check **GTT Relevant**



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

7-17: Fill in the **Main Object table** and **Master Table**





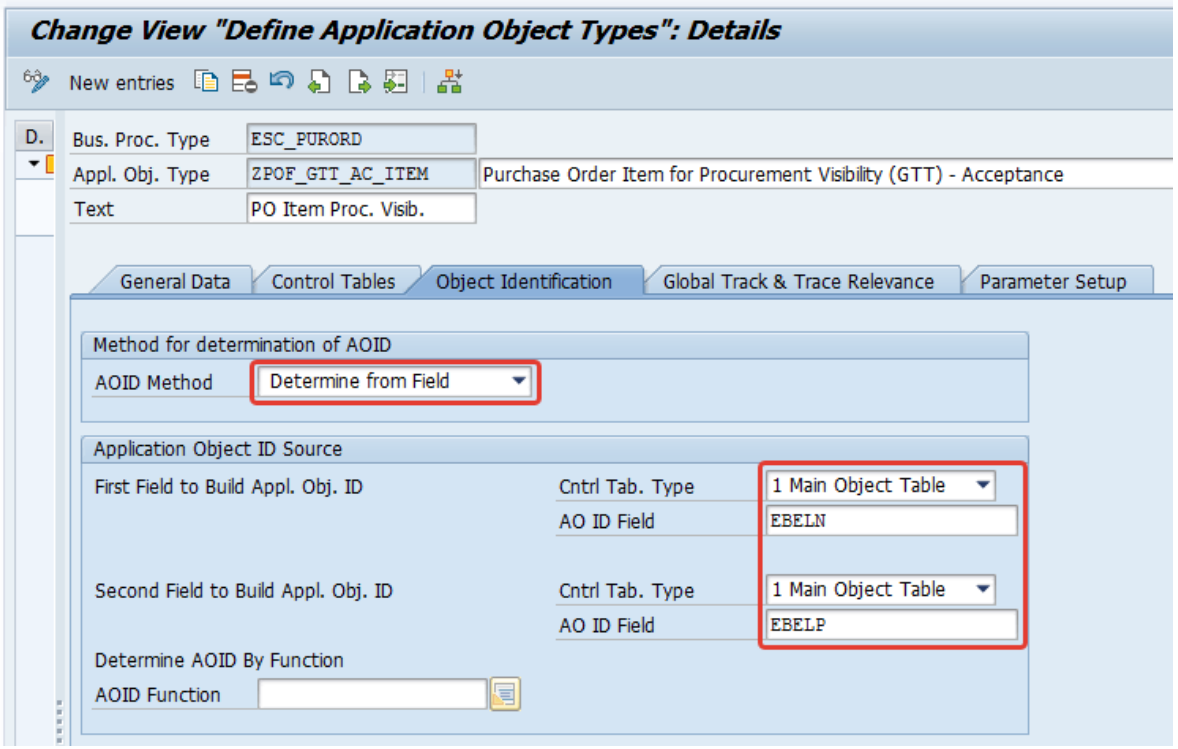
Caution:

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

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**, use the key field to fill the AO ID fields

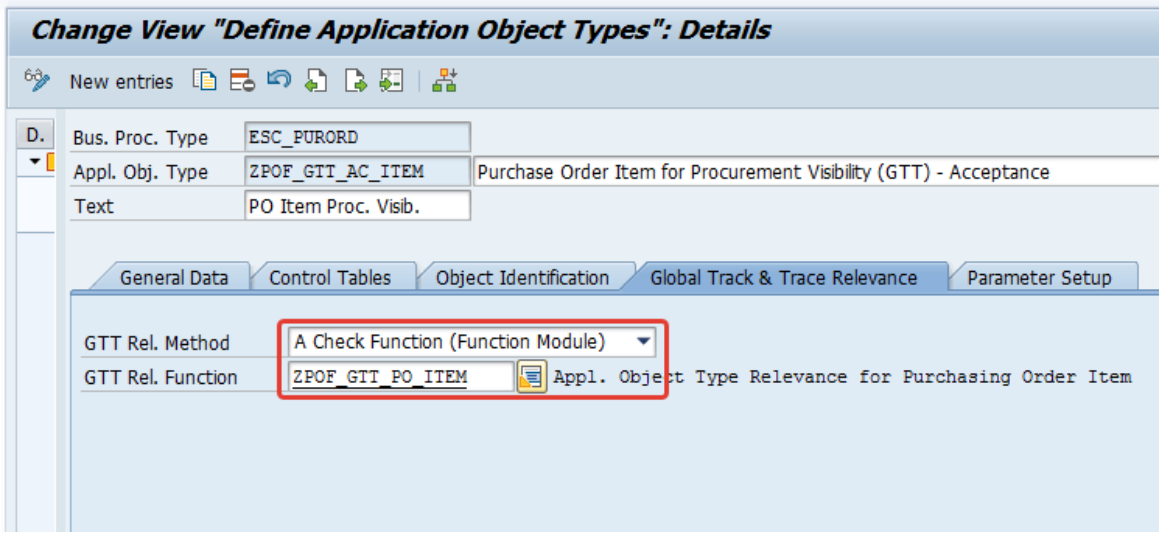
7-19: When choosing **Determine by Function**, you must enter the customized information in the AOID function field.



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*, then 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

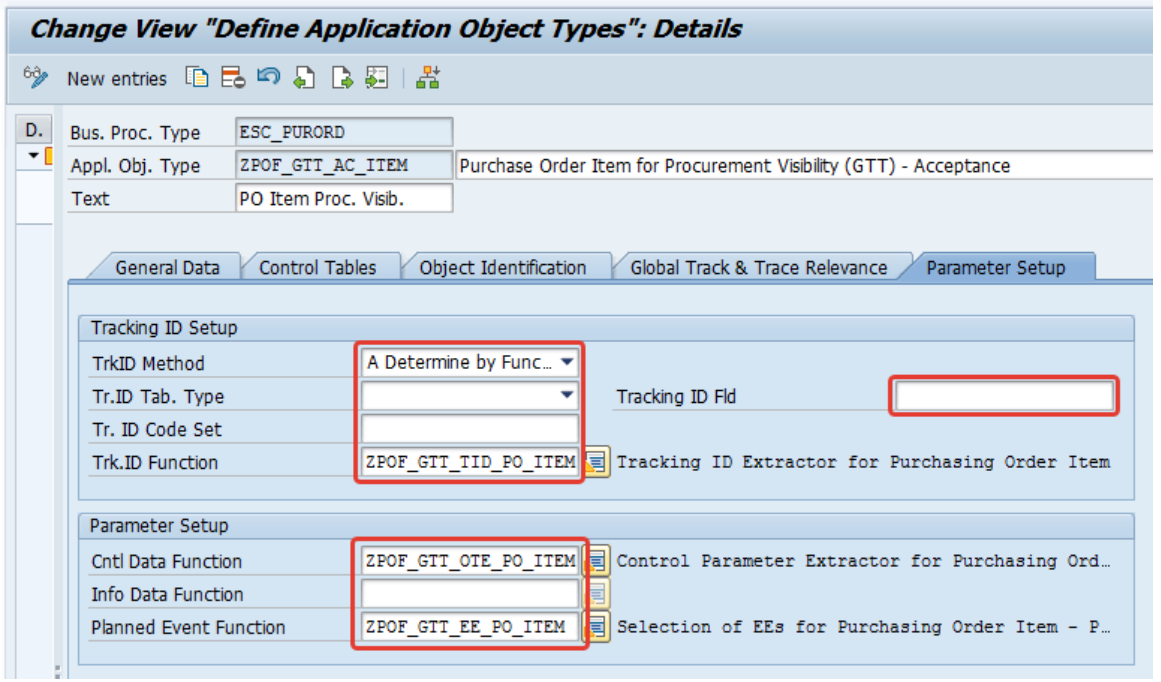
7-21: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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 you need to fill the key field and name the Code Set for the AOT.

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

Click **Save**.



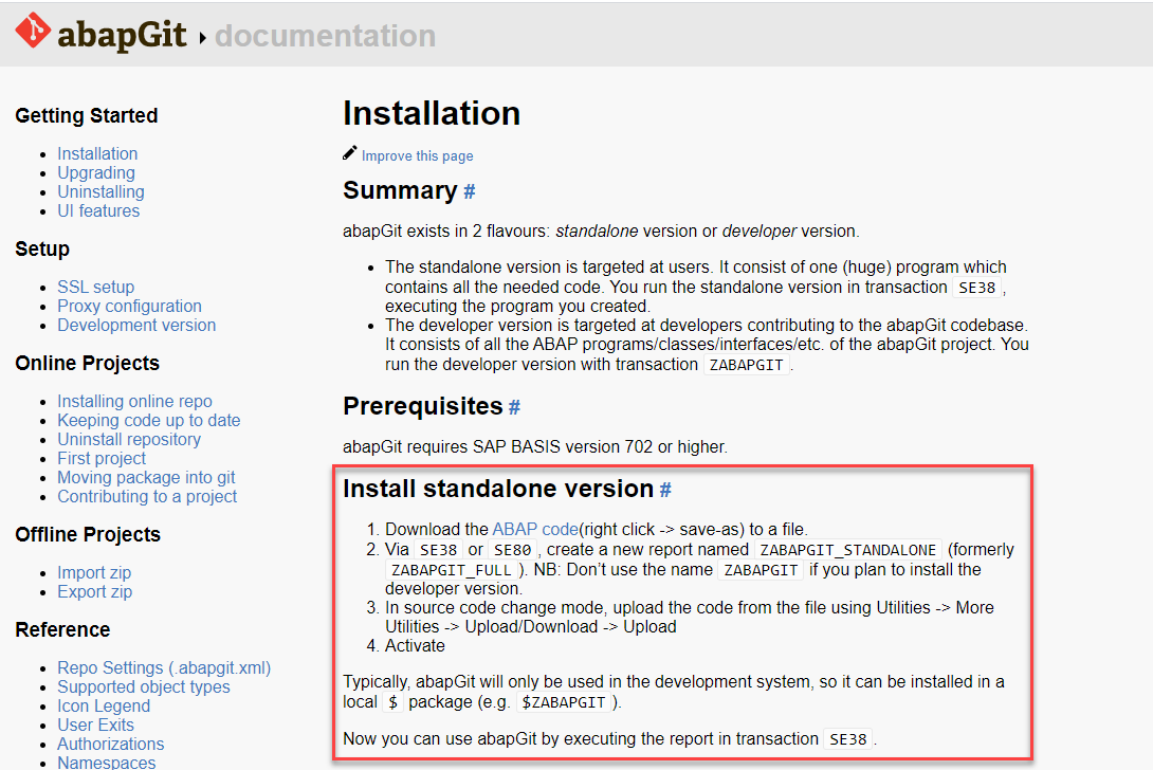
C) Download ABAP Code from GitHub

STEP 1: Install abapGit

You need to install abapGit before downloading codes from GitHub.

To install abapGit, follow the instructions on https://docs.abapgit.org/guide-install.html.
Make sure you **Install the standalone version** in your dev system.

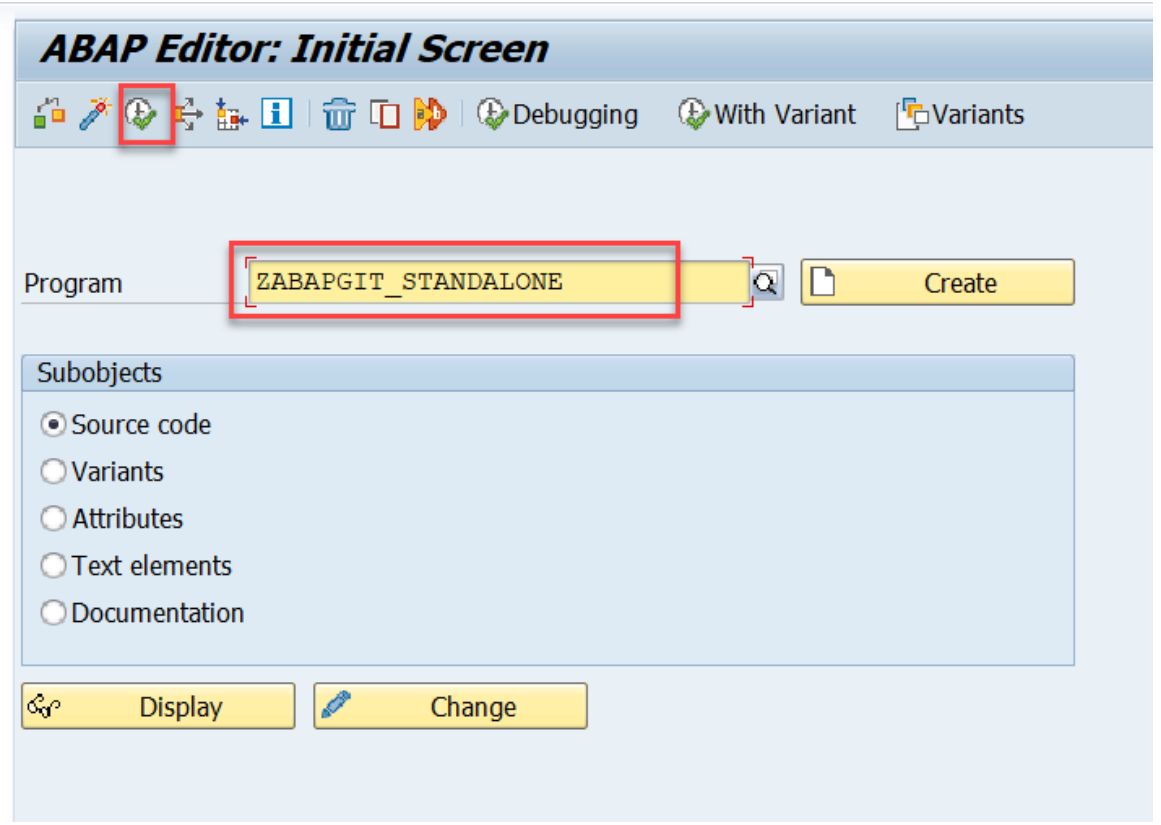
When installation is complete, a new report is created, *ZABAPGIT_STANDALONE*.



