



# SAP Business Network Global Track and Trace **Track Shipments - SAP ERP Integration**

SAP Business Network for Logistics  
March 2021

PUBLIC

# Objectives



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

- Learn what prerequisite is necessary for SAP Business Network Global Track and Trace
- 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 Product Version

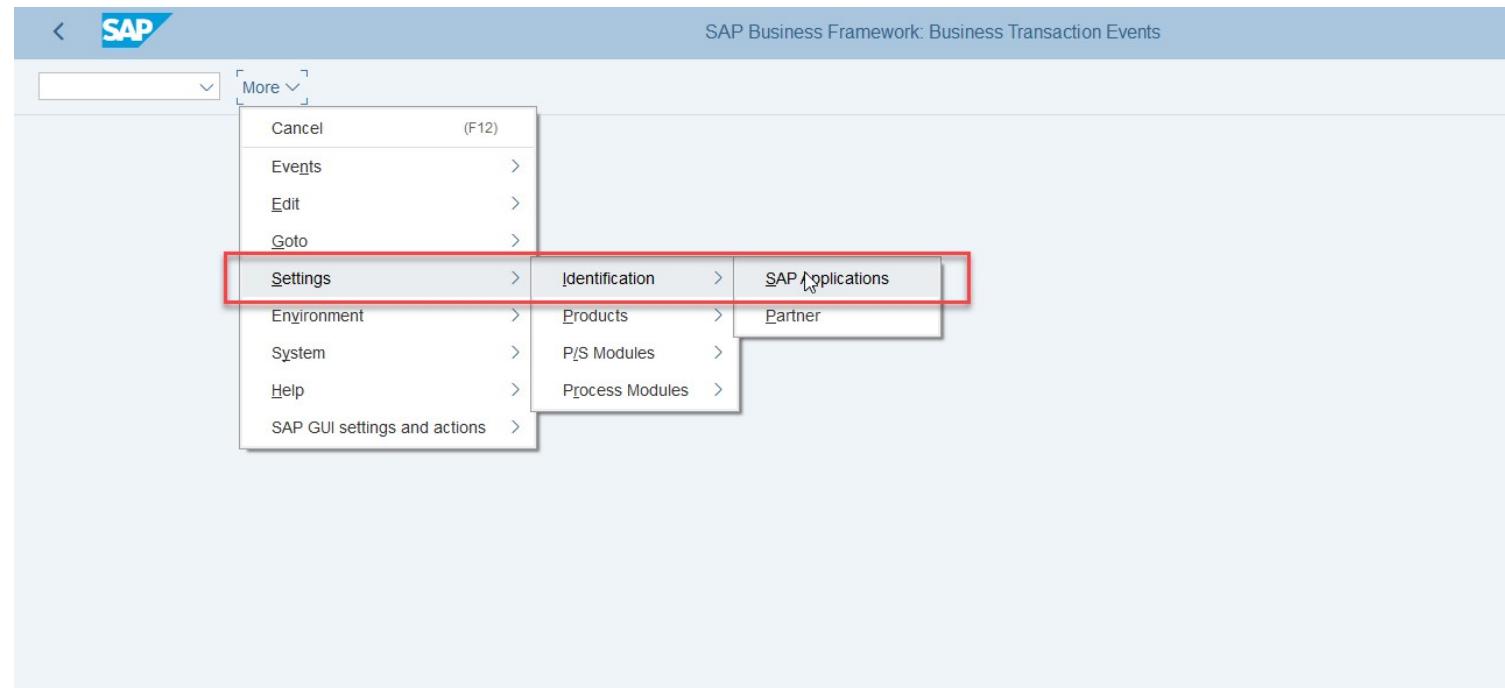
- 1-1: Make sure that you have met the requirements for the product version mentioned in the “[Prerequisites](#)” section of *Appendix one: Connect to SAP ERP in Administration Guide for Version 2*. You can find this guide at <http://help.sap.com/gtt>.
  
- 1-2: The ABAP codes on Github to support sample apps for SAP Business Network Global Track and Trace shall be implemented in SAP S/4HANA 1909 SP03 on premise or higher. Please note that the codes are not validated in its lower version or other ECC series of products, so you might need to do further adaptation work or build your own extractor.

## STEP 2: Log on the Development Client to Configure BTE

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

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

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

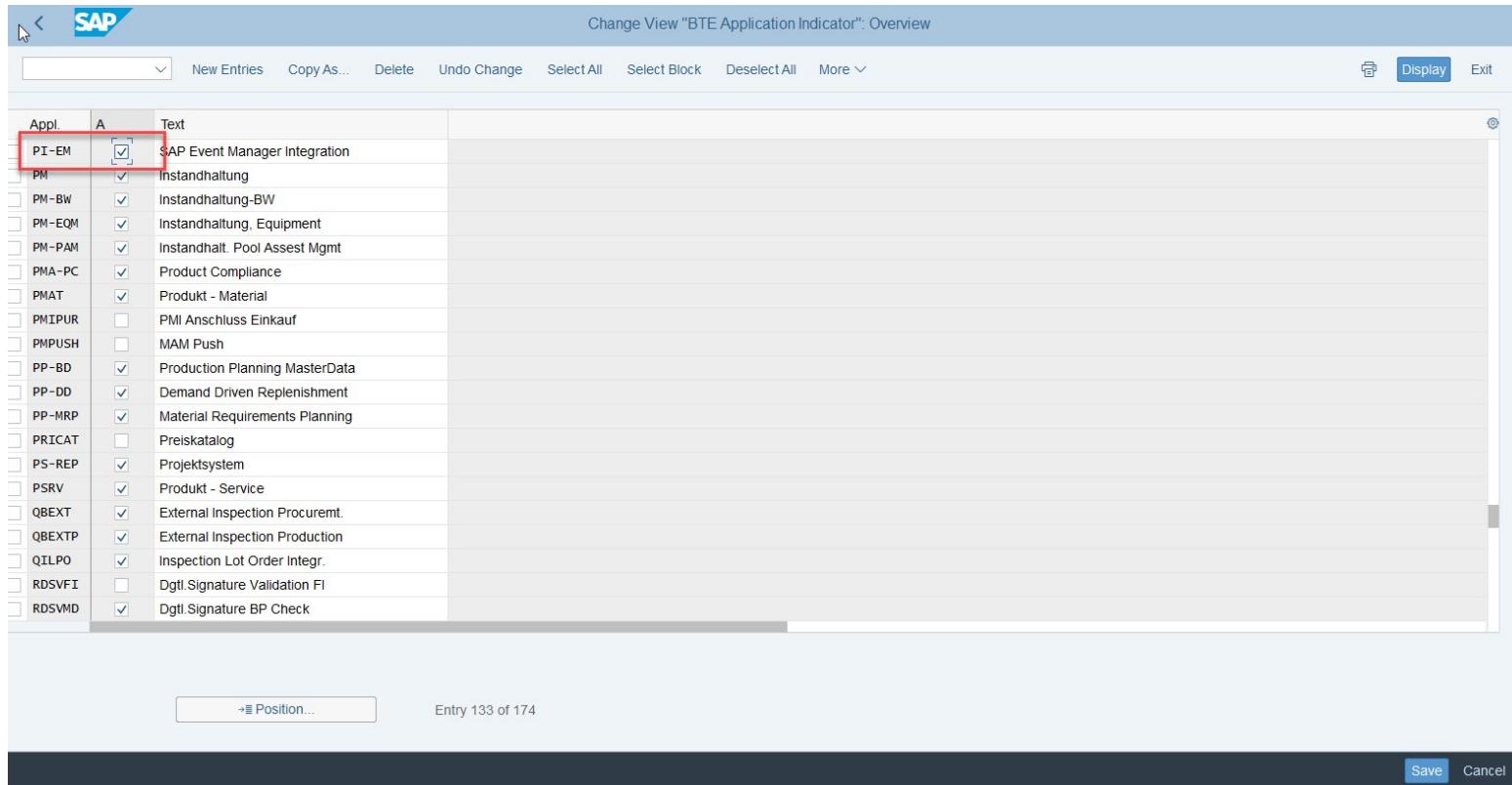


# STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**



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

# B) Configuration and Implementation

## - Basic

### B1. IDOC Configuration



# STEP 1: Define RFC Connection for SAP Business Network Global Track and Trace

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

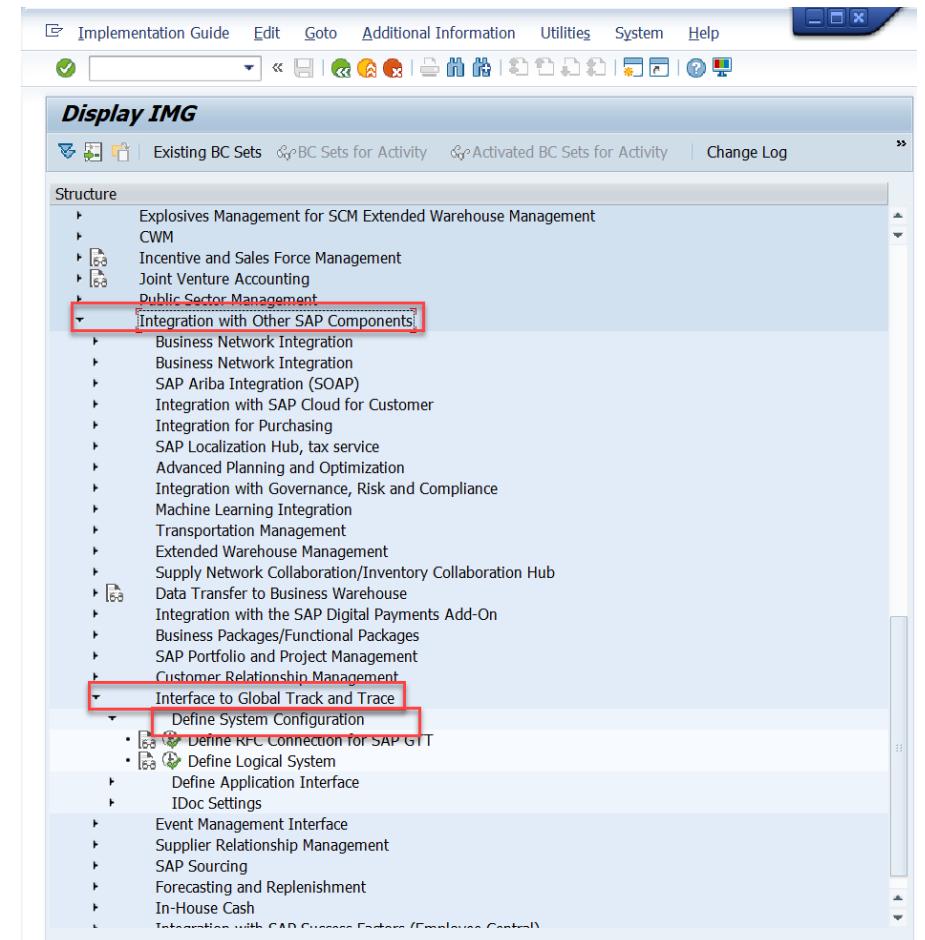
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

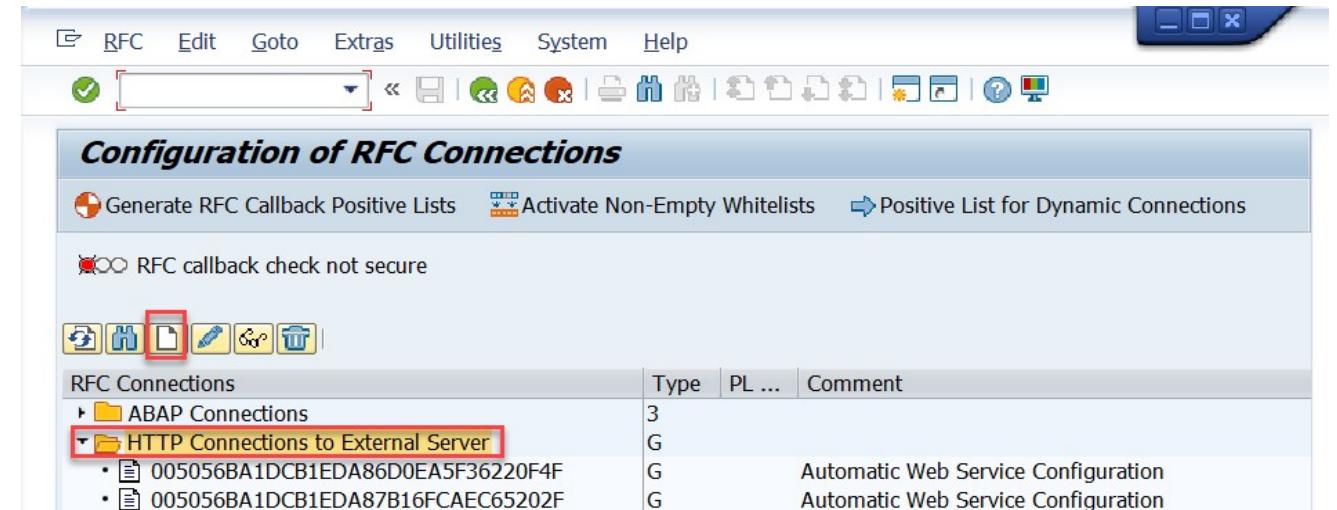
**Define RFC Connection for SAP GTT**



# STEP 1: Define RFC Connection for SAP Business Network Global Track and Trace

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 SAP Business Network Global Track and Trace

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://xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com>

**Host:** `xxxxxx.gtt-flp-lbnplatform.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`

**RFC Destination ZGTT\_SST\_FO\_EVENT\_ACC**

Connection Test

RFC Destination	ZGTT_SST_FO_EVENT_ACC
Connection Type	G HTTP Connection to External Server
Description	<p>Description 1: RFC for Events of SST Sample Application to Acceptance</p> <p>Description 2:</p> <p>Description 3:</p>

Administration    Technical Settings    Logon & Security    Special Options

**Target System Settings**

Host	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	Port	443
Path Prefix	/api/idoc/em/v1/Event		

**HTTP Proxy Options**

Global Configuration	
Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

RFC Destination	RFC Destination Description	Host	Path Prefix	Port
ZGTT_SST_FO_EVENT_ACC	RFC for Event of Tracking Shipments Sample Application	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/Event	443
ZGTT_SST_FO_TP_ACC	RFC for Tracked Process of Tracking Shipments Sample Application	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/TrackedProcess	443

# STEP 1: Define RFC Connection for SAP Business Network Global Track and Trace

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.

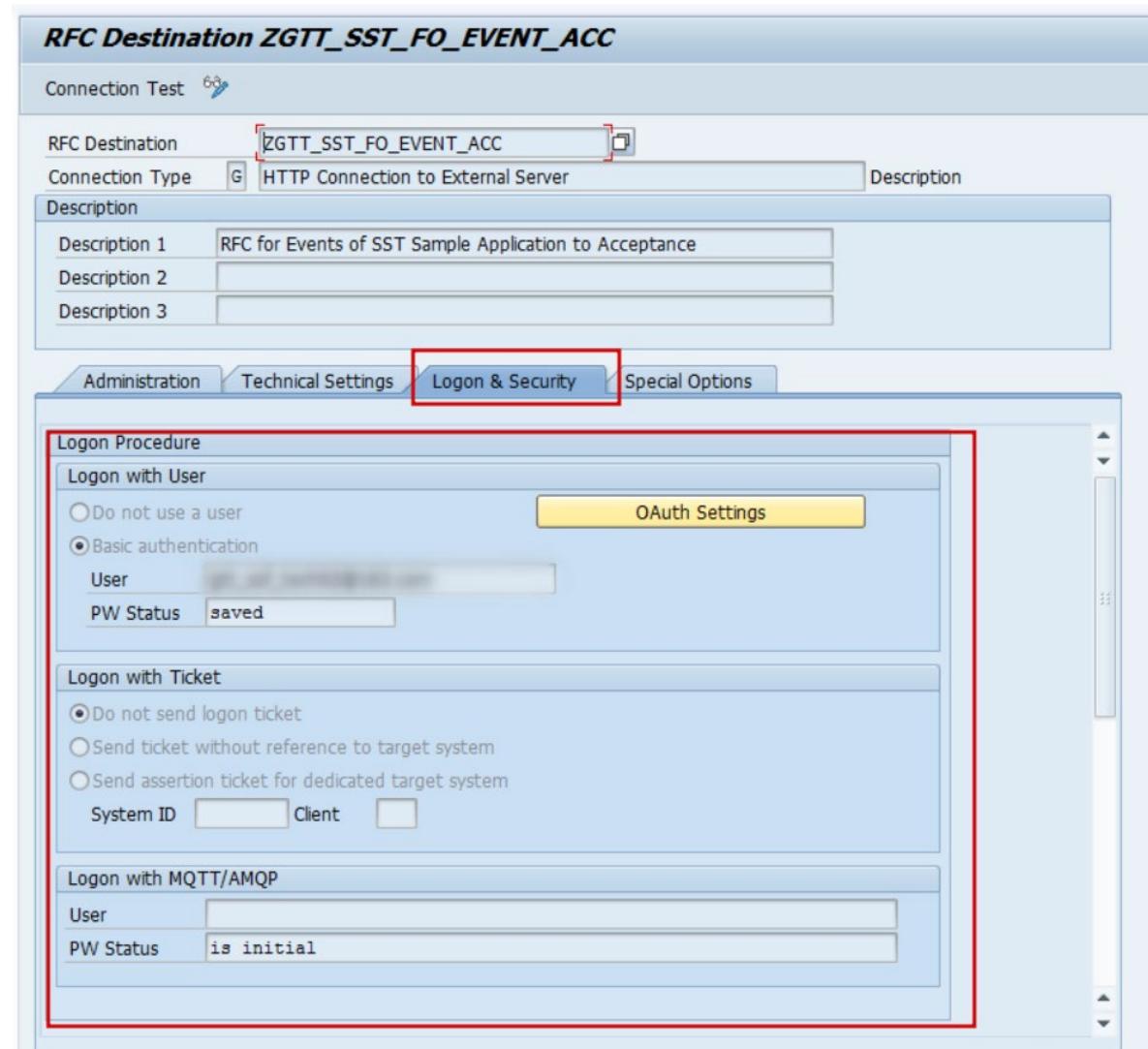
Also, SSL must be *Active*.

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

1-10: Save the configuration

**Caution:** You need to configure two RFC Connections:

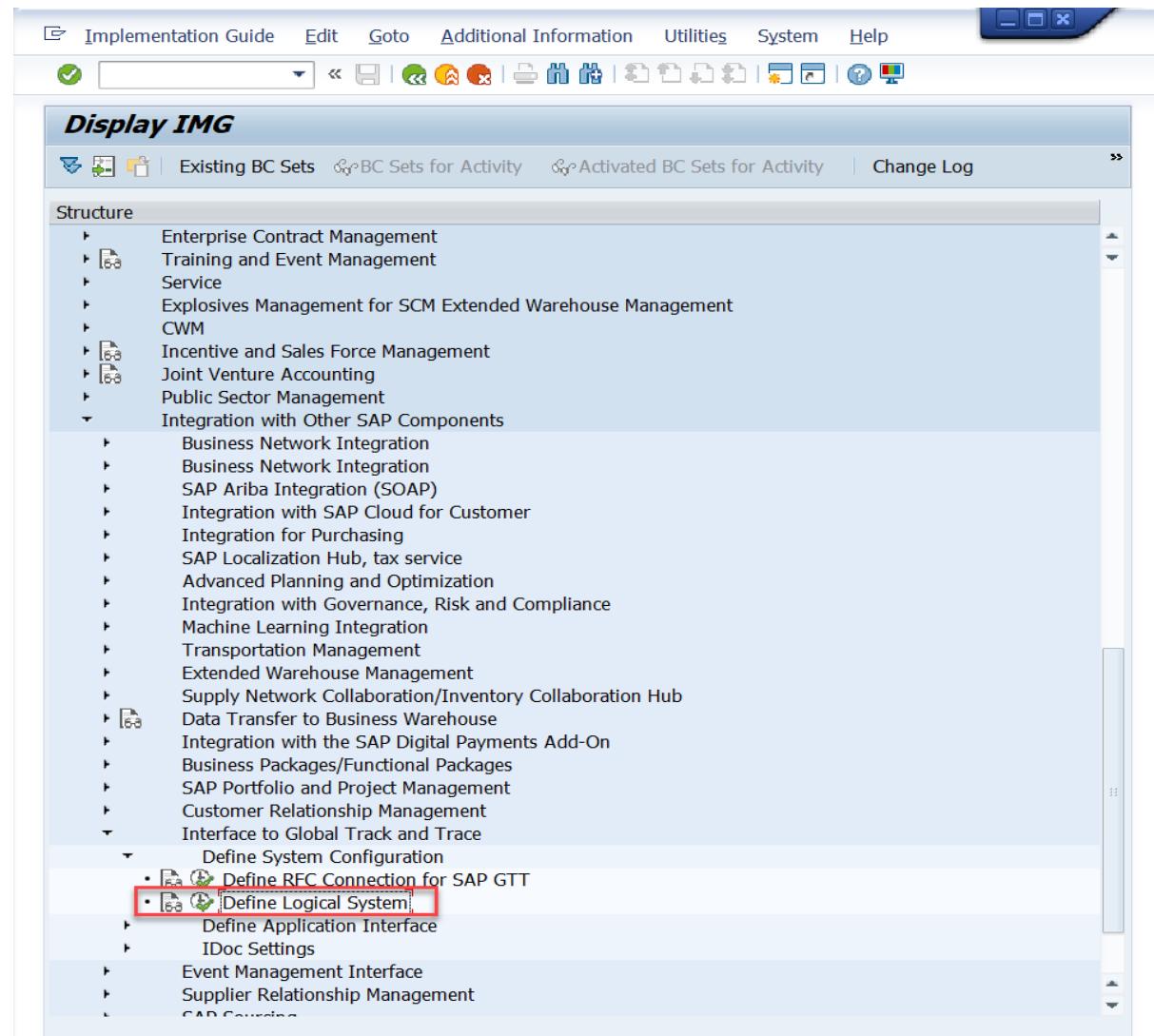
- one for event and
- the other for tracked process.