STEP 2: Download ABAP Code

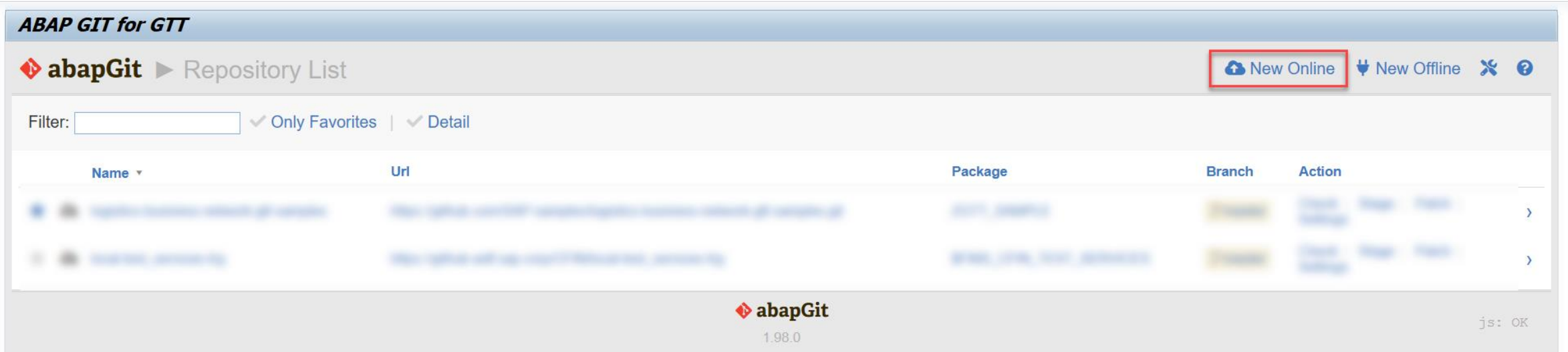
2-1: Enter T-code *SE38* and fill in the report name from STEP 1, *ZABAPGIT_STANDALONE*

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



STEP 2: Download ABAP Code

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



STEP 2: Download ABAP Code

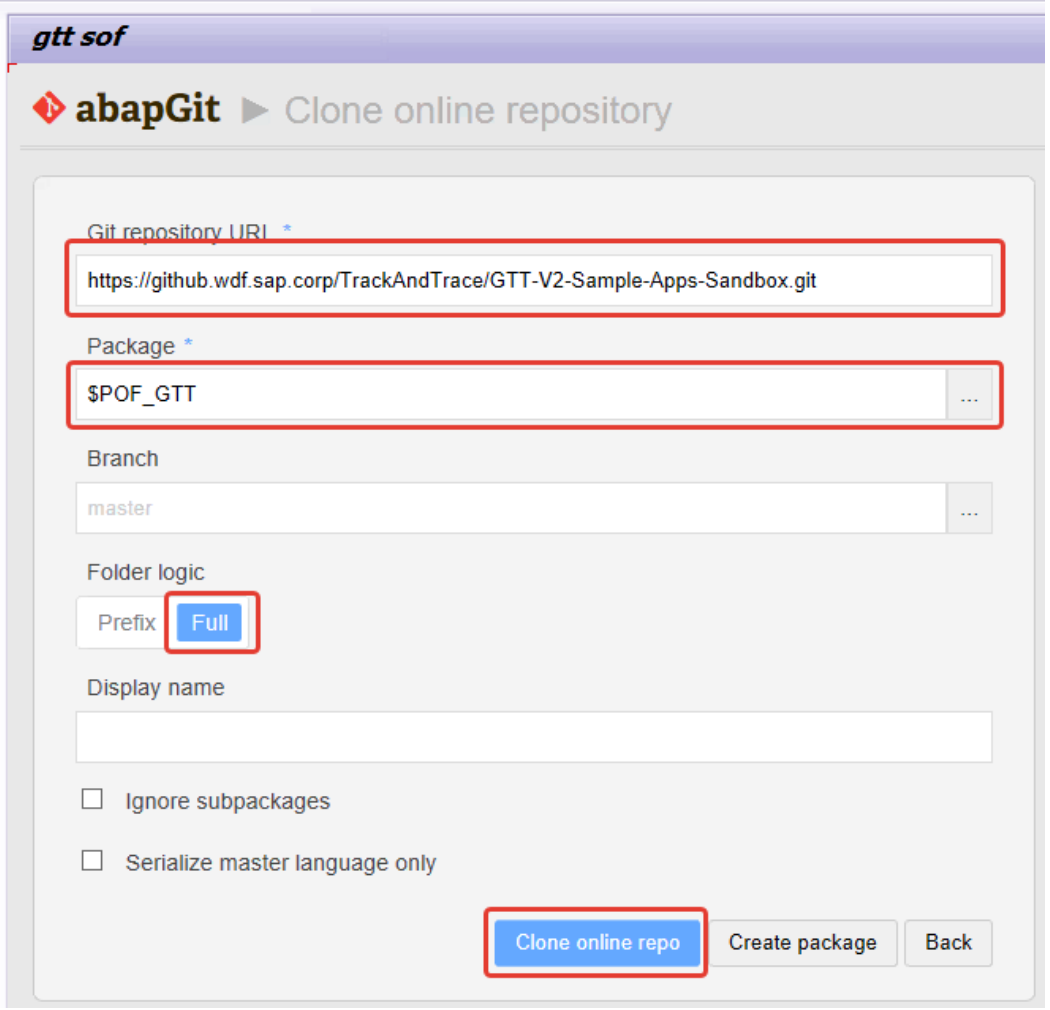
2-4: Fill in the **Git repository URL**:

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**

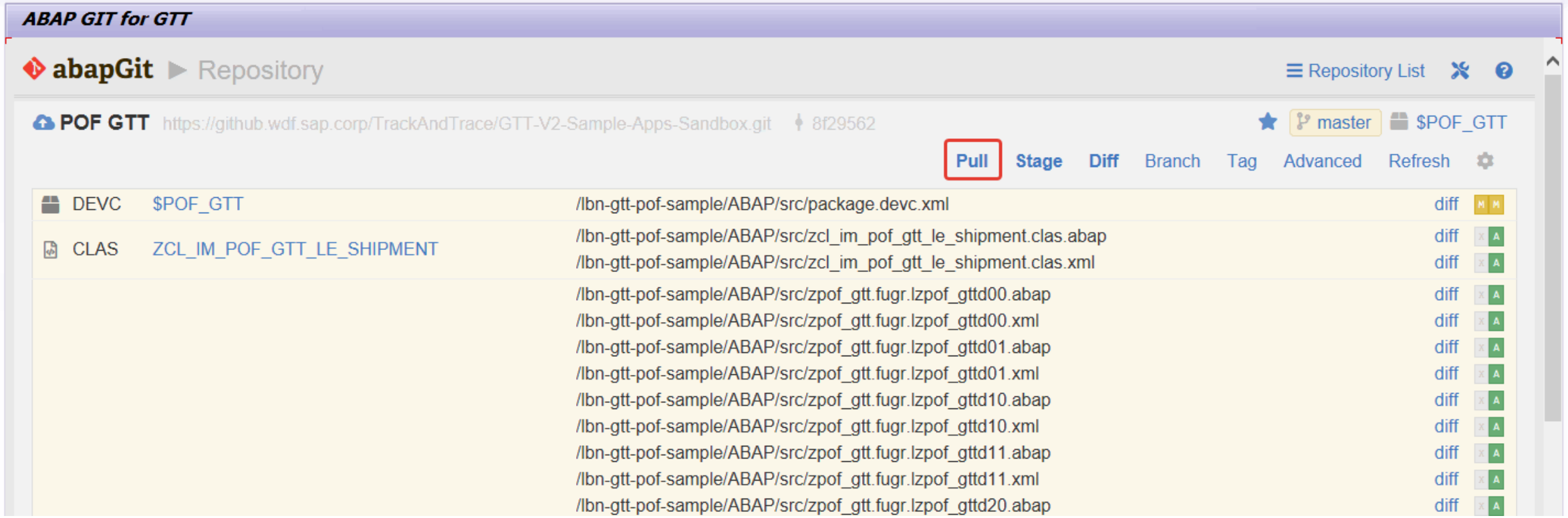
2-7: Click **Clone online repo** to download the code



STEP 2: Download ABAP Code

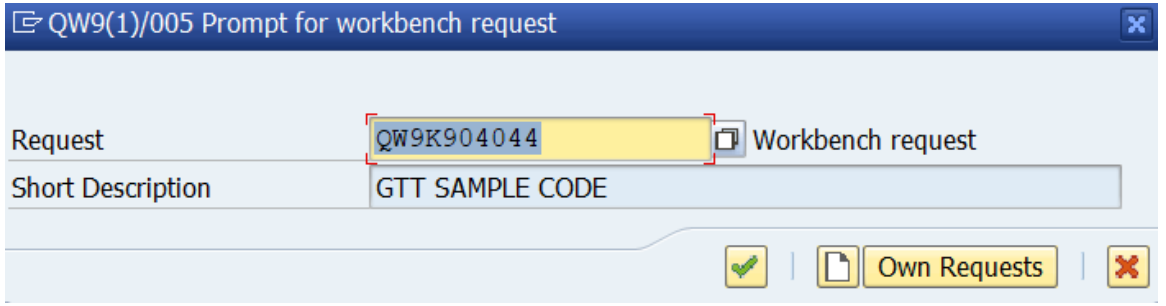
2-8: Click **Pull** to pull down the latest version code

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



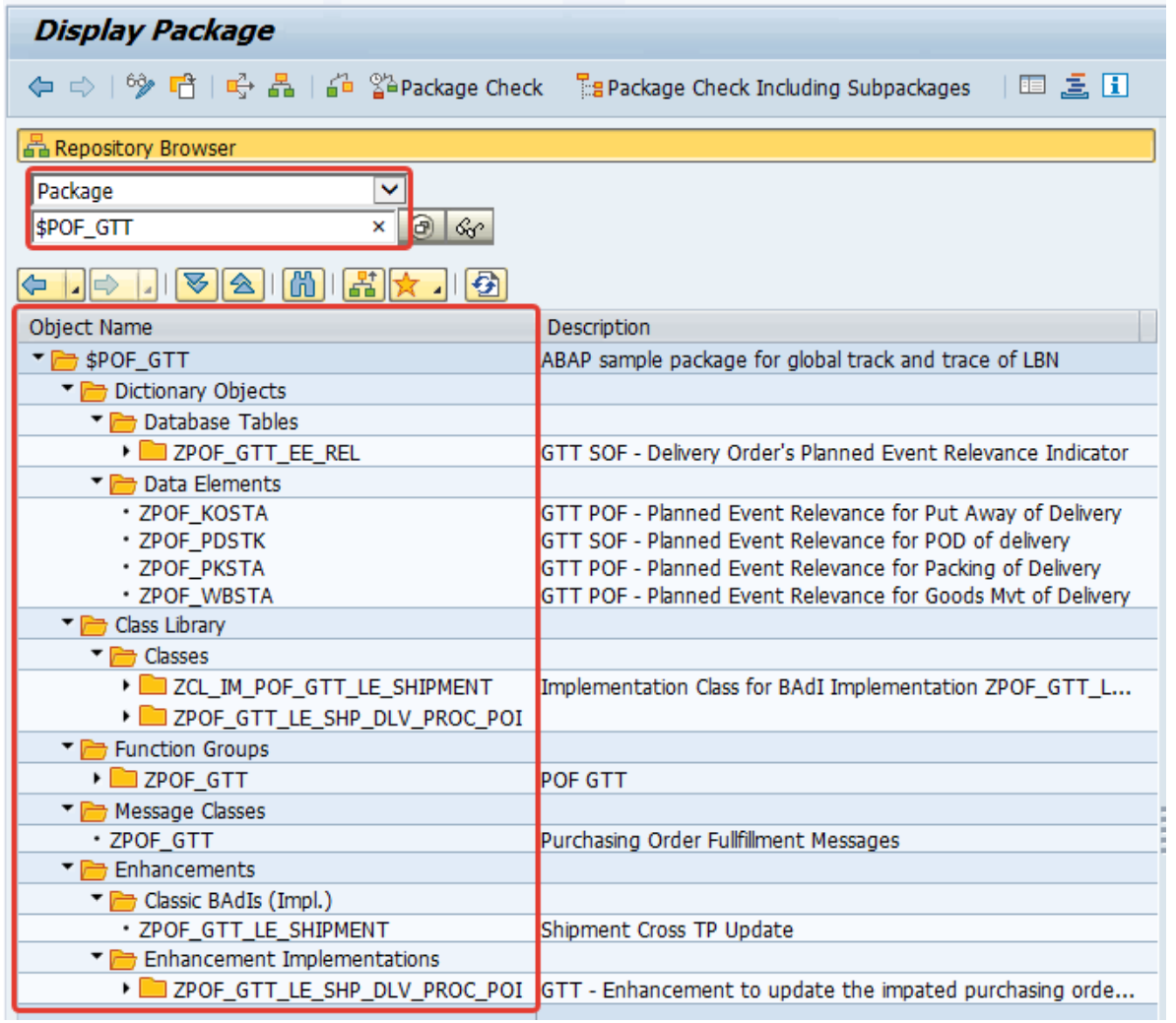
STEP 2: Download ABAP Code

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



STEP 2: Download ABAP Code

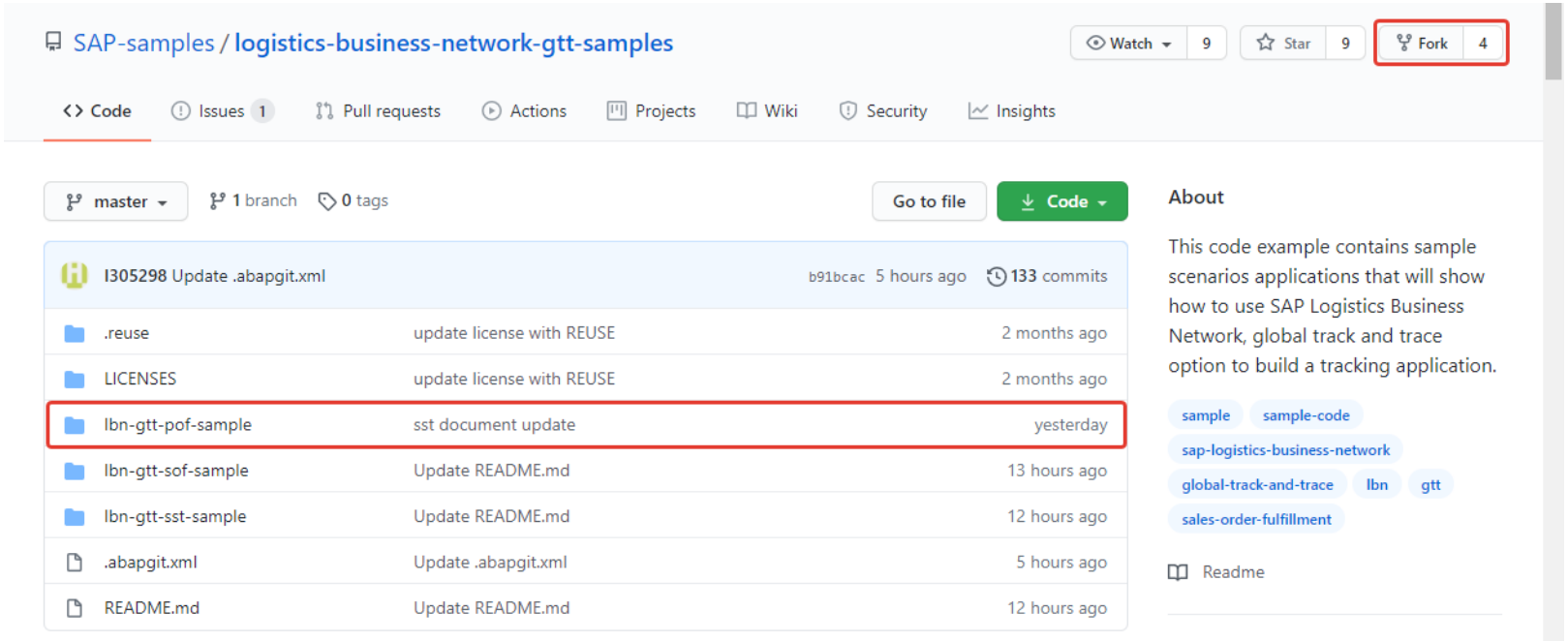
2-10: After you download the code, you can check them with T-code *SE80*.

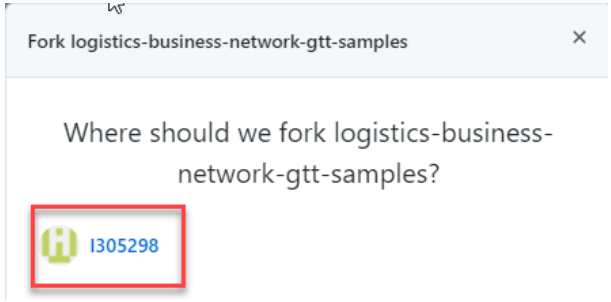


Step 3: Download ABAP Code Within the Specified Folder

3-1: If the user wants to download only the sample codes in the folder of ‘lbn-gtt-pof-sample’ from Github instead of all downloaded, 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.