# STEP 2: Define Logical System

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

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



## STEP 2: Define Logical System

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

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

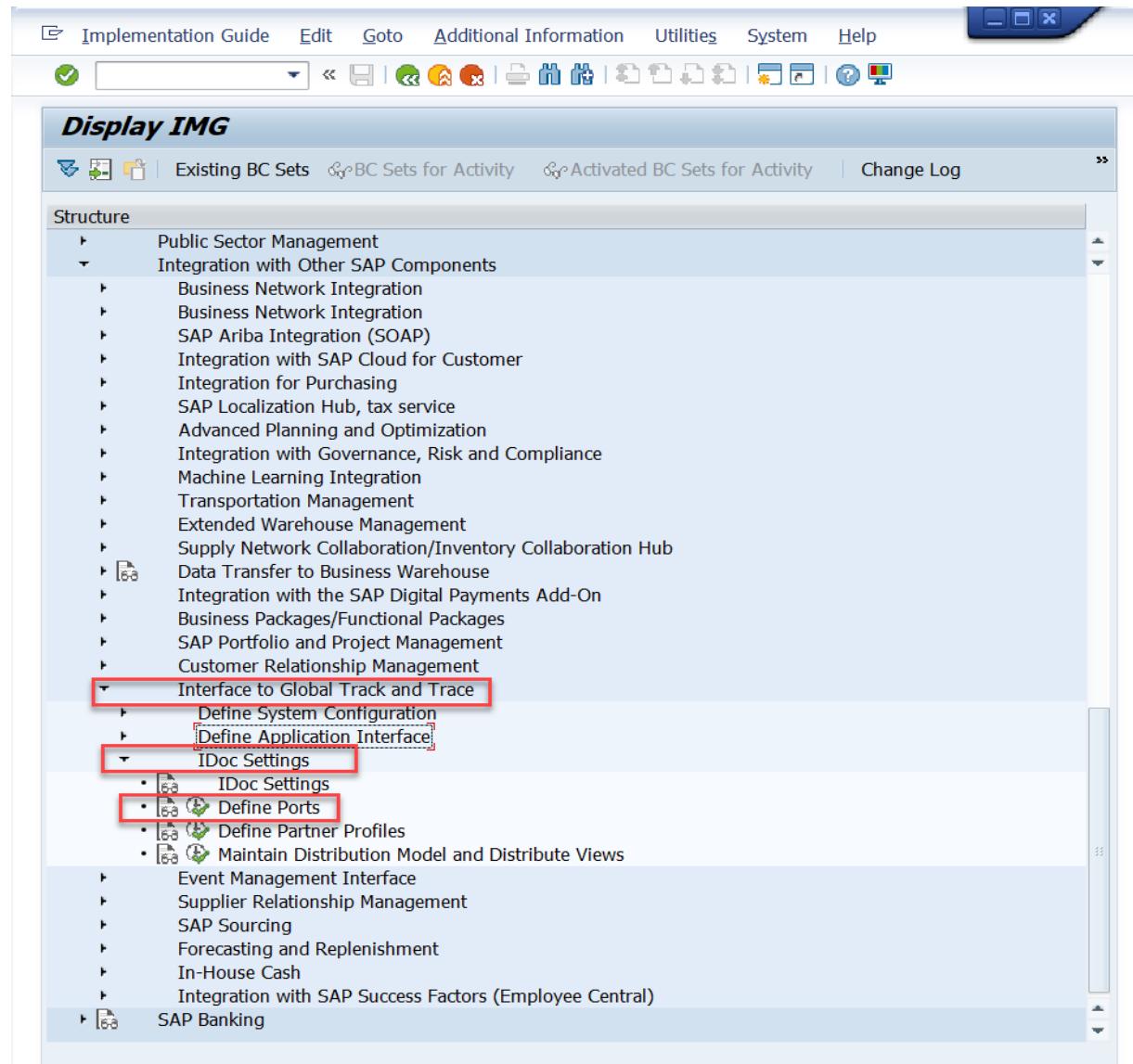
2-4: Save the configuration

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

# STEP 3: Define Ports

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

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



## STEP 3: Define Ports

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

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

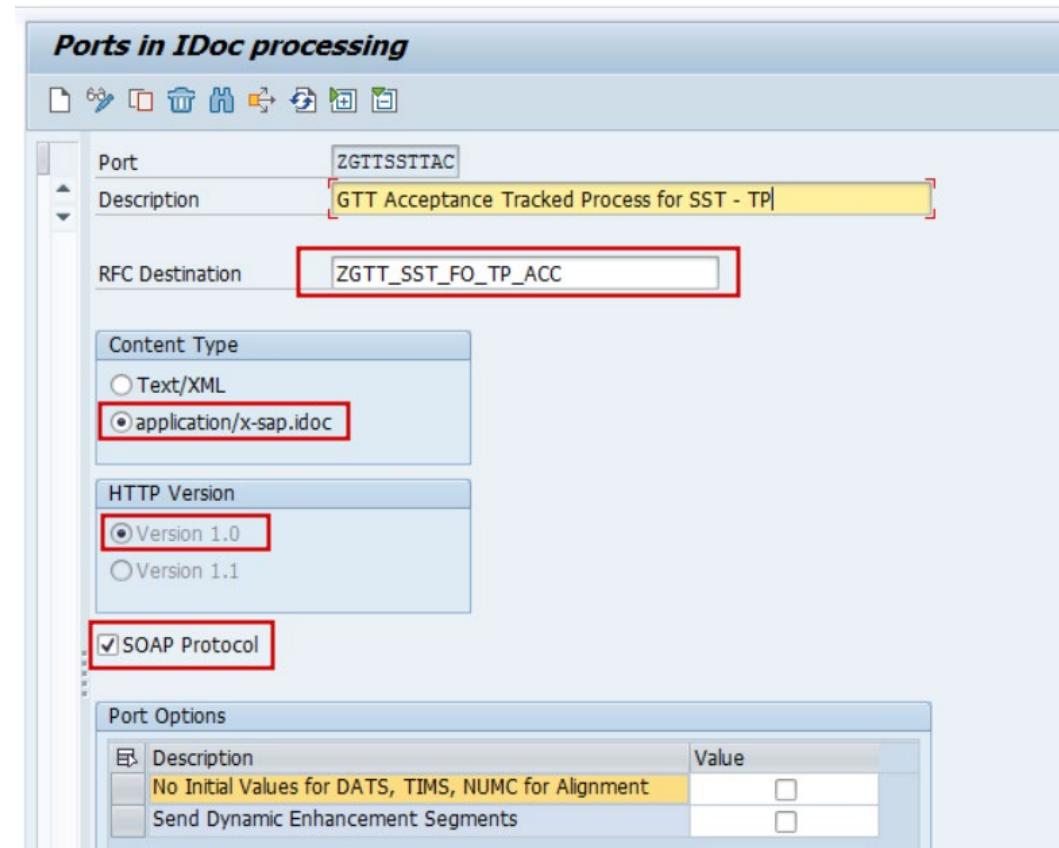
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol 8

3-8: Save the configuration

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

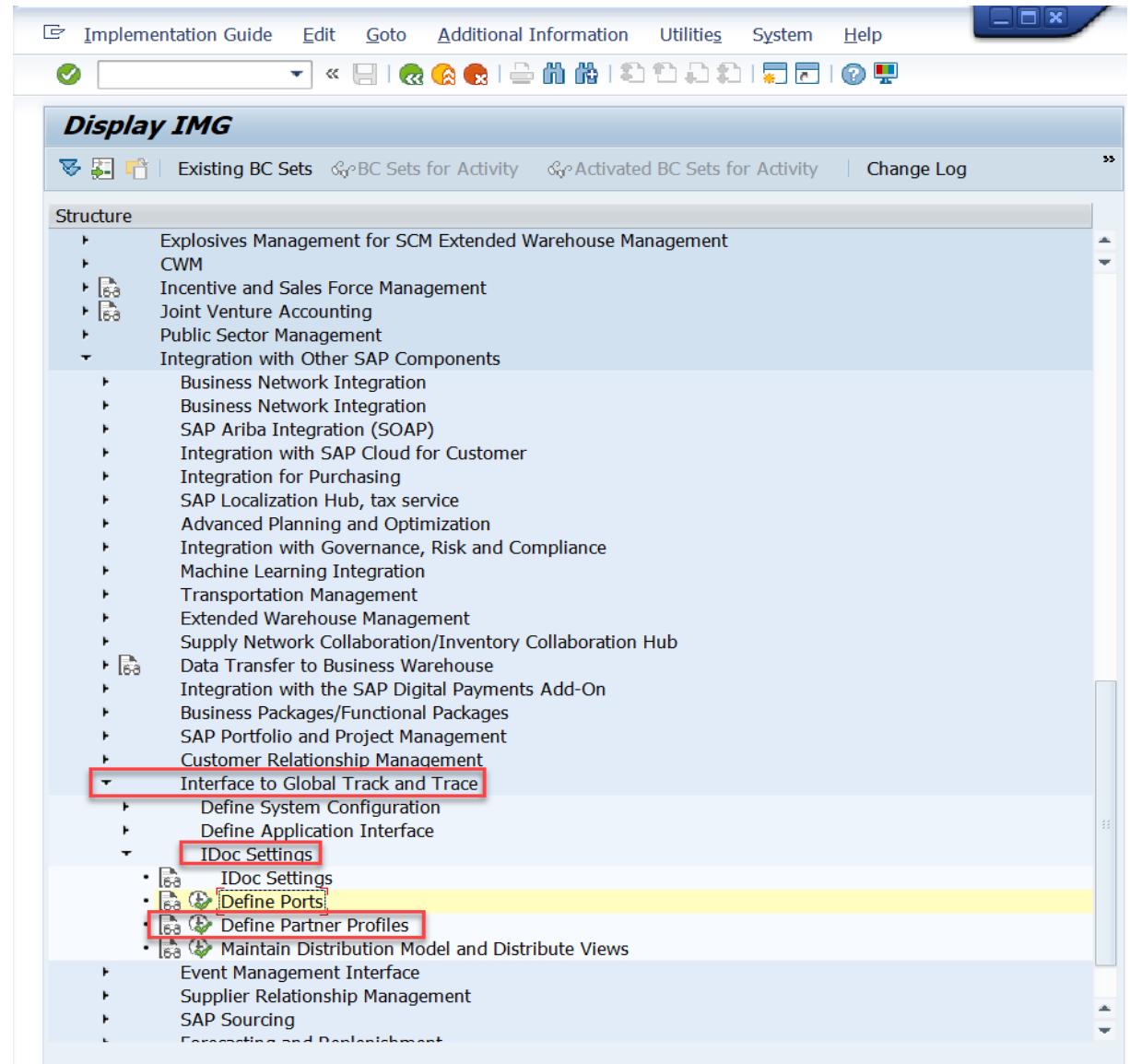


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSTEAC	GTT Acceptance Event for SST	ZGTT_SST_FO_EVENT_ACC	application/x-sap.idoc	Version 1.0	Checked
ZGTTSTTAC	GTT Acceptance Tracked Process for SST	ZGTT_SST_FO_TP_ACC	application/x-sap.idoc	Version 1.0	Checked