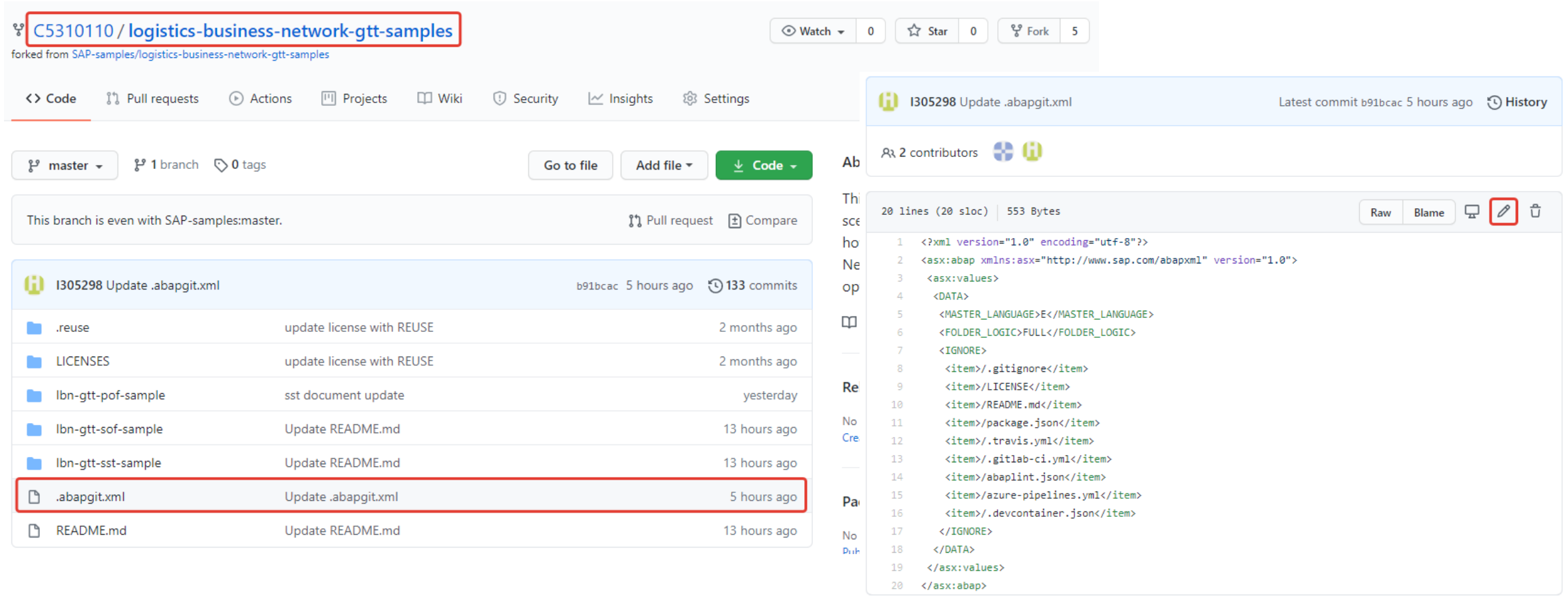




Step 3: Download ABAP Code Within the Specified Folder

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

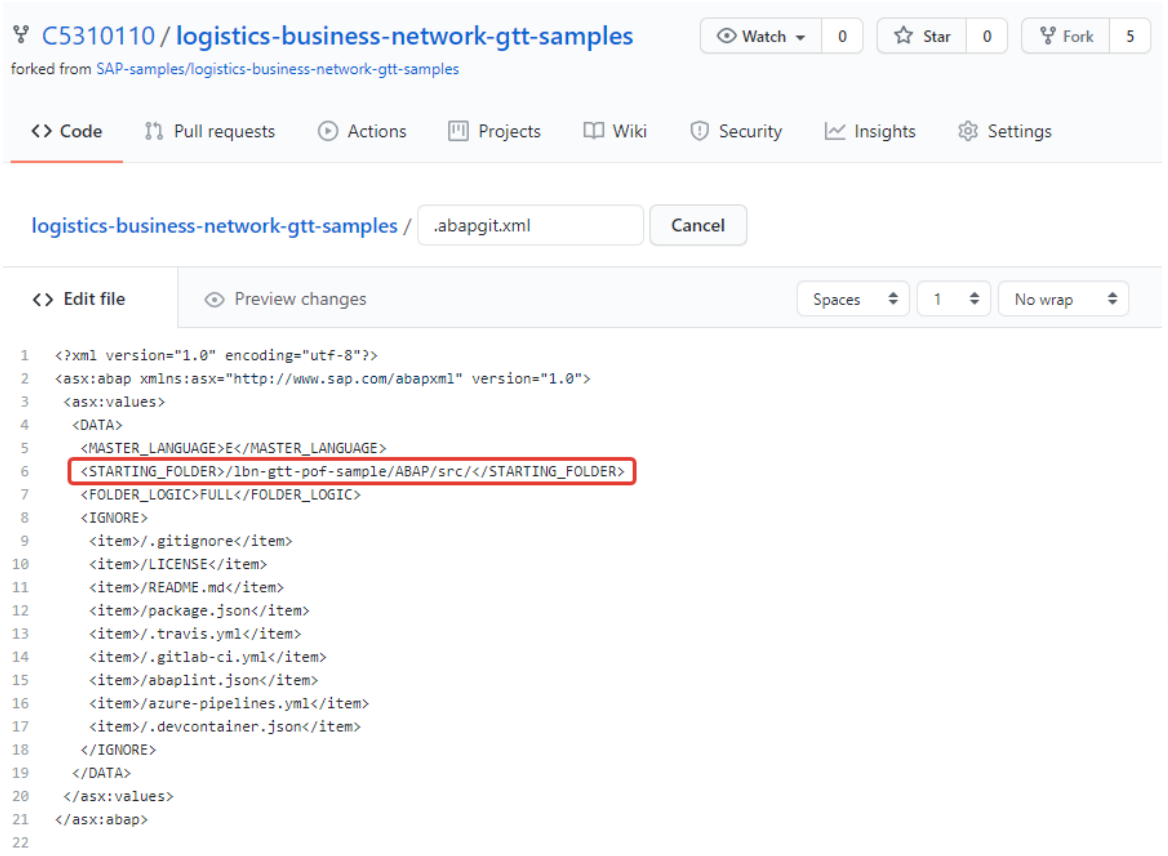
3-4: Click ✎ button to edit the file

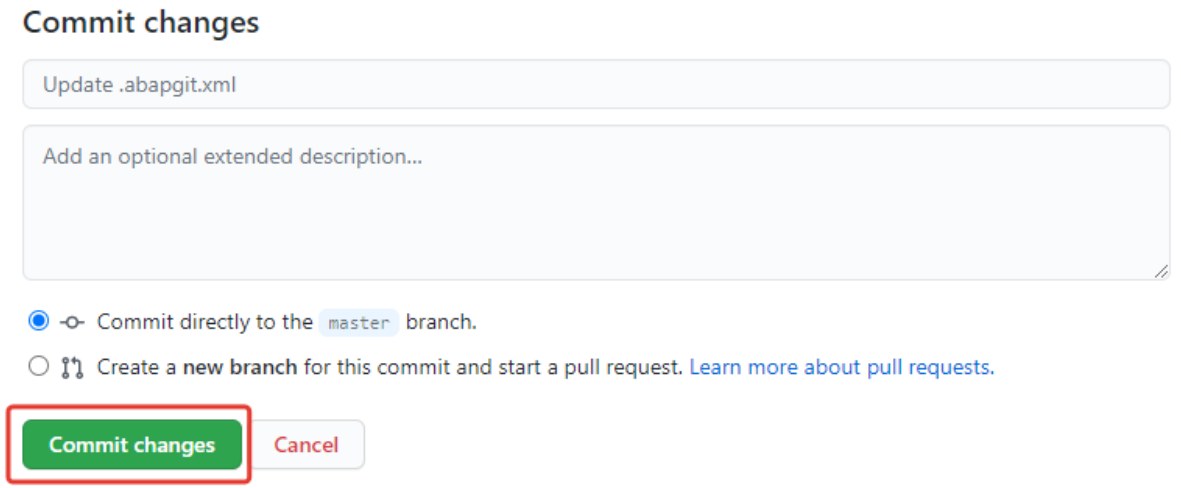


Step 3: Download ABAP Code Within the Specified Folder

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

3-6: Commit change

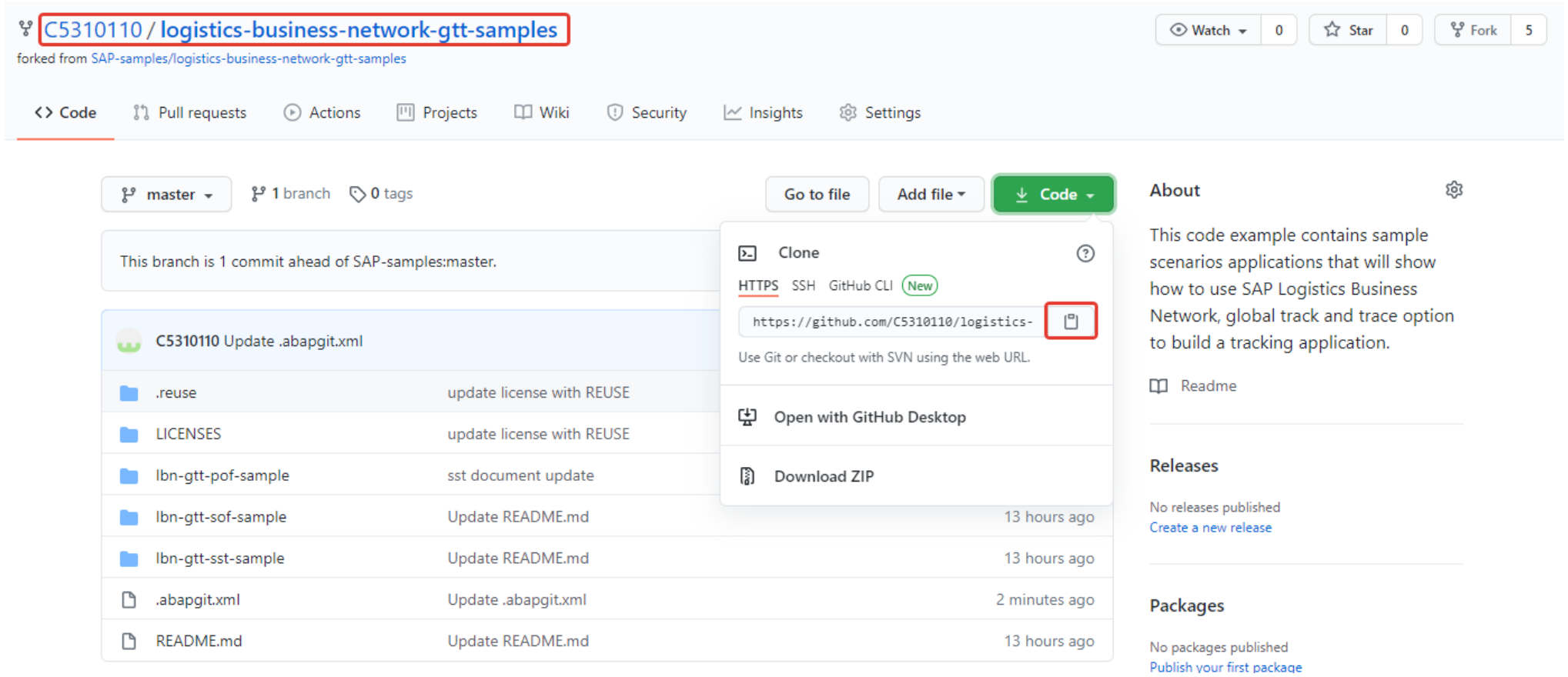




Step 3: Download ABAP Code Within the Specified Folder

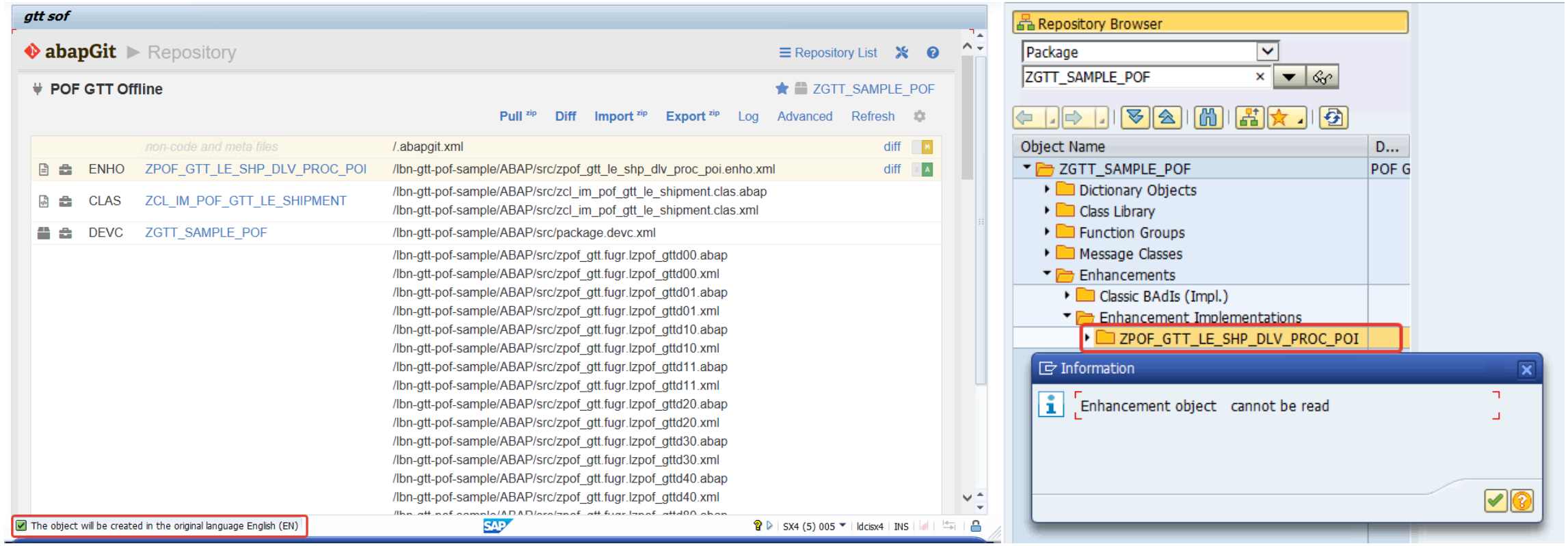
3-7: Go to the root and copy the repository URL by clicking button

3-8: Repeat Step 2-4 – 2.10



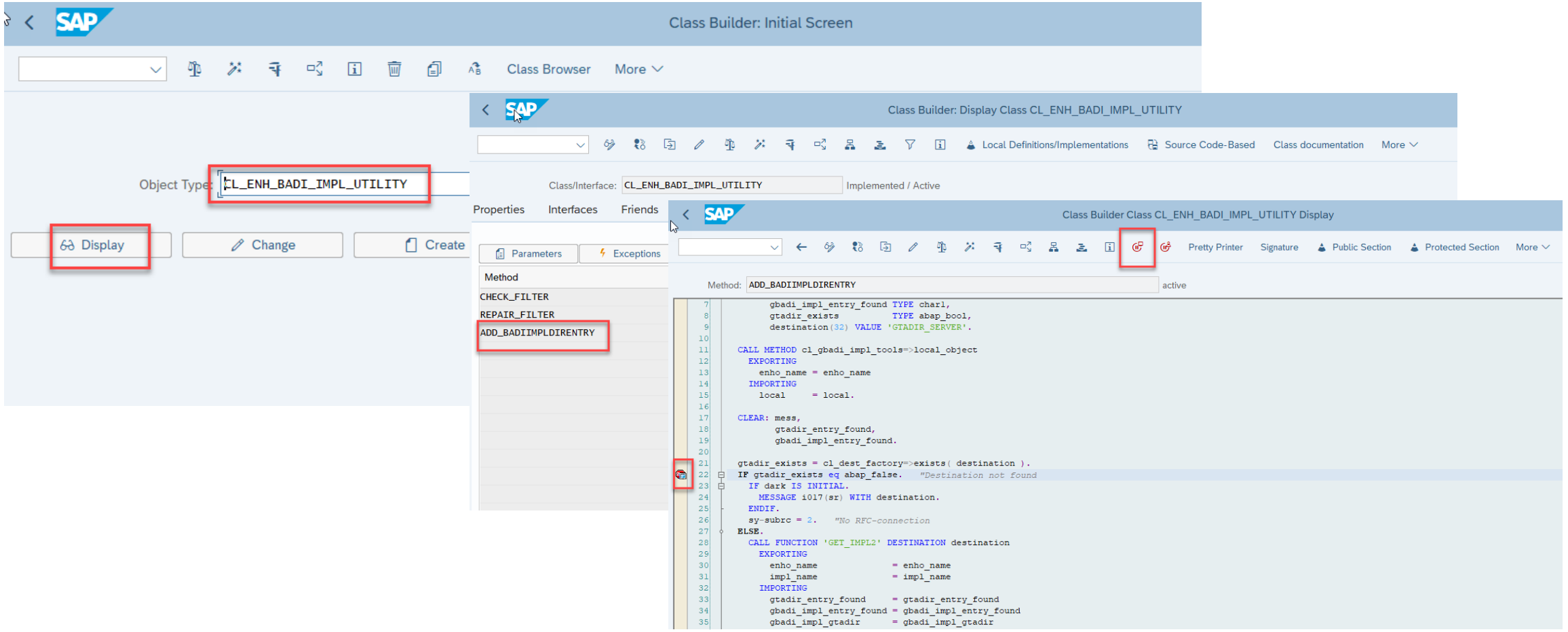
Known Issue: Failure of New BADI Importing

Symptom: If the user wants to update an existing New BADI by report *ZABAPGIT_STANDALONE*, there is no error message, but BADI ZPOF_GTT_LE_SHP_DLV_PROC_POI is not updated or is not created if it was deleted before.



Known Issue: Failure of New BADI Importing

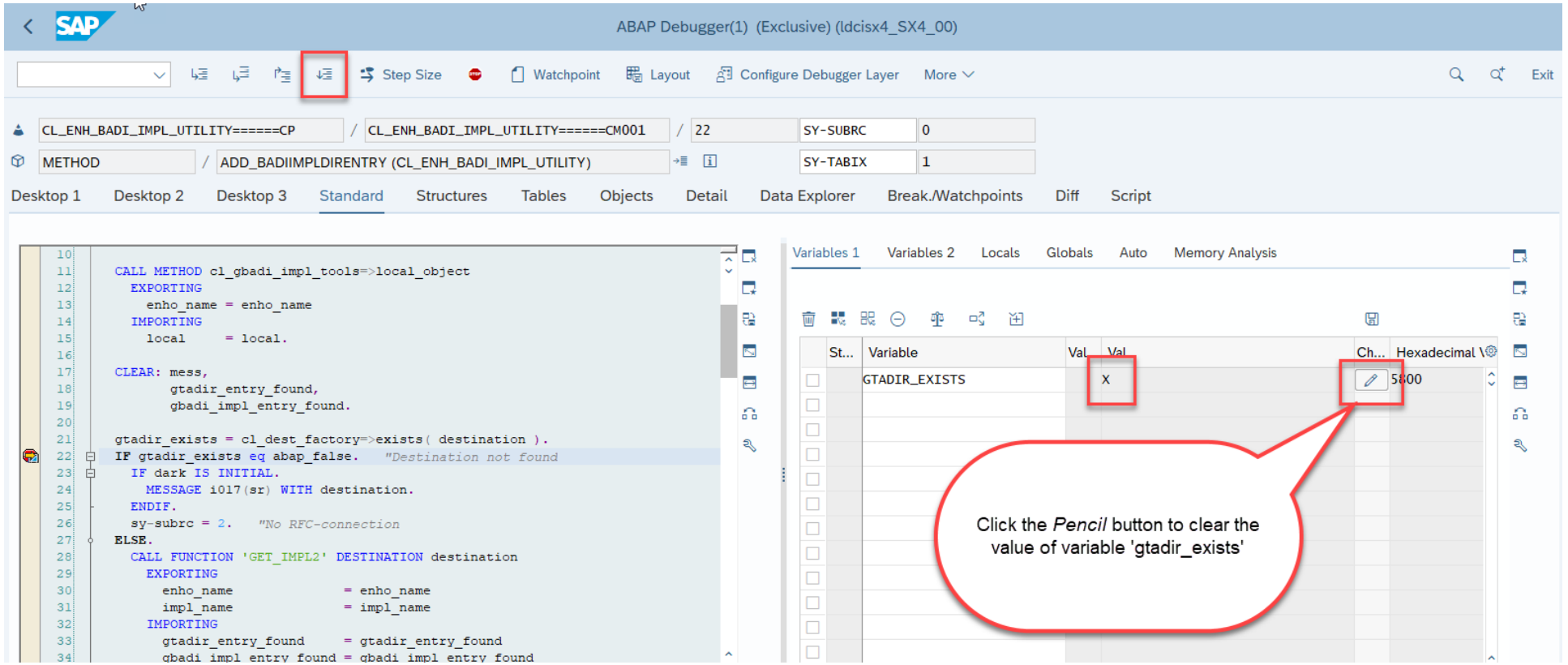
Solution: Set the breakpoint in Line 22 of class method: *CL_ENH_BADI_IMPL_UTILITY~ADD_BADIIMPLDIRENTRY* in transaction code *SE24*



Known Issue: Failure of New BADI Importing

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

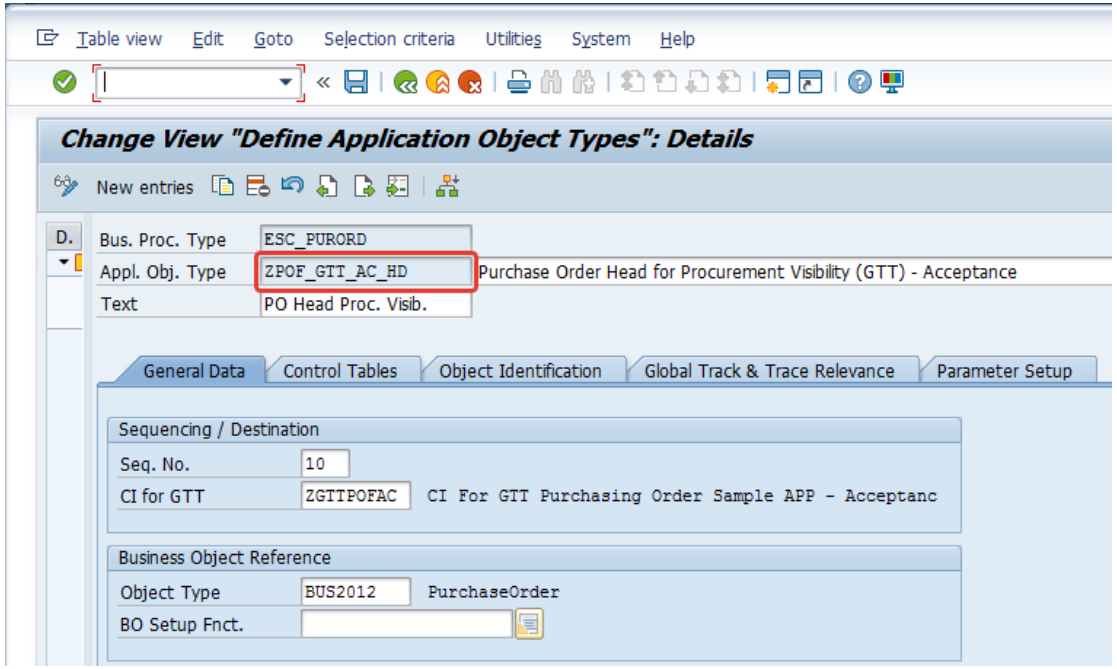
TIP: debugging needs related developer authorization for which you should ask your system administrator

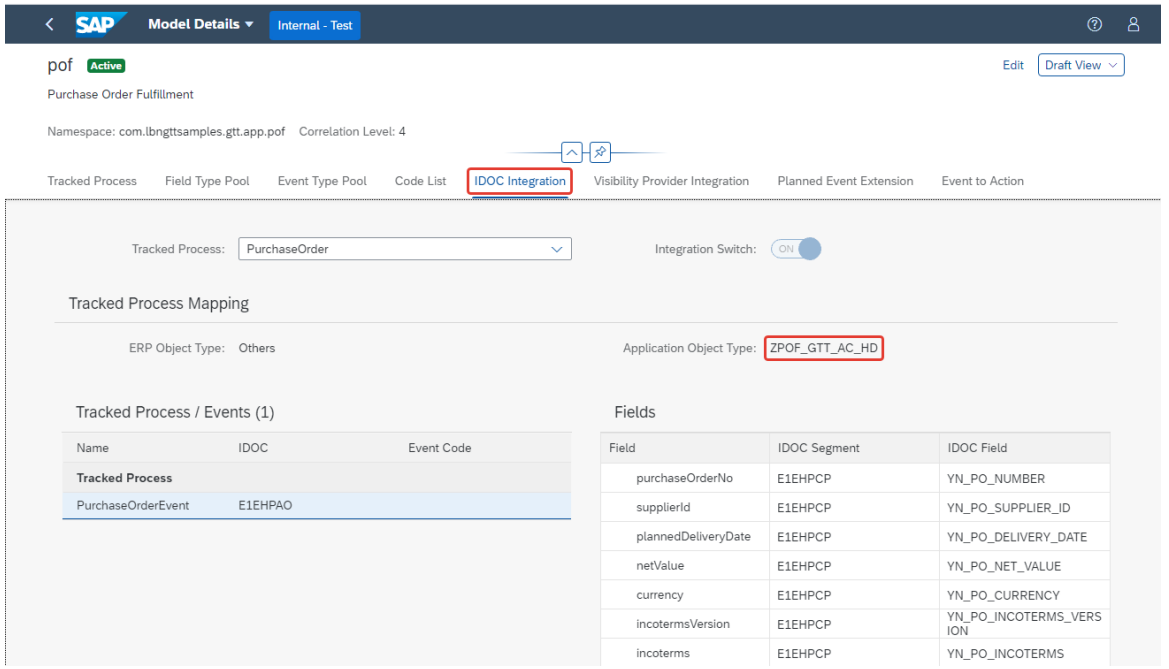


D) Configuration and Coding Guide - Advanced

1: Maintain AOT type

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

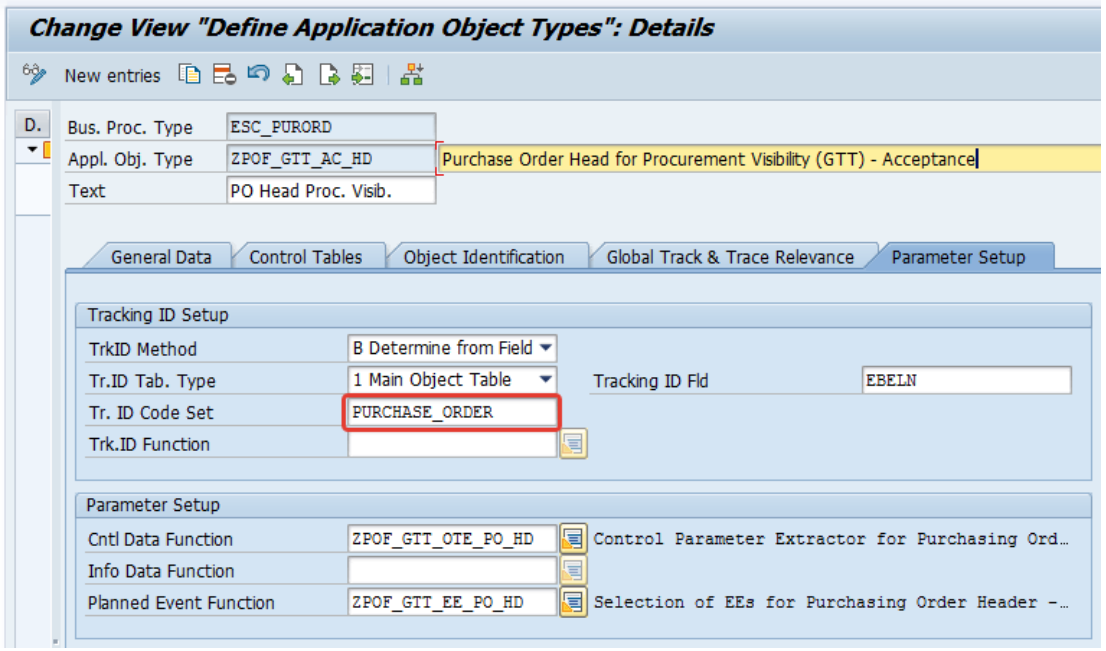


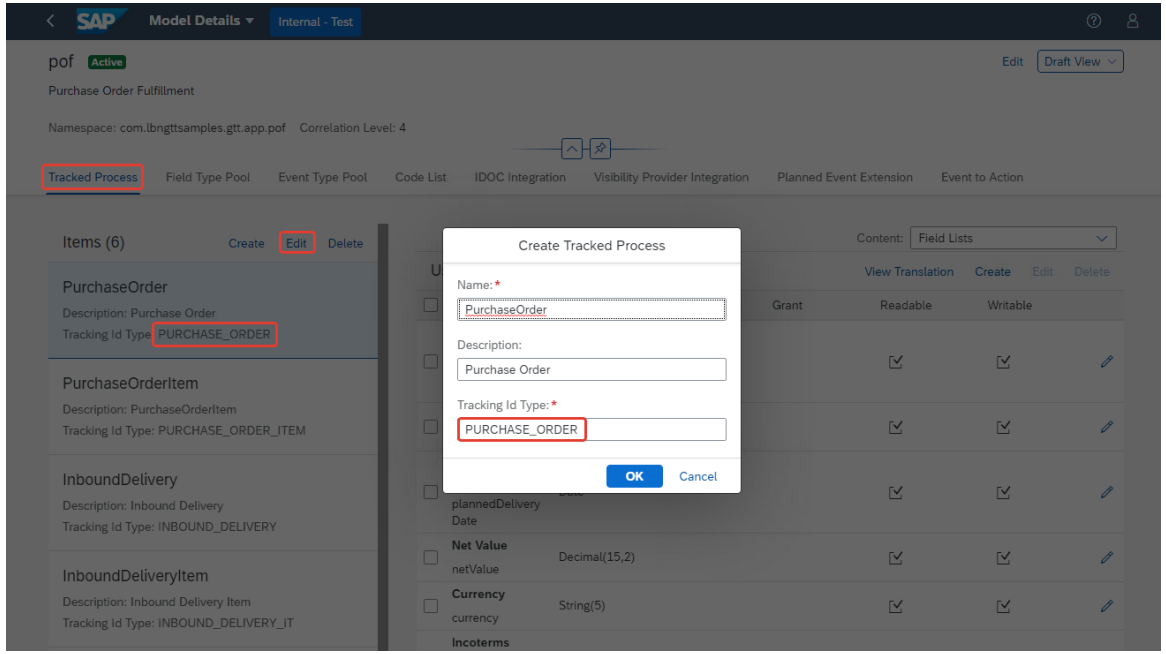


2: Maintain Tracking ID Type

In the AOT you maintained, make sure the Tracking ID Type is the same as the name which is defined in the corresponding process type of the model in Manage Models app in GTT V2.

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





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

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

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

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

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 Procrocess
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_HD	Control Parameter Extractor for Outbound Delivery Header
ZGTT_OTE_DE_ITEM	Control Parameter Extractor for Outbound Delivery Item
ZGTT_OTE_SHP_HD	Control Parameter Extractor for Shipment Header
ZGTT_OTE_SO_HD	Control Parameter Extractor for Sales Order Header
ZPOF_GTT_OTE_DL_HD	Control Parameter Extractor for Inbound Delivery Header
ZPOF_GTT_OTE_DL_ITEM	Control Parameter Extractor for Inbound Delivery Item
ZPOF_GTT_OTE_PO_HD	Control Parameter Extractor for Purchasing Order Header
ZPOF_GTT_OTE_PO_ITEM	Control Parameter Extractor for Purchasing Order Item
ZPOF_GTT_OTE_SH_HD	Control Parameter Extractor for Shipment Header
ZSST_GTT_OTE_FO_HD	Control Parameter Extractor for Freight Order

4: Sample Codes for Track Purchase Orders App

4-1 To support the Track Purchase Orders 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_ITEM_GR	Actual event PO Item Goods Receipt
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA	Actual event PO Item Put Away
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG	Actual event PO 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_ITEM	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 Purchase Orders App

4-2 To support the Track Purchase Orders App, the sample codes cover the following cases by function group ZPOF_GTT:

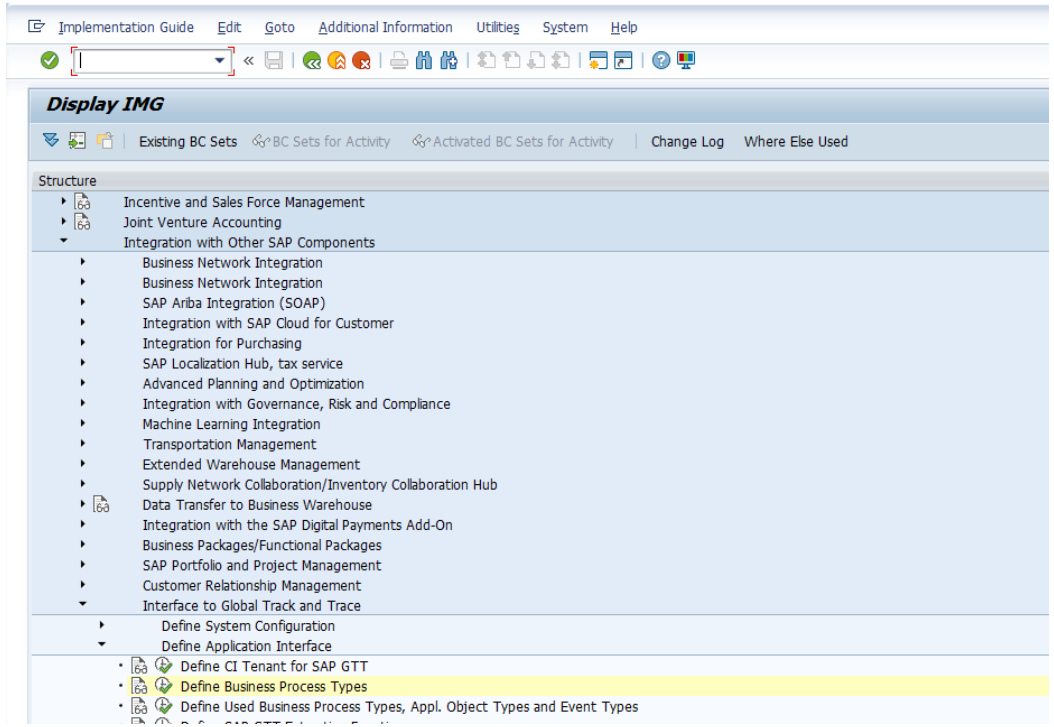
Category	Business Process Type	Function Module Name	Description
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_HDR_REL	Appl. Object Type Relevance for Inbound Delivery Header
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM_REL	Appl. Object Type Relevance for Inbound Delivery Item
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_ITEM_GR_REL	Relevance function for Actual event PO Item Goods Receipt
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA_REL	Relevance function for Actual event PO Item Put Away
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG_REL	Relevance function for Actual event PO 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_PURORD	ZPOF_GTT_CTP_DL_TO_PO	Cross TP Update from Delivery to Purchase Order
Cross TP Update Function	ESC_DELIV	ZPOF_GTT_CTP_SH_TO_DL	Cross TP Update from Shipment to Delivery

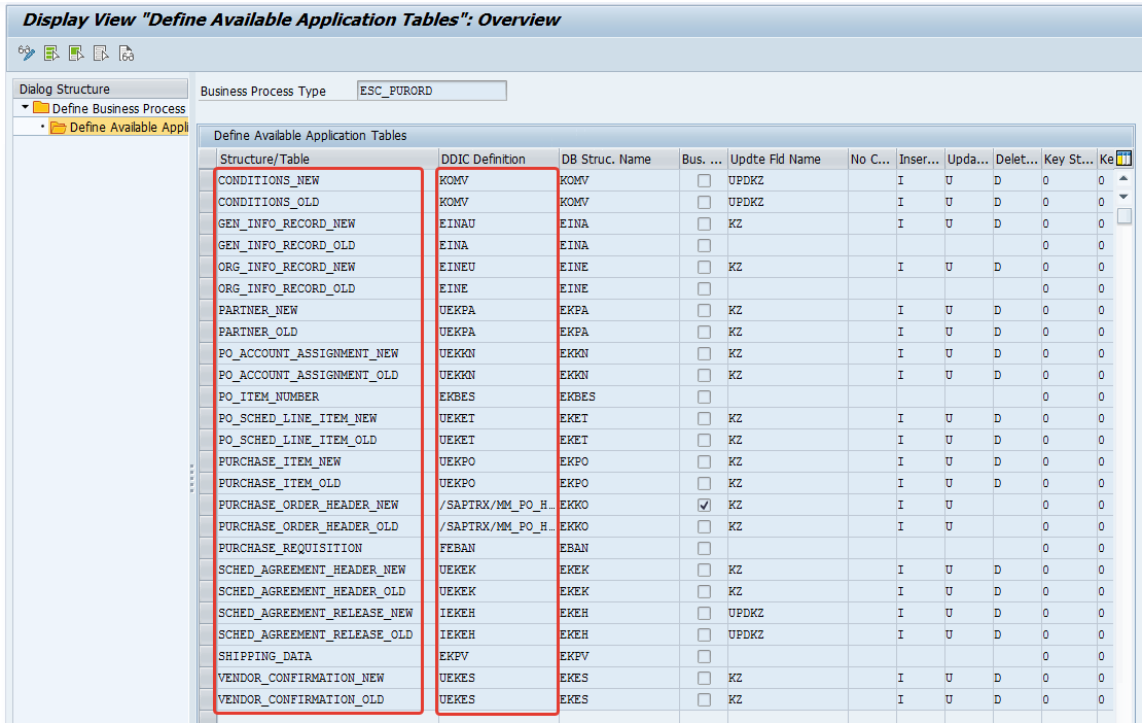
5: Available Contexts for the Extractors' Modules