# STEP 4: Define Partner Profiles

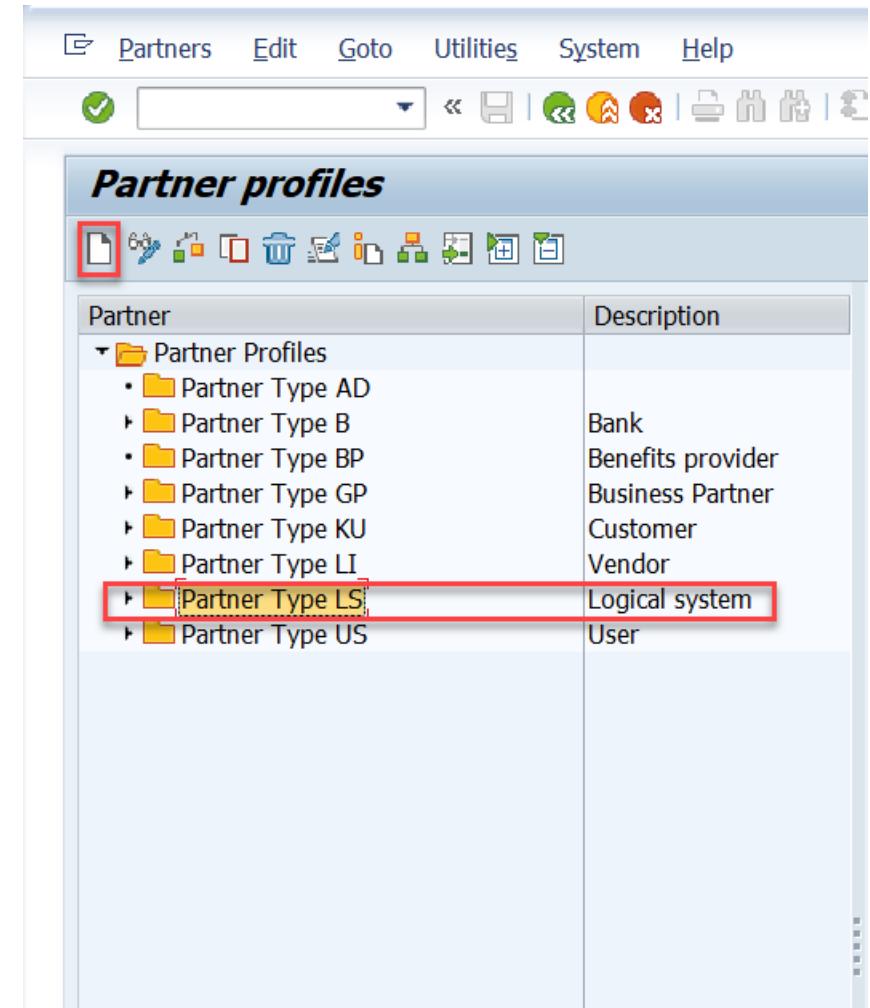
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -  
> **Interface to Global Track and Trace** -  
> **IDoc Settings**

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



## STEP 4: Define Partner Profiles

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



## STEP 4: Define Partner Profiles

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

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

The screenshot shows the SAP Fiori interface for managing Partner profiles. The top section displays basic partner information: Partner No. (ZGTTSSSTAC) and Type (LS). The 'Processor' field is highlighted with a red box, indicating it is the current focus for configuration. Below this, the 'Outbound' section lists message types and their corresponding receiver partners and basic types. The 'Inbound' section is currently empty.

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

## STEP 4: Define Partner Profiles

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

The screenshot shows the SAP Partner profiles interface. At the top, there is a toolbar with various icons. Below the toolbar, the main area is divided into several sections:

- Partner No.**: ZGTTSSITAC (Logical System For GTT SST - Accept)
- Type**: LS (Logical system)
- Post Processing: Valid Processors**: Shows Ty. (US), Processor (blurred), and Lang. (EN) English.
- Classification** and **Telephony** tabs are also visible.
- Outbound** tab: A table showing partner roles and message types.

Partner Role	Message Type	Message Va...	Function	Test	Receiver P...	I... Pa...	Basic Type
	AOPOST			<input type="checkbox"/>	ZGTTSSITAC	0	EHPOST01
	EVMSTA			<input type="checkbox"/>	ZGTTSSSTEAC	0	EVMSTA02
- Inbound** tab: An empty table with columns: Partner Role, Message Type, Message Va..., Function, Test, P.., and Process Code.

At the bottom of the interface, there are several icons: a magnifying glass, a plus sign, a minus sign, and other navigation symbols. The 'Add' icon (plus sign) is highlighted with a red box.

# STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

**Message Type:** EVMSTA

For the tracked Process:

**Message Type:** AOPOST

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

4-9: Save the configuration

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

**Partner profiles: Outbound parameters**

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

Outbound Options    Message Control    Post Processing: Valid Processors    Tele...

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

IDoc Type

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

# STEP 4: Define Partner Profiles

4-10: Fill in the Message Type.

For the Tracked Process:

**Message Type:** AOPOST

4-11: Fill in the Receiver Port, that you created in  
STEP 3

4-12: Save the configuration

Partner No.	Type	Outbound	Message Type	Receiver Port	IDoc Type
ZGTTSSSTAC	LS	Yes	AOPOST	ZGTTSSSTTAC	EHPOST01
ZGTTSSSTAC	LS	Yes	EVMSTA	ZGTTSSSTEAC	EVMSTA02

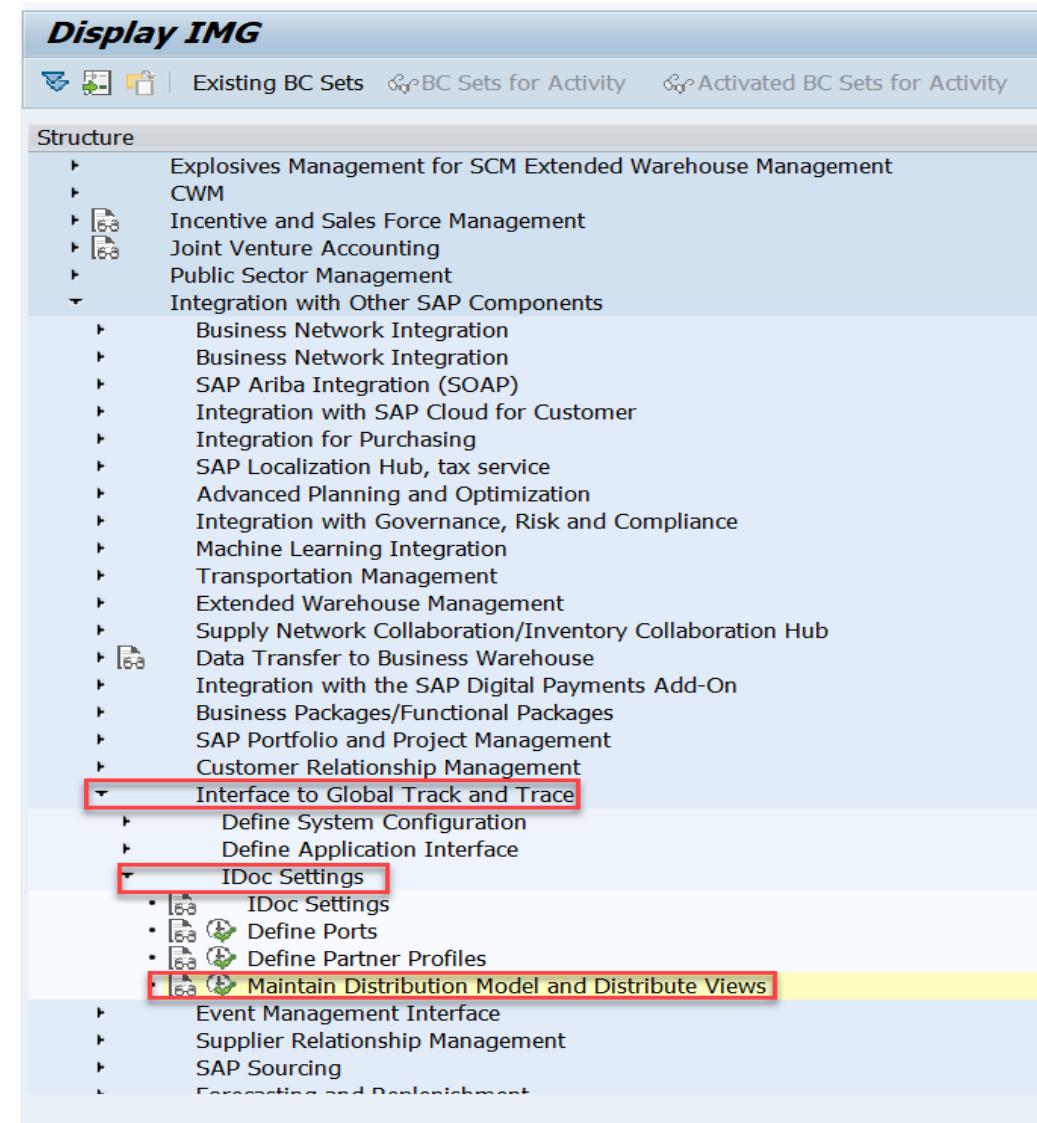
**Partner profiles: Outbound parameters**

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system
Partner Role		
Message Type	AOPOST	
Message Code		
Message Function		<input type="checkbox"/> Test
Outbound Options    Message Control    Post Processing: Valid Processors    Tele...		
Receiver Port	ZGTTSSSTTAC	GTT Acceptance Tracked Proc...
Pack. Size		
<input type="checkbox"/> Queue Processing		
Output Mode		
<input checked="" type="radio"/> Pass IDoc Immediately	Output Mode	2
<input type="radio"/> Collect IDocs		
IDoc Type		
Basic Type	EHPOST01	SCEM: Event Handler Posting
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		Application Release