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



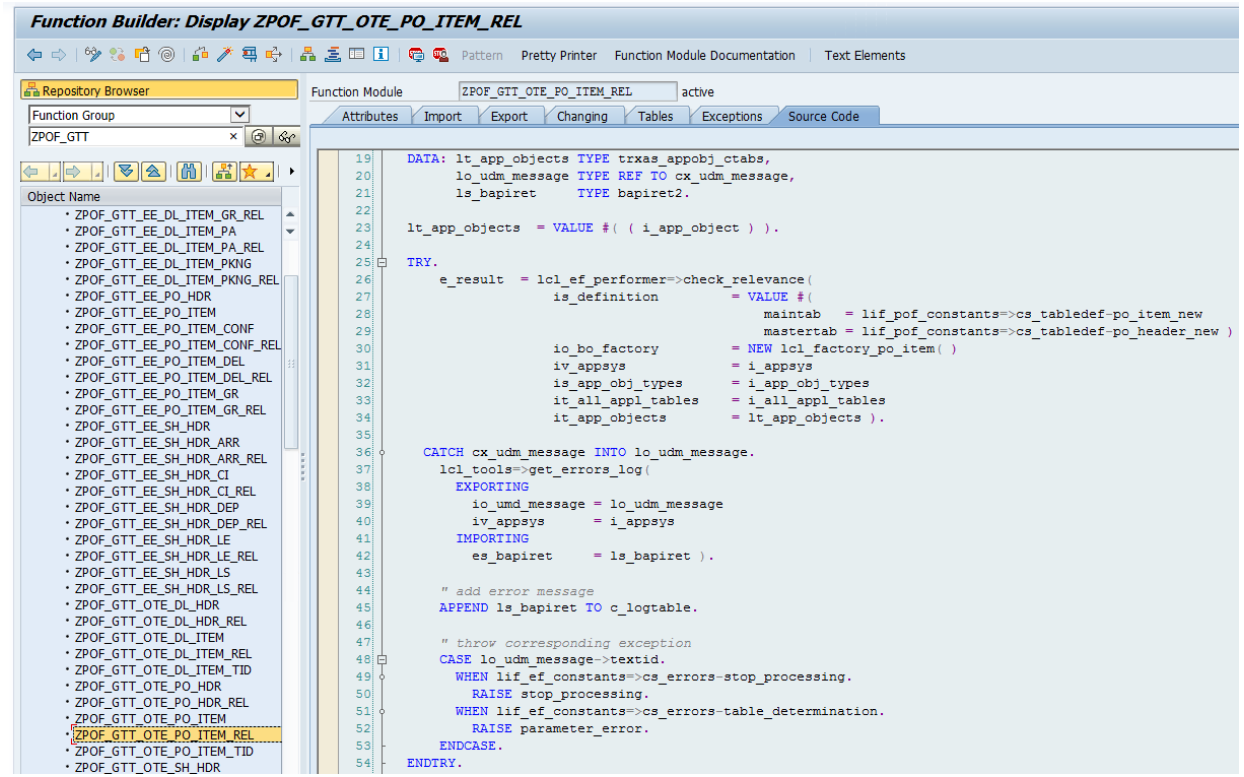


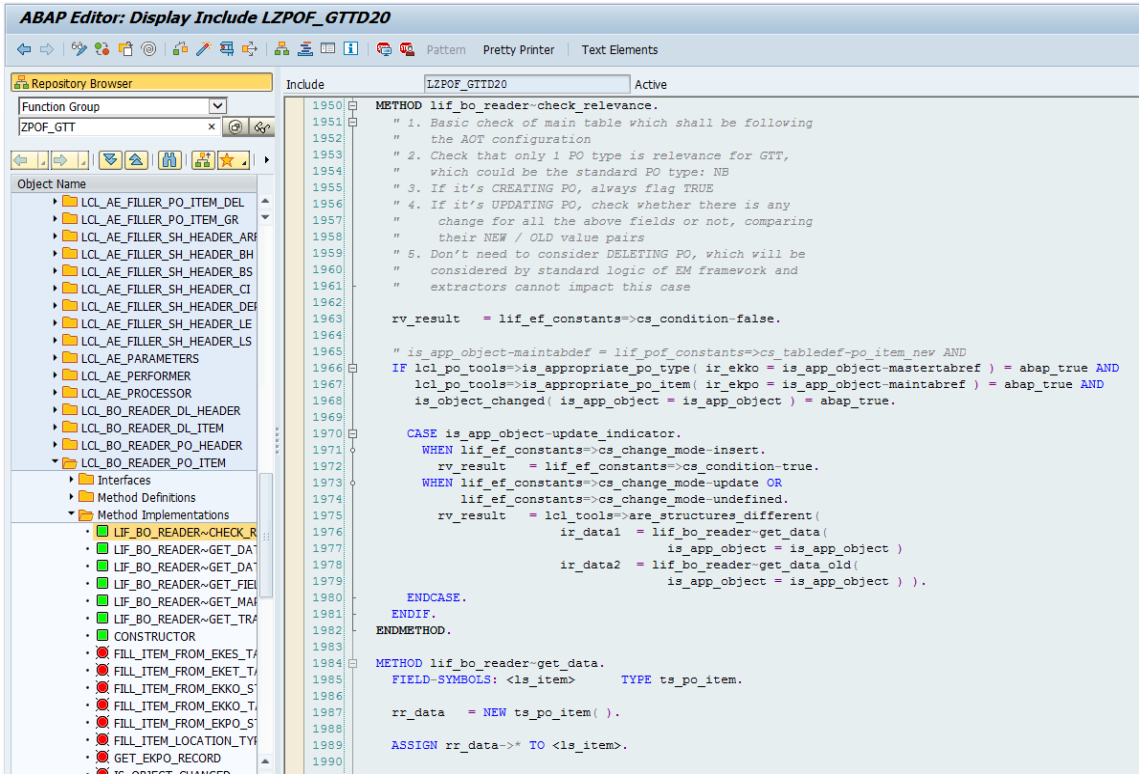
6: Coding Tips in the GTT Relevance Function Modules

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

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

See sample code of function: *ZPOF_GTT_OTE_PO_ITEM_REL*



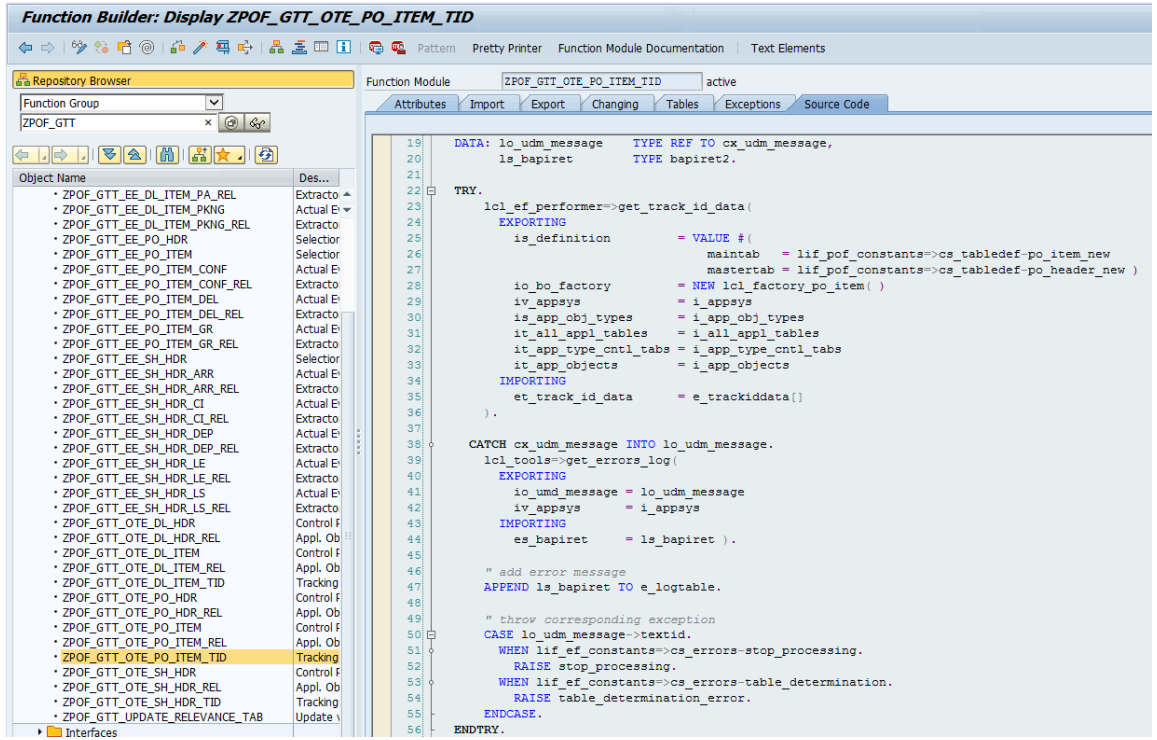


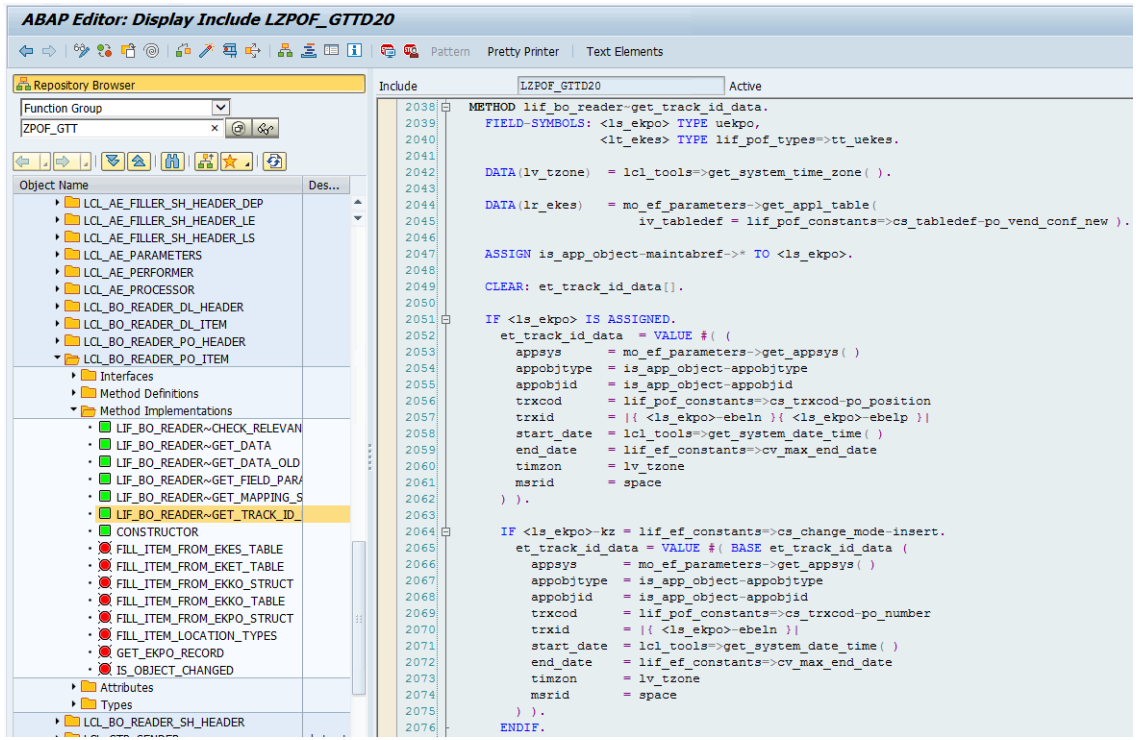
7: Coding Tips in the Tracking ID Function Modules

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

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

See sample code of function: *ZPOF_GTT_OTE_PO_ITEM_TID*





8: Coding Tips in the Control Parameter Function Modules

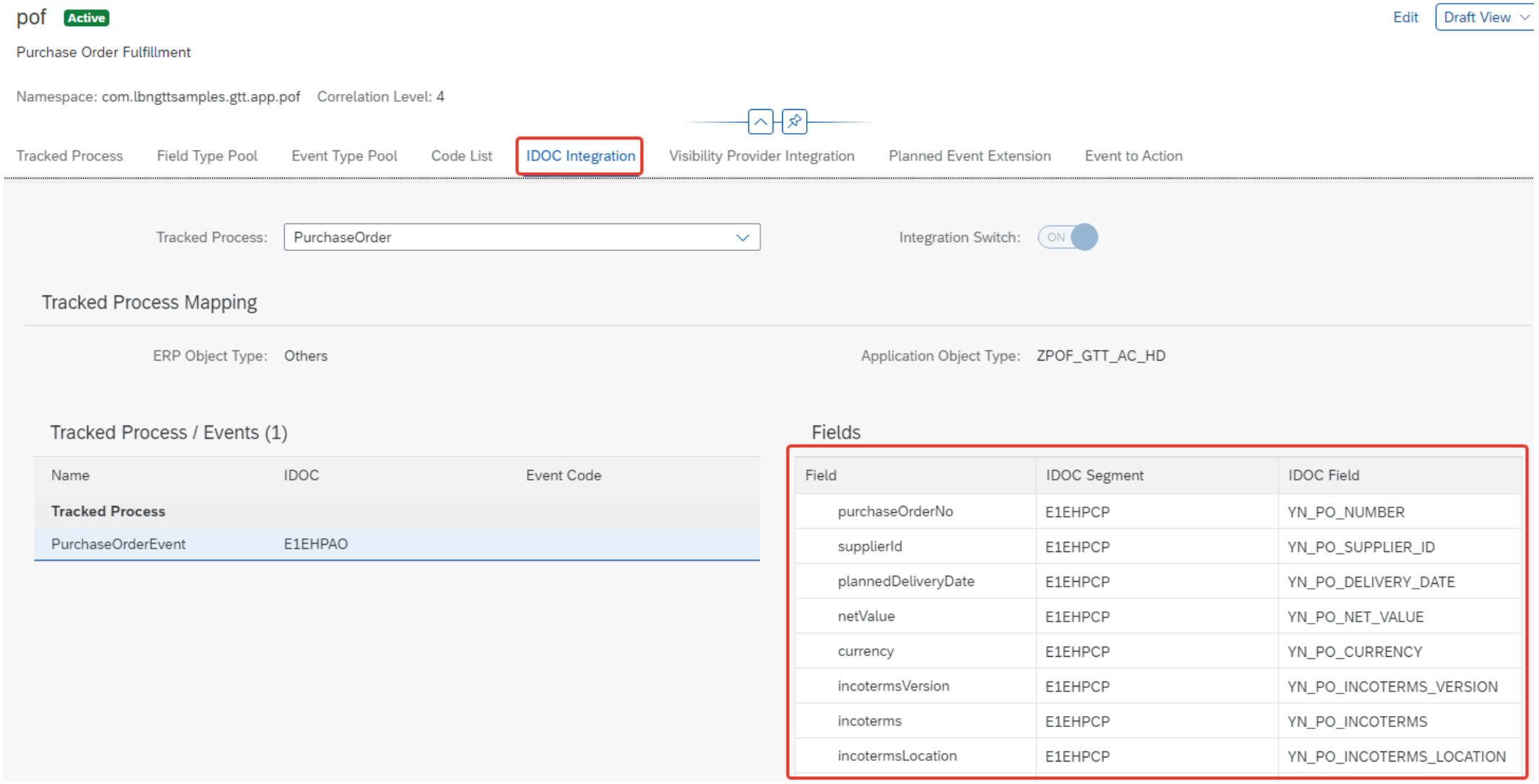
To customize the Control Parameter function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_CONTROL_DATA*.
3. GTT v2 asks for full transport for all the control parameters, which means that all the fields needs to be extracted in all cases, no matter whether their values have been changed.
4. To fill up the composition (table) fields defined in Manage 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 the key field using invalid values, for which key attribute has been checked in Manage Model app. It's not recommended to fill a code list type field to clear a composition even if it's a key field.
6. The field with fixed name 'ACTUAL_BUSINESS_DATETIME' and 'ACTUAL_BUSINESS_TIMEZONE' are mandatory fields to be transported for event handling sequencing in GTT V2.
7. In Manage Models app, click tab *IDOC Integration* to map the parameter names and model field names.
8. For DATE or DATETIME fields, when the source value is initial like '00000000' '000000000000000', then please ensure to only enable *PARAMNAME and PARAMINDEX in the extractor code, not enable VALUE for IDOC sending.*
9. For Amount field which has reference currency, please ensure to call BAPI 'BAPI_CURRENCY_CONV_TO_EXTERNAL' using the reference currency to make the amount tracked correctly by GTT v2. The BAPI will output the conversion result in 4 decimals as fixed, which needs additional rounding in the extractor if the corresponding field defined in the tracking model is of less then 4 decimals.

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 Manage Models app in IDOC Integration section:



8: Coding Tips in the Control Parameter Function Modules

Main logic of Purchase Order Item is implemented in class LCL_BO_READER_PO_ITEM

Function Module ZPOF_GTT_OTE_PO_ITEM active

Attributes | Import | Export | Changing | Tables | Exceptions | Source Code

```
19   DATA: lo_udm_message    TYPE REF TO cx_udm_message,
20         ls_bapiret        TYPE bapiret2.
21
22   TRY.
23       lcl_ef_performer=>get_control_data(
24         EXPORTING
25           is_definition         = VALUE #(
26                                     maintab    = lif_pof_constants=>cs_tabledef-po_item_new
27                                     mastertab = lif_pof_constants=>cs_tabledef-po_header_new )
28           io_bo_factory         = NEW lcl_factory_po_item( )
29           iv_appsys             = i_appsys
30           is_app_obj_types      = i_app_obj_types
31           it_all_appl_tables    = i_all_appl_tables
32           it_app_type_cntl_tabs = i_app_type_cntl_tabs
33           it_app_objects        = i_app_objects
34         CHANGING
35           ct_control_data       = e_control_data[] ).
36
37     CATCH cx_udm_message INTO lo_udm_message.
38       lcl_tools=>get_errors_log(
39         EXPORTING
40           io_umd_message = lo_udm_message
41           iv_appsys      = i_appsys
42         IMPORTING
43           es_bapiret     = ls_bapiret ).
44
45       " add error message
46       APPEND ls_bapiret TO e_logtable.
47
48       " throw corresponding exception
49       CASE lo_udm_message->textid.
50         WHEN lif_ef_constants=>cs_errors-stop_processing.
51           RAISE stop_processing.
52         WHEN lif_ef_constants=>cs_errors-table_determination.
53           RAISE table_determination_error.
54       ENDCASE.
55   ENDTRY.
56 ENDFUNCTION.
```

ABAP Editor: Display Include LZPOF_GTTD20

Pattern | Pretty Printer | Text Elements

Repository Browser

Function Group: ZPOF_GTT

Object Name

- LCL_AE_PROCESSOR
- LCL_BO_READER_DL_HEADER
- LCL_BO_READER_DL_ITEM
- LCL_BO_READER_PO_HEADER
- LCL_BO_READER_PO_ITEM
  - Interfaces
  - Method Definitions
  - Method Implementations
    - LIF_BO_READER~CHECK_RELEVAI
    - LIF_BO_READER~GET_DATA
    - LIF_BO_READER~GET_DATA_OLD
    - 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_EKKO_TABLE
    - FILL_ITEM_FROM_EKPO_STRUCT
    - FILL_ITEM_LOCATION_TYPES
    - GET_EKPO_RECORD
    - IS_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

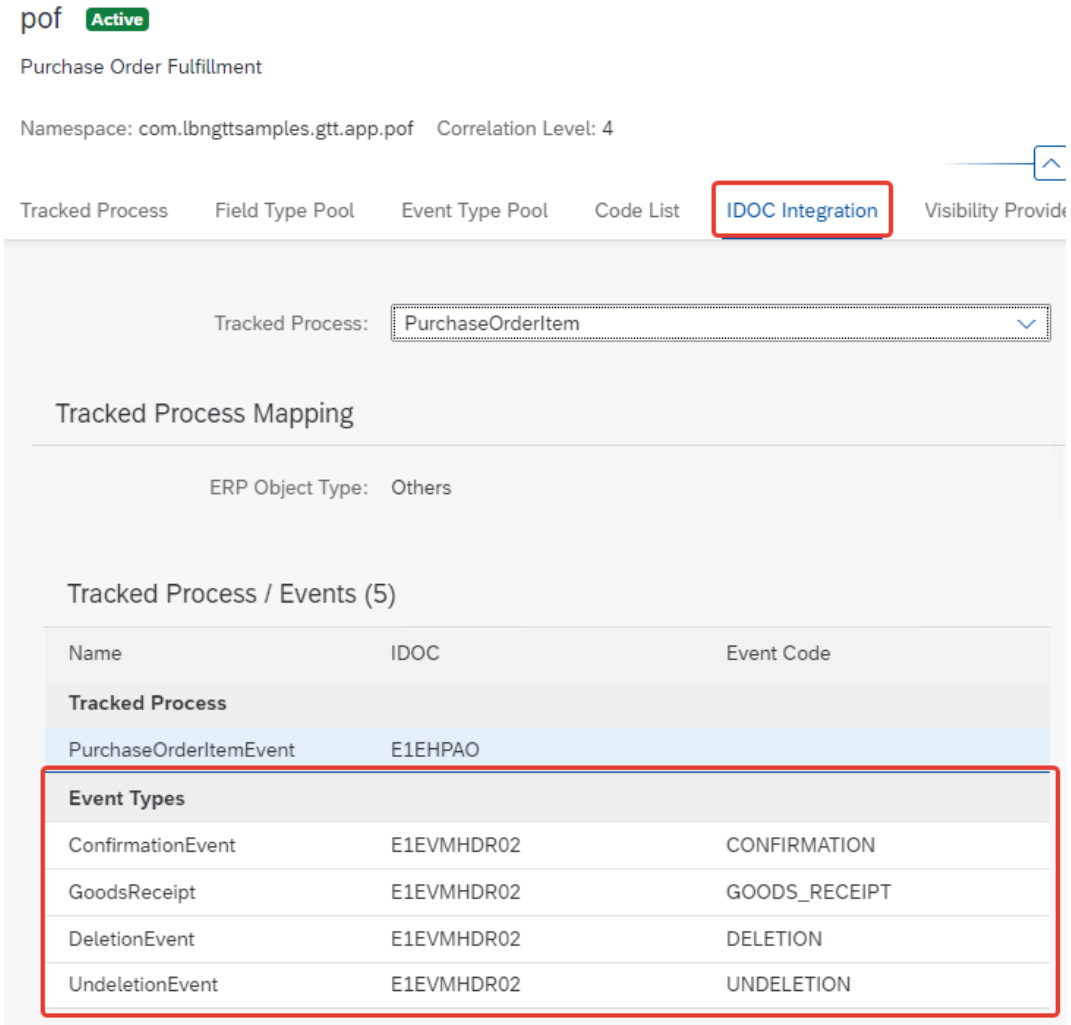
Include: LZPOF_GTTD20 Active

```
1984   METHOD lif_bo_reader~get_data.
1985     FIELD-SYMBOLS: <ls_item>      TYPE ts_po_item.
1986
1987     rr_data   = NEW ts_po_item( ).
1988
1989     ASSIGN rr_data->* TO <ls_item>.
1990
1991     fill_item_from_ekko_struct(
1992       EXPORTING
1993         ir_ekko       = is_app_object-mastertabref
1994       CHANGING
1995         cs_po_item    = <ls_item> ).
1996
1997     fill_item_from_ekpo_struct(
1998       EXPORTING
1999         ir_ekpo       = is_app_object-maintabref
2000       CHANGING
2001         cs_po_item    = <ls_item> ).
2002
2003     fill_item_from_eket_table(
2004       EXPORTING
2005         ir_ekpo       = is_app_object-maintabref
2006         ir_eket       = mo_ef_parameters->get_appl_table(
2007                           iv_tabledef = lif_pof_constants=>cs_tabledef-po_sched_new )
2008       CHANGING
2009         cs_po_item  = <ls_item> ).
2010
2011     fill_item_from_ekes_table(
2012       EXPORTING
2013         ir_ekpo    = is_app_object-maintabref
2014         ir_ekes    = mo_ef_parameters->get_appl_table(
2015                        iv_tabledef = lif_pof_constants=>cs_tabledef-po_vend_conf_new )
2016       CHANGING
2017         cs_po_item = <ls_item> ).
2018
2019     fill_item_location_types(
2020       CHANGING
2021         cs_po_item = <ls_item> ).
2022   ENDMETHOD.
```

9: Coding Tips in the Planned Event Function Modules

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

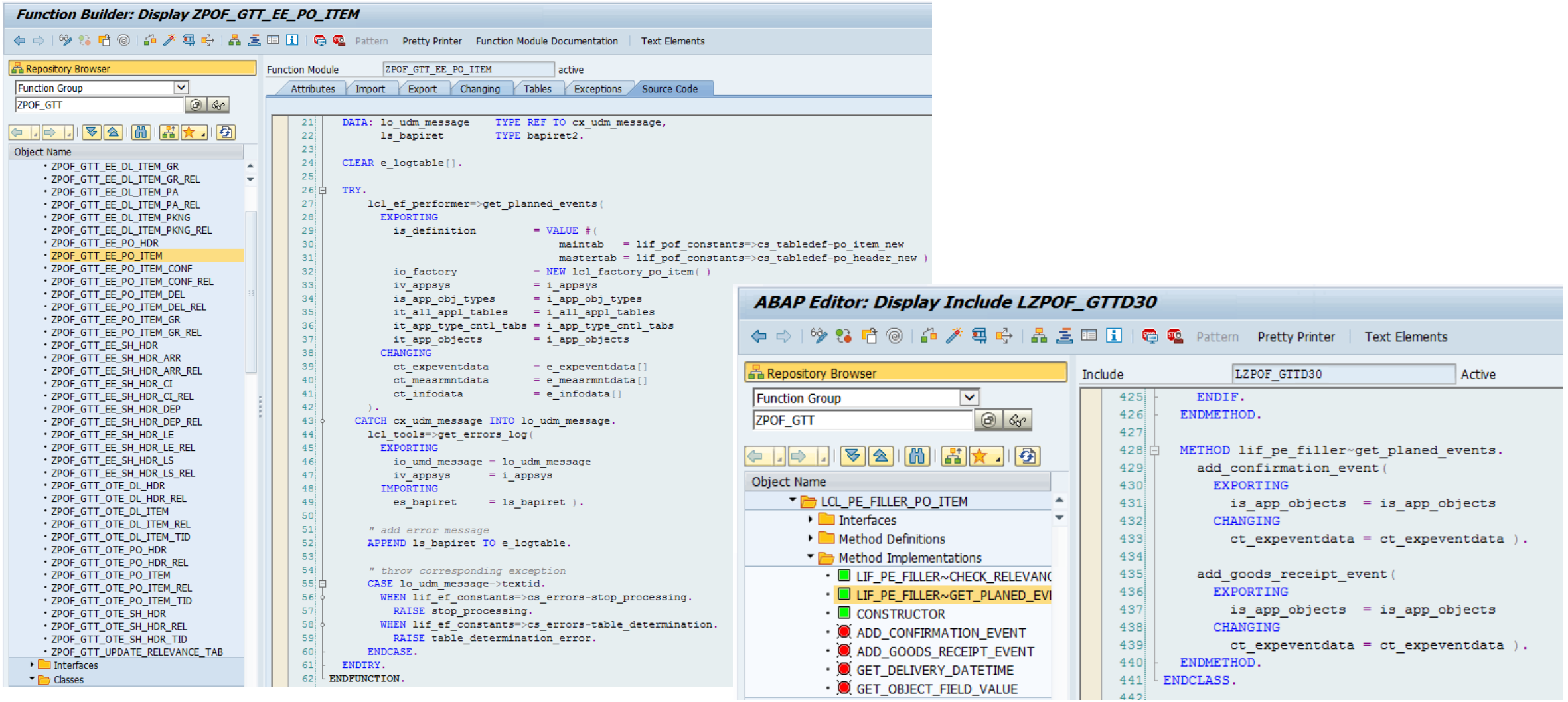
1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. GTT v2 asks for full transport for all the planned events, which means that all the events needs to be extracted in all cases, no matter whether their values have been changed.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT_EXP_DATETIME* is optional, but need to be filled with relevant time zone *EVT_EXP_TZONE* together if it needs to be transported.
6. The field *LOC_ID1* is optional, but need to be filled with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* app in GTT V2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.