# STEP 5: Maintain Distribution Model and Distribute Views

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

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



# STEP 5: Maintain Distribution Model and Distribute Views

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

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

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

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

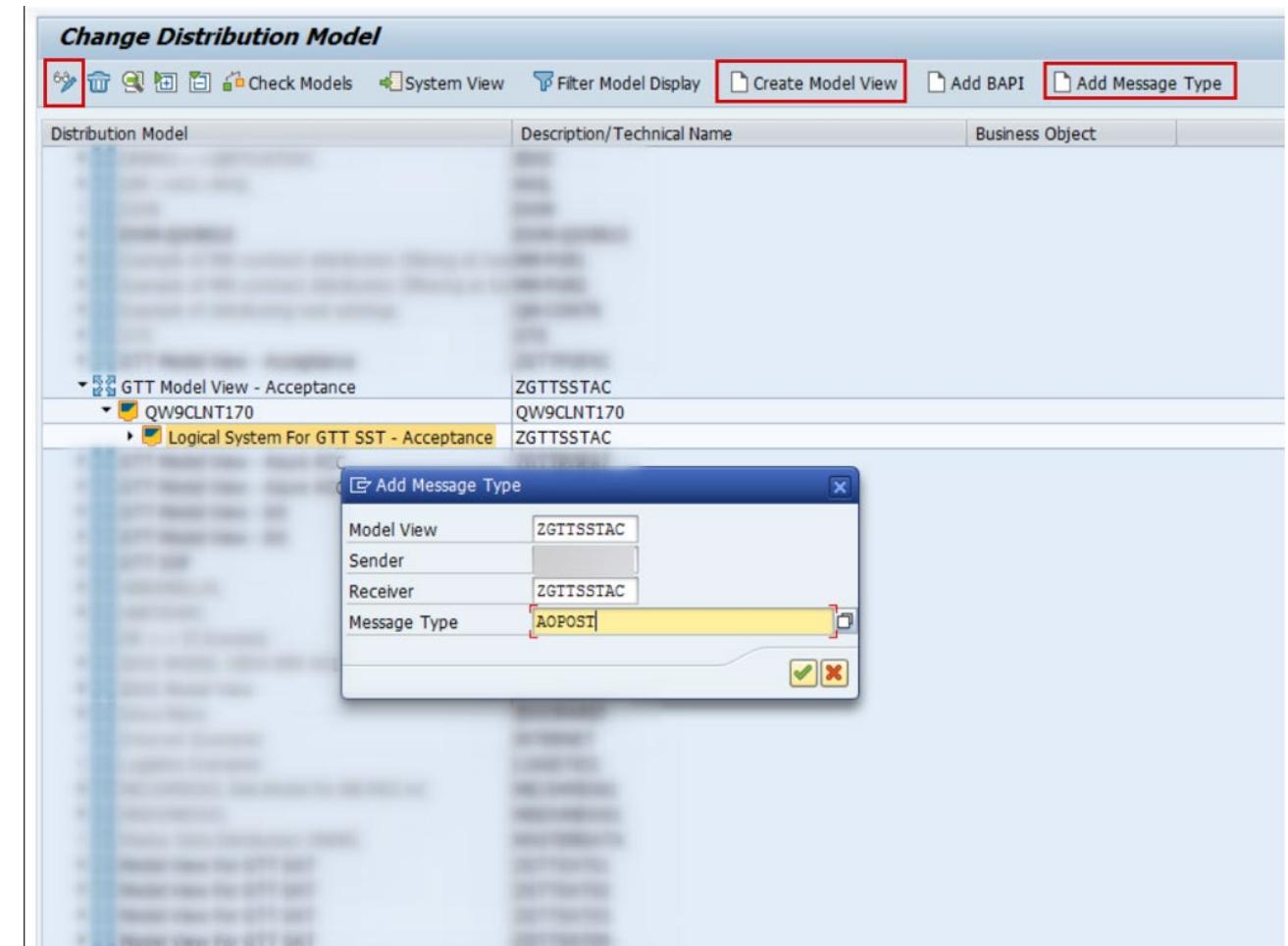
For the event:

**Message Type:** EVMSTA

For the tracked Process:

**Message Type:** AOPOST

5-7: Save the configuration



# B) Configuration and Implementation

## - Basic

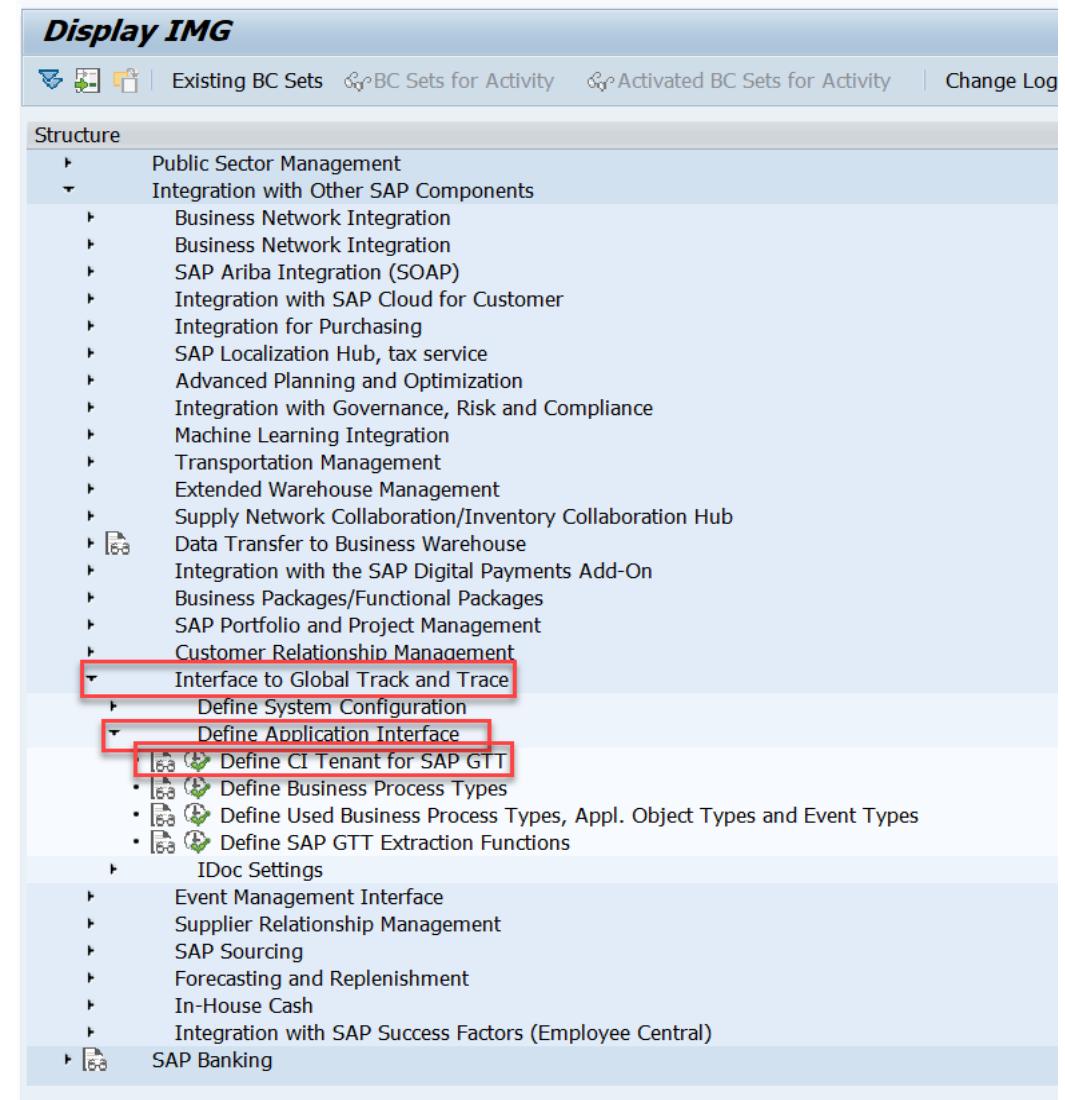
### B2. Extractor Configuration



# STEP 6: Define CI Tenant for SAP Business Network Global Track and Trace

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

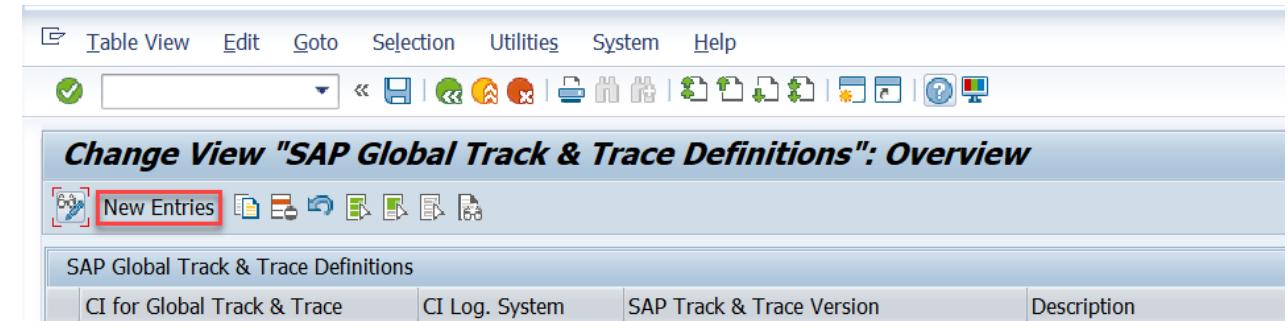
- 6-2: Choose activity  
**Define CI Tenant for SAP GTT**



# STEP 6: Define CI Tenant for SAP Business Network Global Track and Trace

6-3: Click **New Entries** to create a new CI tenant for SAP Business Network Global Track and Trace

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



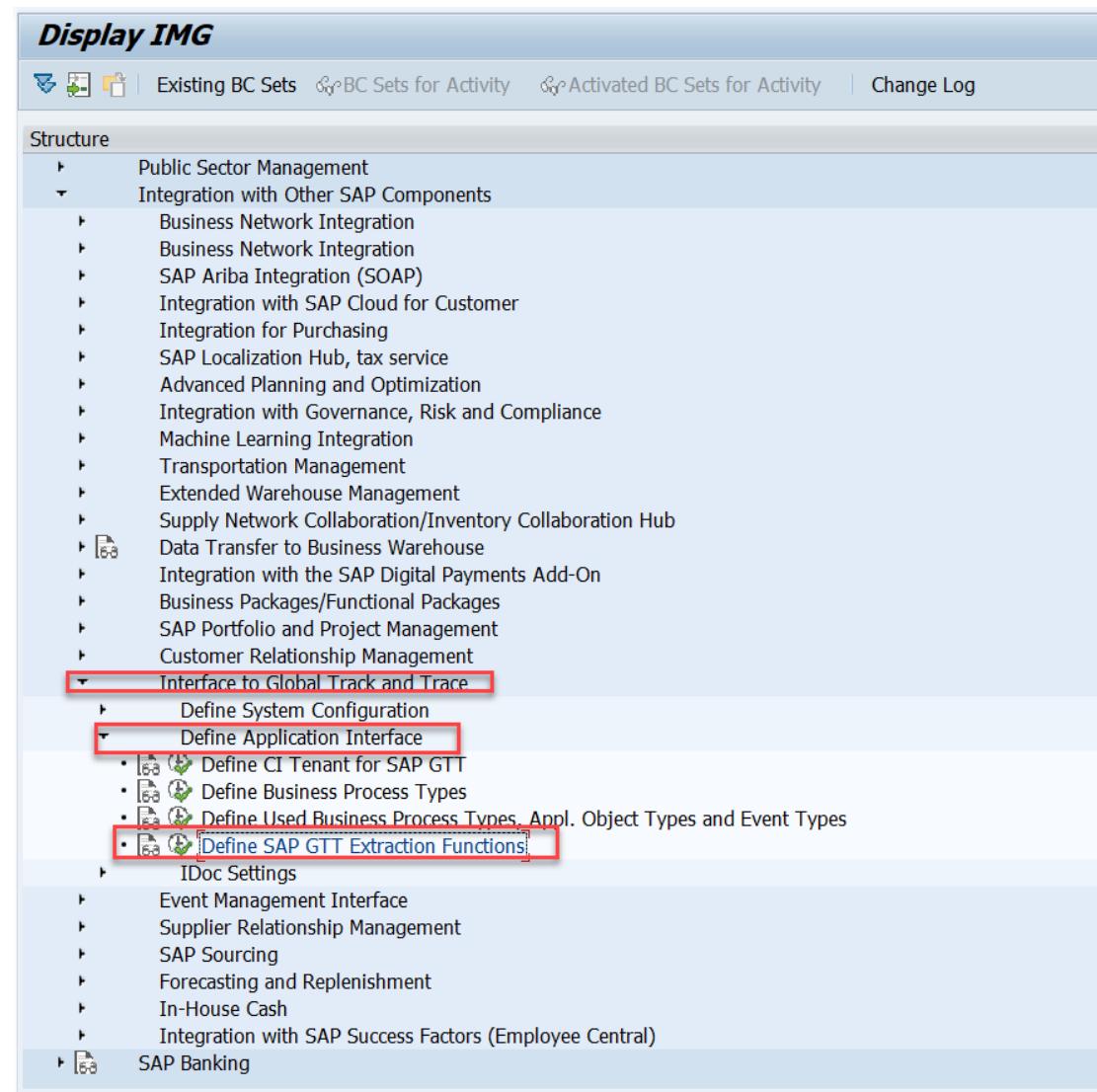
The screenshot shows the SAP Global Track & Trace Definitions overview screen. The table has the following data:

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

# STEP 7: Define GTT Extraction Functions

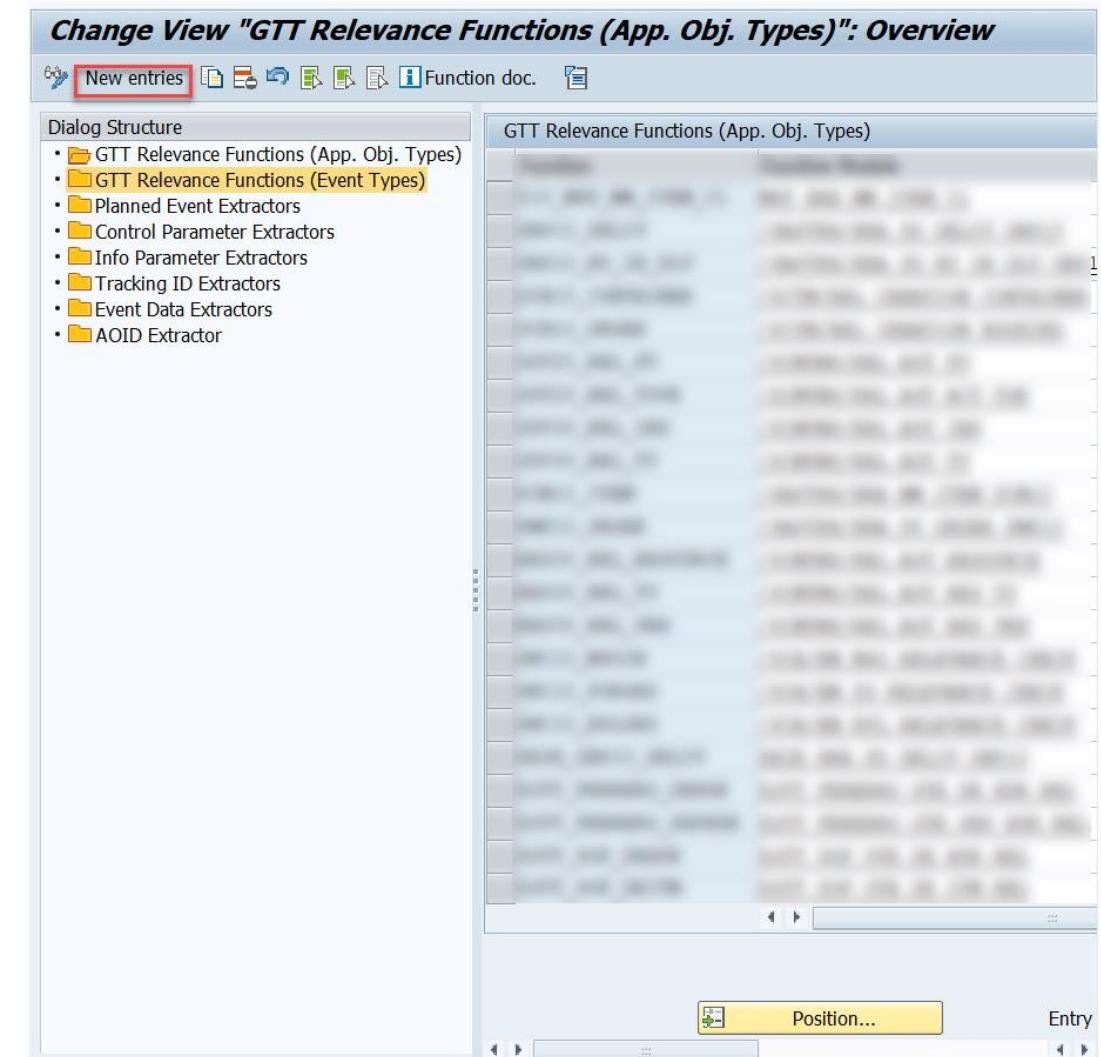
7-1: In **Display IMG** page, click  
**Integration with Other SAP Components -**  
-> **Interface to Global Track and Trace -**  
-> **Define Application Interface**

7-2: Choose activity  
**Define SAP GTT Extraction Functions**



# STEP 7: Define GTT Extraction Functions

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



# STEP 7: Define GTT Extraction Functions

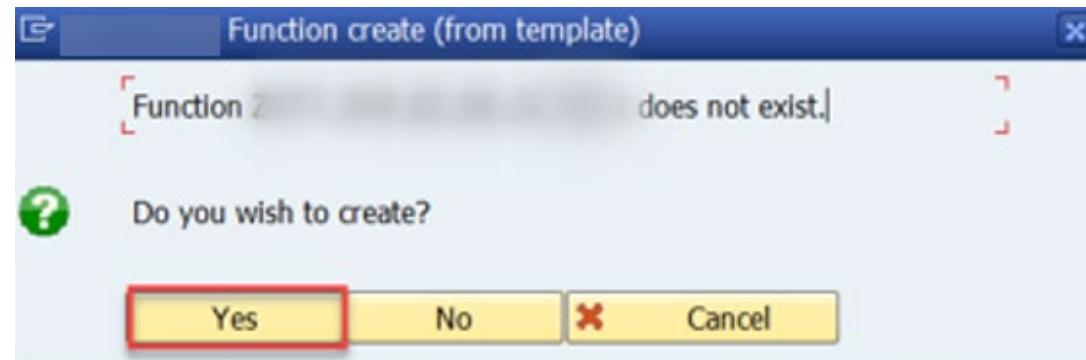
7-4: Input the **Function name** and **Function Module** for the newly created extraction function

7-5: Click **Save**

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

## STEP 7: Define GTT Extraction Functions

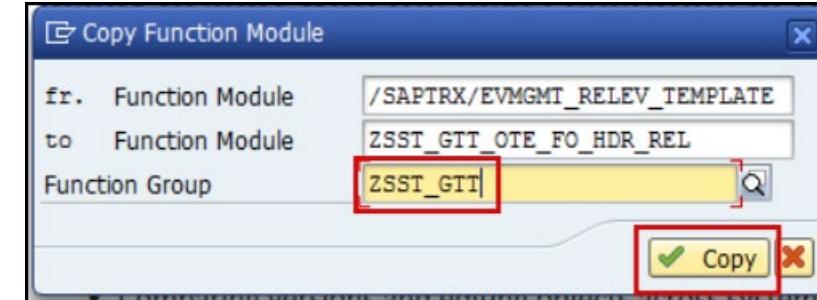
7-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



# STEP 7: Define GTT Extraction Functions

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

7-8: Click **Copy**



# STEP 7: Define GTT Extraction Functions

7-9: Use T-Code SE80 to check the function module you just created

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

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

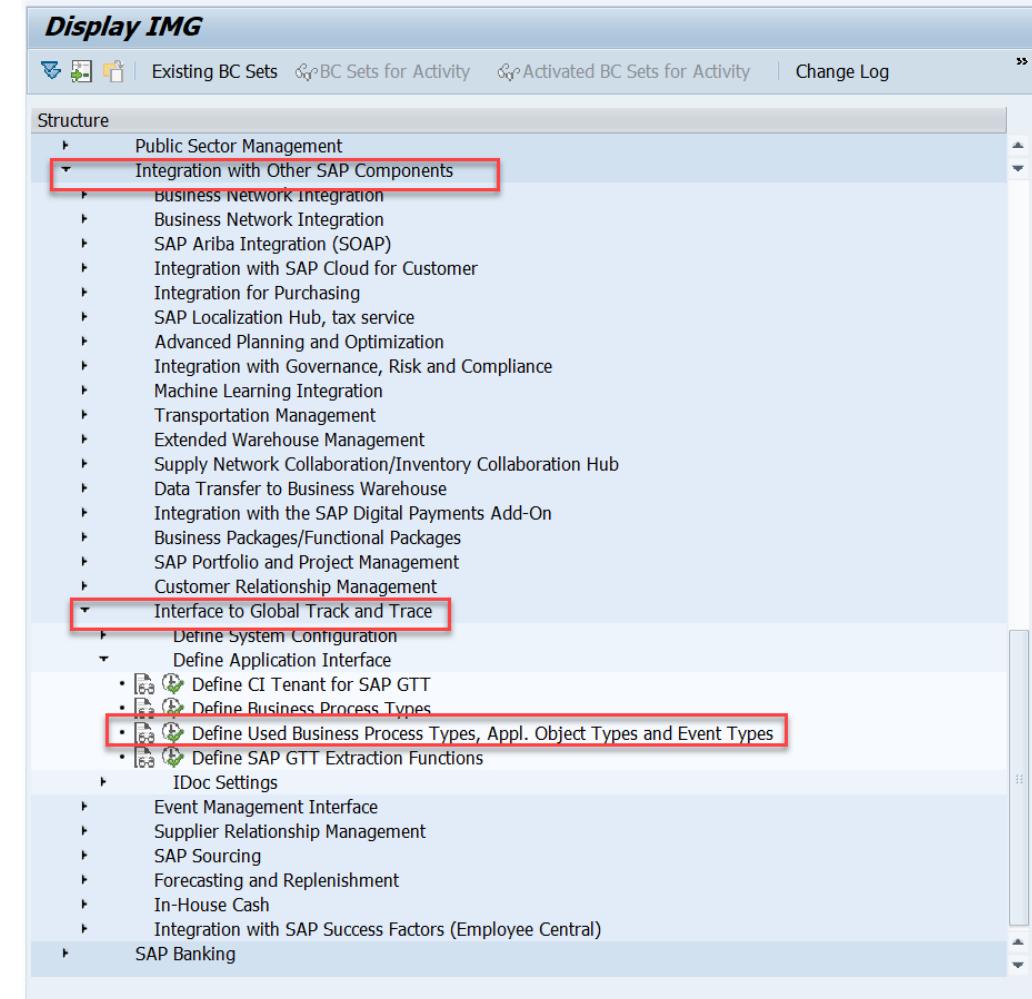
- Title Bar:** Function Builder: Display ZSST\_GTT\_OTE\_FO\_HDR\_REL
- Toolbar:** Includes icons for saving, opening, printing, and navigating.
- Repository Browser:** Shows the function group ZSST\_GTT selected.
- Function Module:** ZSST\_GTT\_OTE\_FO\_HDR\_REL (active)
- Tabs:** Attributes, Import, Export, Changing, Tables, Exceptions, Source Code (selected).
- Source Code:**

```
1 FUNCTION ZSST_GTT_OTE_FO_HDR_REL .
2   *"*
3   *" Local Interface:
4   *"   IMPORTING
5   *"     REFERENCE(l_APPSYS) TYPE /SAPTRX/APPLSYSTEM
6   *"     REFERENCE(l_APP_OBJ_TYPES) TYPE /SAPTRX/ACTYPES
7   *"     REFERENCE(l_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER
8   *"     REFERENCE(l_APPTYPE_TAB) TYPE TRXAS_APPTYPE_TABS_WA
9   *"     REFERENCE(l_APP_OBJECT) TYPE TRXAS_APPOBJ_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  DATA: lt_app_objects TYPE trxas_appobj_ctabs,
20        lo_udm_message TYPE REF TO cx_udm_message,
21        ls_bapiret TYPE bapiret2.
22
23  lt_app_objects = VALUE #( ( i_app_object ) ).
24
25  TRY.
26    e_result = lcl_ef_performer->check_relevance(
27      is_definition = VALUE #( maintab = lif_sst_constants->cs_tabledef-fo_header_new )
28      io_bo_factory = NEW lcl_tor_factory( )
29      iv_abosvs = i_abosvs
30
31  Scope: \FUNCTION ZSST_GTT_OTE_FO_HDR_REL ABAP | Ln 9 Col 10
```