See sample code of function: *ZPOF_GTT_EE_PO_ITEM*

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



10: Coding Tips in the Event Data Function Modules

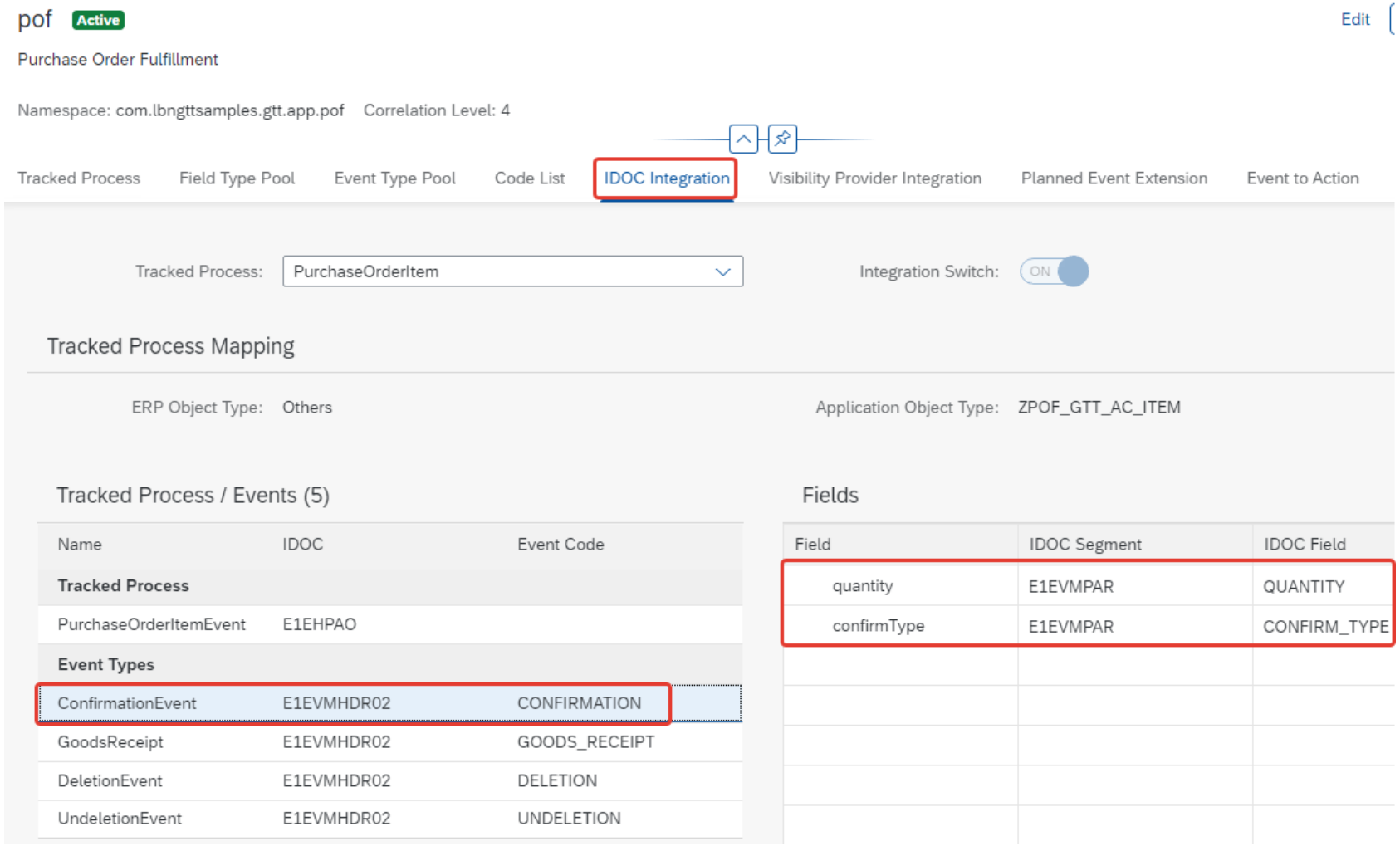
To customize the Event Data function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding Event Type.
2. Add customization logic to fill the output table *CT_TRACKINGHEADER, CT_TRACKLOCATION, C_EVENTID_MAP.*
3. If the event has user-defined fields in Manage Models app, fill the table *CT_TRACKPARAMETERS.*
4. If the event has reference table information, fill the table *CT_TRACKREFERENCES.*
5. The field *CT_TRACKINGHEADER-SRCCOD, SRCID, SRCTX* is used for event reason transport.
6. In Manage 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:



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

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

TRY.
    lcl_ae_performer=>get_event_data(
      EXPORTING
        is_definition         = VALUE #(
                                  maintab   = lif_pof_constants=>cs_tabledef-po_item_new
                                  mastertab = lif_pof_constants=>cs_tabledef-po_header_new )
        io_ae_factory         = NEW lcl_ae_factory_po_item_conf( )
        iv_appsys             = i_appsys
        is_event_type         = i_event_type
        it_all_appl_tables    = i_all_appl_tables
        it_event_type_cntl_tabs = i_event_type_cntl_tabs
        it_events             = i_events
      CHANGING
        ct_eventid_map        = c_eventid_map[]
        ct_trackingheader     = ct_trackingheader[]
        ct_tracklocation      = ct_tracklocation[]
        ct_trackreferences    = ct_trackreferences[]
        ct_trackparameters    = ct_trackparameters[]
    ).

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

    " add error message
    APPEND ls_bapiret TO ct_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 event_data_error.
    ENDCASE.
ENDTRY.
ENDFUNCTION.
```

ABAP Editor: Display Include LZPOF_GTTD40

Pattern | Pretty Printer | Text Elements

Repository Browser

Function Group: ZPOF_GTT

Object Name

- ZPOF_GTT_UPDATE_RELEVANCE_TAB
- Interfaces
- 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
    - Interfaces
    - Method Definitions
    - Method Implementations
      - LIF_AE_FILLER~CHECK_RELEVANC
      - LIF_AE_FILLER~GET_EVENT_DAT
      - CONSTRUCTOR
      - GET_CONFIRMATION_QUANTITY
      - GET_CONFIRMATION_QUANTITY_
      - HAS_CHANGES
      - IS_APPROPRIATE_CONF_CONTRC
      - 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

```
METHOD lif_ae_filler~get_event_data.
  DATA(lv_difference)   =  get_confirmation_quantity_diff(
                             is_events = is_events ).

  ct_trackingheader = VALUE #( BASE ct_trackingheader (
    language    = sy-langu
    trxid       = lcl_po_tools=>get_tracking_id_po_item(
                    ir_ekpo = is_events-maintabref )
    trxcod      = lif_pof_constants=>cs_trxcod-po_position
    evtcnt      = is_events-eventid
    evtid       = lif_pof_constants=>cs_milestone-po_confirmation
    evtdat      = sy-datum
    evttim      = sy-uzeit
    evtzon      = lcl_tools=>get_system_time_zone( )
  ) ).

  ct_eventid_map  = VALUE #( BASE ct_eventid_map (
    eventid     = is_events-eventid
    evtcnt      = is_events-eventid
  ) ).

  ct_tracklocation  = VALUE #( BASE ct_tracklocation (
    evtcnt      = is_events-eventid
    loccod      = lif_ef_constants=>cs_loc_types-plant
    locid1      = lcl_tools=>get_field_of_structure(
                    ir_struct_data = is_events-maintabref
                    iv_field_name  = 'WERKS' )
  ) ).

  " QUANTITY
  ct_trackparameters  = VALUE #( BASE ct_trackparameters (
    evtcnt      = is_events-eventid
    param_name  = lif_pof_constants=>cs_event_param-quantity
    param_value = lcl_tools=>get_pretty_value( iv_value = lv_difference )
  ) ).

  " CONFIRMATION TYPE
  ct_trackparameters  = VALUE #( BASE ct_trackparameters (
    evtcnt      = is_events-eventid
    param_name  = lif_pof_constants=>cs_event_param-confirm_type
    param_value = lif_pof_constants=>cs_relevance-ebtyp
  ) ).
ENDMETHOD.
```

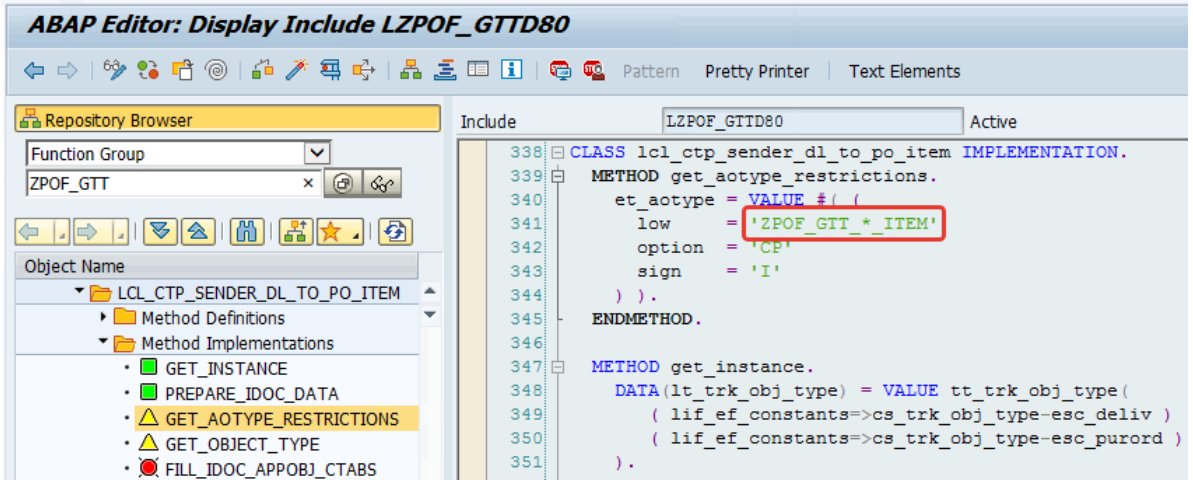
11: Enhancement Codes for Cross-process Tracking

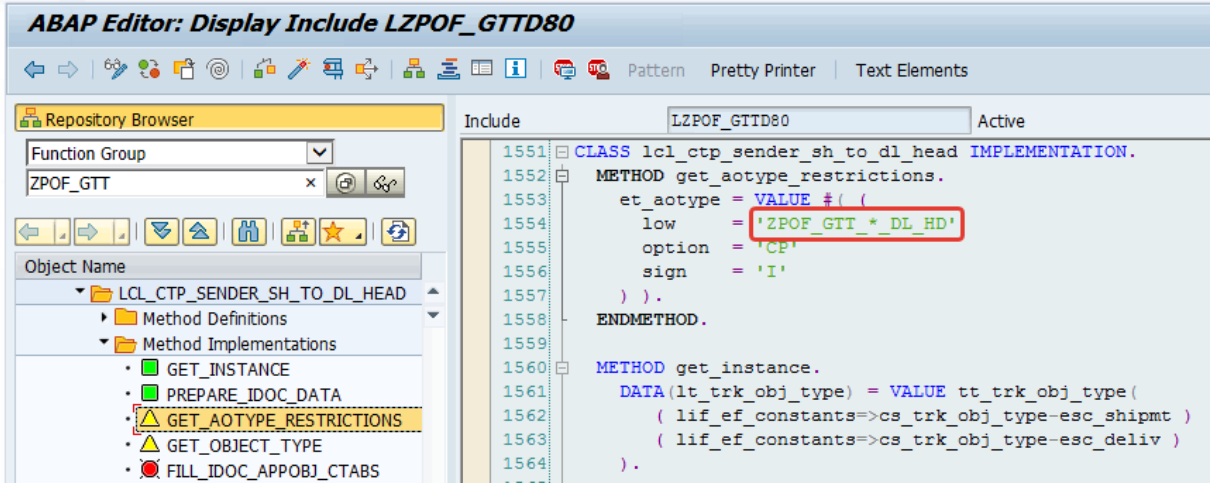
The Track Purchase Orders app asks for cross-processes tracking, which is used in below cases:

1. When the inbound delivery item process is updated and transported to GTT, the preceding purchasing order item process needs to be updated and transported to GTT.
2. 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, please update the below sample codes after downloading:

1. Update Purchase Order Item AOT type Mask in Method GET_AOTYPE_RESTRICTIONS of LCL_CTP_SENDER_DL_TO_PO_ITEM
2. 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





11: Enhancement Codes for Cross-process Tracking

The cross-process tracking scenarios cover the following:

Delivery Item -> Purchase Order Item:

1\ Inbound Delivery Item Composition (Full Transport)
- Case: Inbound Delivery Item Create / Delete
- Case: Inbound Delivery Create / Delete

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

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

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

2. IDOC sequencing issue

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

This issue will be covered in short future.

Follow us

www.sap.com/contactsap

THE BEST RUN SAP®