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

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

- 8-2: Choose activity **Define Used Business Process Types, Appl. Object Types and Event Types**



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

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

In the following:

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

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

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

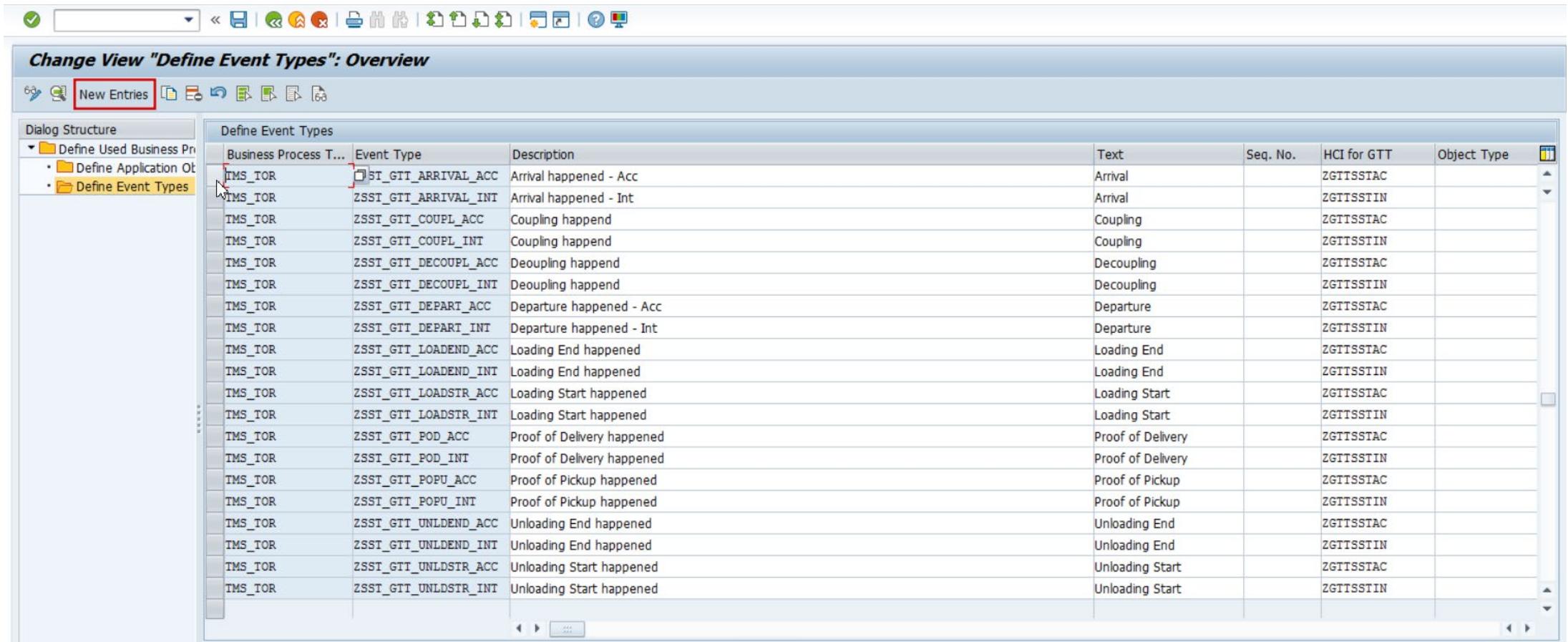
8-3: Choose the business process type from the **Define Used Business Process Types** on the right side

8-4: Double click **Define Event Types**

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

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

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



The screenshot shows the SAP interface for defining event types. The title bar reads "Change View 'Define Event Types': Overview". The toolbar includes standard SAP icons for saving, canceling, and navigating. A red box highlights the "New Entries" button in the toolbar. On the left, a "Dialog Structure" tree shows "Define Used Business Pro" expanded, with "Define Application Obj" and "Define Event Types" selected. The main area is titled "Define Event Types" and contains a table with the following data:

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

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

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

8-7: Fill in the information required in

the **General Data** tab.

**HCI for GTT** is the CI Tenant you created in STEP 6.

**Event Function** is the extractor function you created in STEP 7.

8-8: Check **GTT Relevant**

Bus. Proc. Type	TMS_TOR
Event Type	ZSST_GTT_ARRIVAL_ACC
Text	Arrival

General Data    Control Tables    Global Track & Trace Relevance

Sequencing / Destination

Seq. No.

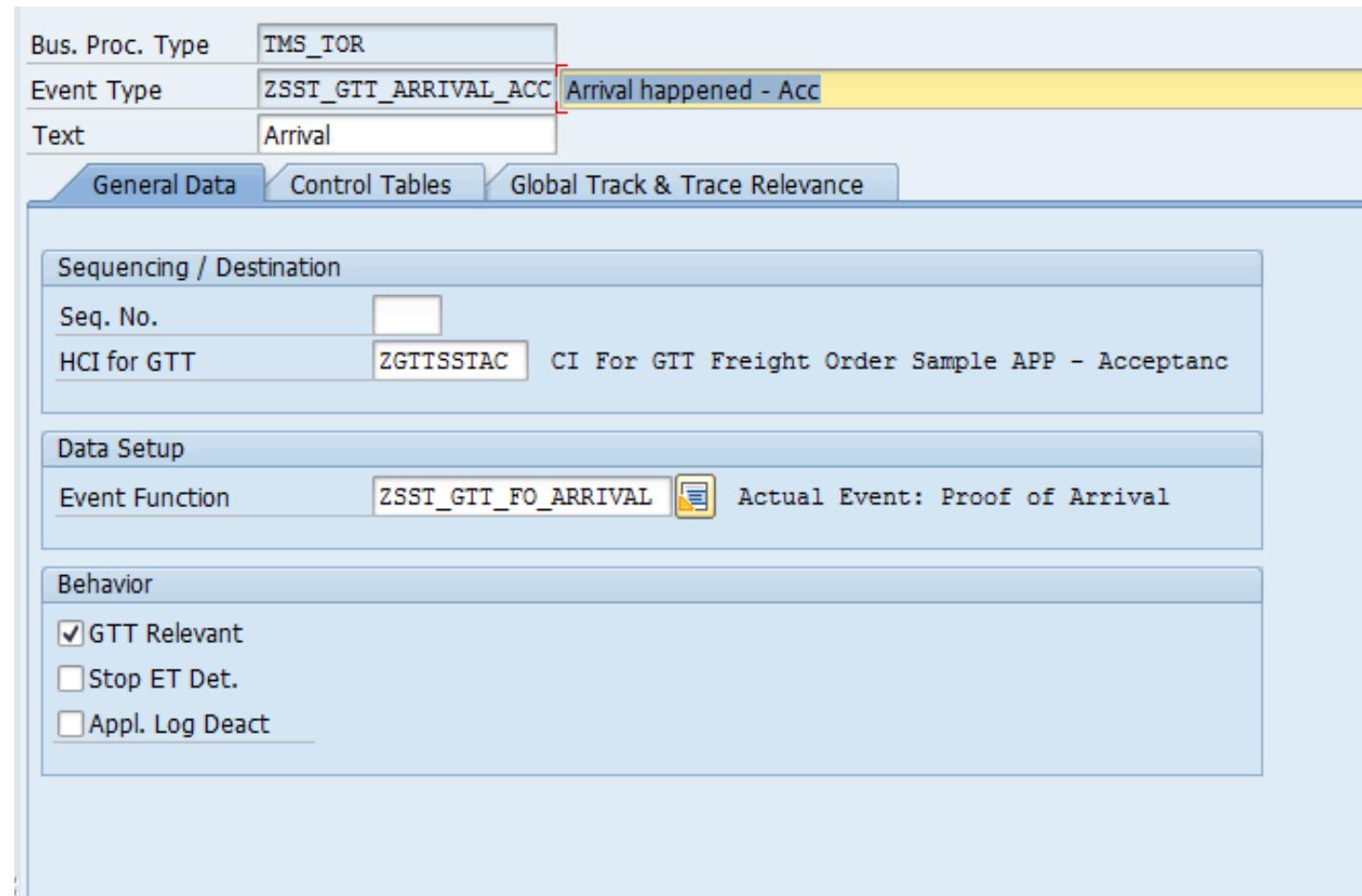
HCI for GTT ZGTTSSSTAC CI For GTT Freight Order Sample APP - Acceptanc

Data Setup

Event Function ZSST\_GTT\_FO\_ARRIVAL Actual Event: Proof of Arrival

Behavior

GTT Relevant  
 Stop ET Det.  
 Appl. Log Deact

A screenshot of the SAP Fiori interface for defining business process types. The top section shows the 'General Data' tab selected, displaying fields for Bus. Proc. Type (TMS\_TOR), Event Type (ZSST\_GTT\_ARRIVAL\_ACC), and Text (Arrival). The Event Type field has a tooltip 'Arrival happened - Acc'. Below this are three tabs: General Data (selected), Control Tables, and Global Track & Trace Relevance. The General Data section contains fields for Sequencing / Destination (Seq. No., HCI for GTT), Data Setup (Event Function, Actual Event), and Behavior (checkboxes for GTT Relevant, Stop ET Det., and Appl. Log Deact). The Behavior section has three checkboxes: 'GTT Relevant' (checked), 'Stop ET Det.', and 'Appl. Log Deact'.

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

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

### Caution:

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

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

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

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

8-10: In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need

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

Click **Save**.

Bus. Proc. Type	TMS_TOR
Event Type	ZSST_GTT_ARRIVAL_ACC Arrival happened - Acc
Text	Arrival

General Data    Control Tables    Global Track & Trace Relevance

GTT Rel. Method	Check Function (Func...)
GTT Rel. Function	ZSST_GTT_FO_ARR_REL Actual Event Relevance: Arr

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

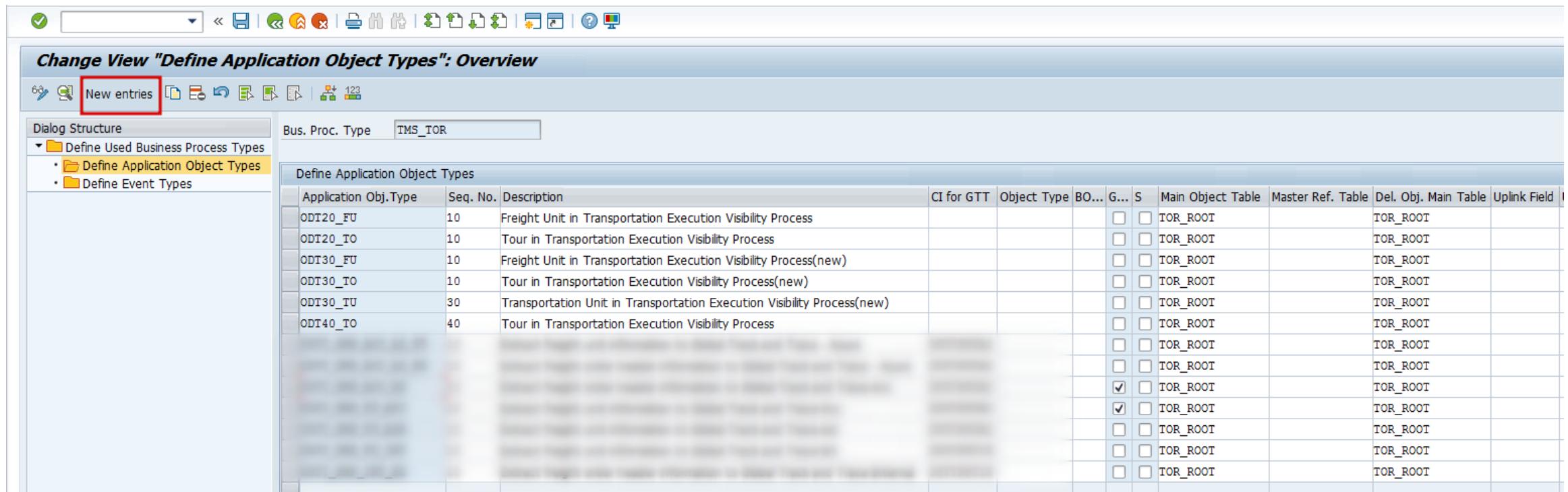
8-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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

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



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

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

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

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

8-15: Fill in the information required in the **General Data** tab.

**HCI for GTT** is the CI Tenant you created in STEP 6.

**Event Function** is the extractor function you created in STEP 7.

8-16: Check **GTT Relevant**

The screenshot shows the SAP Fiori interface for defining business process types. The top section displays the 'General Data' tab, which includes fields for 'Bus. Proc. Type' (set to 'TMS\_TOR'), 'Appl. Obj. Type' (set to 'ZGTT\_SHP\_ACC\_HD'), and 'Text'. A tooltip for the application object type is visible, stating 'Extract freight order header information to Global Track and Trace-Acc'. Below the tabs, there are three main configuration sections: 'Sequencing / Destination', 'Business Object Reference', and 'Behavior'. The 'Behavior' section contains checkboxes for 'GTT Relevant' (which is checked and highlighted with a red border), 'Stop AO Determ.', and 'Appl. Log Deact'. The 'Alt. BusProcType' field is also present in this section.

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

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

## Caution:

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

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

The screenshot shows a configuration screen for a business process type. At the top, there are three input fields: 'Bus. Proc. Type' (TMS\_TOR), 'Appl. Obj. Type' (ZGTT\_SHP\_ACC\_HD), and 'Text'. A tooltip for the application object type says 'Extract freight order header information to Global Track and Trace-Acc'. Below these are five tabs: General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. Under 'Object Identification', there are two sections: 'Data Source for Created and Updated Objects' (Main Obj. Table: TOR\_ROOT) and 'Data Source for Deleted Objects' (Del.Obj. Table: TOR\_ROOT). There is also a section for 'Reference Between Main and Master Table'.

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

8-18: In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need.

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

Click **Save**.

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

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

GTT Rel. Method   Check Function (Function Module)   ▾

GTT Rel. Function   ZSST\_GTT\_FO\_HDR    Appl. Object Type Relevance for Freight Order Header

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

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

Click **Save**.

The screenshot shows the SAP Fiori interface for defining business process types and object identification. The top navigation bar includes tabs for General Data, Control Tables, Object Identification (which is selected), Global Track & Trace Relevance, and Parameter Setup. The main content area is divided into several sections:

- Bus. Proc. Type:** TMS\_TOR
- Appl. Obj. Type:** ZGTT\_SHP\_ACC\_HD (highlighted with a red box)
- Text:** Extract freight order header information to Global Track and Trace-Acc (highlighted with a yellow box)
- Method for determination of AOID:** AOID Method dropdown set to "Determine from Field".
- Application Object ID Source:**
  - First Field to Build Appl. Obj. ID: Cntrl Tab. Type dropdown set to "Main Object Table", AO ID Field input field set to "TOR\_ID".
  - Second Field to Build Appl. Obj. ID: Cntrl Tab. Type dropdown, AO ID Field input field.
- Determine AOID By Function:** AOID Function input field.

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

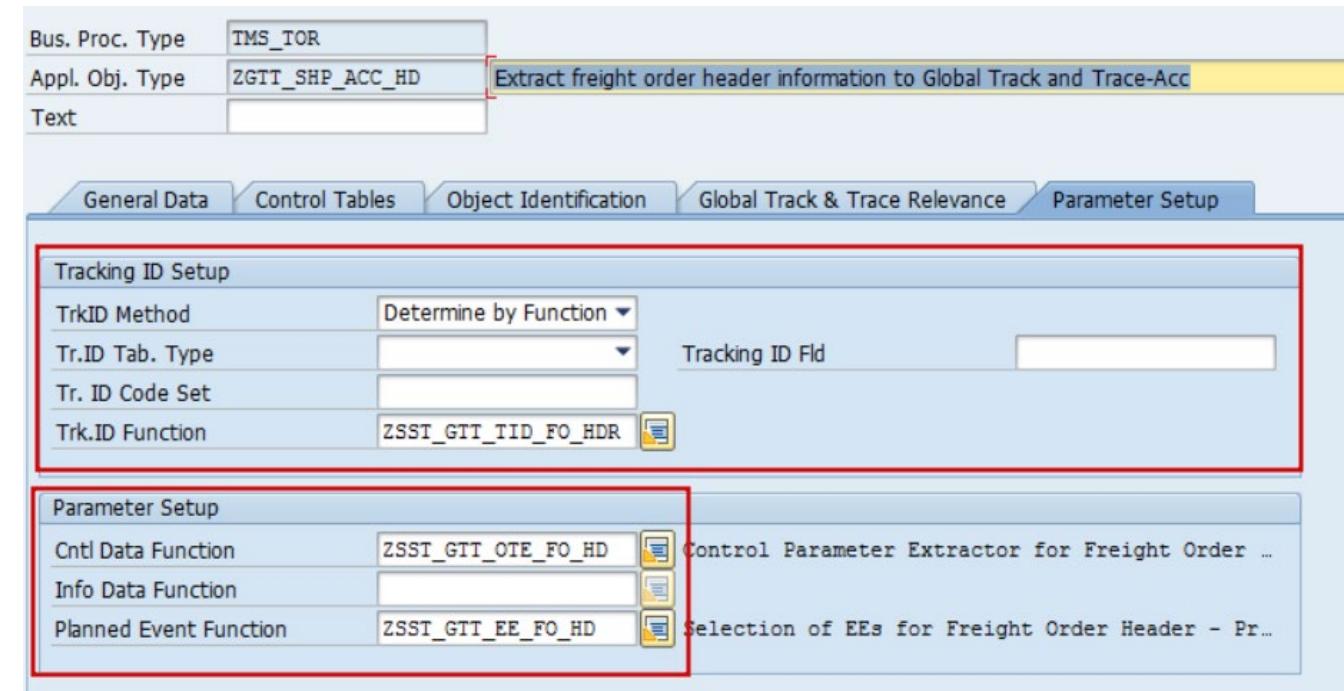
8-20: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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

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

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

Click **Save**.



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

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

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

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
<a href="#">General Data</a> <a href="#">Control Tables</a> <a href="#">Object Identification</a> <a href="#">Global Track &amp; Trace Relevance</a> <a href="#">Parameters</a>	
<b>Data Source for Created and Updated Objects</b>	
Main Obj. Table	TOR_ROOT
Master Table	
<b>Data Source for Deleted Objects</b>	
Del.Obj. Table	TOR_ROOT
<b>Reference Between Main and Master Table</b>	
First Field Reference from Main to Master Table	
Second Field Reference from Main to Master Table	

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

The image displays two side-by-side screenshots of the SAP Fiori interface, likely from the SAP S/4HANA or SAP ERP system, illustrating the configuration of Business Process Types (BPT) and Application Object Types (AOT).

**Screenshot 1 (Left): General Data Configuration**

- General Data:**
  - Bus. Proc. Type: TMS\_TOR
  - Appl. Obj. Type: ZGTT\_SHP\_FU\_ACC
  - Text: Extract freight unit information to Global Track and Trace-Acc
- Control Tables:**
  - Method for determination of AOID
  - AOID Method: Determine from Field
- Object Identification:**
  - First Field to Build Appl. Obj. ID: Cntrl Tab. Type: Main Object Table; AO ID Field: TOR\_ID
  - Second Field to Build Appl. Obj. ID: Cntrl Tab. Type: (empty); AO ID Field: (empty)
- Global Track & Trace Relevance:**
  - Determine AOID By Function: AOID Function: (empty)

**Screenshot 2 (Right): Parameter Setup Configuration**

- General Data:**
  - Bus. Proc. Type: TMS\_TOR
  - Appl. Obj. Type: ZGTT\_SHP\_FU\_ACC
  - Text: Extract freight unit information to Global Track and Trace-Acc
- Control Tables:**
  - Tracking ID Setup:
    - TrkID Method: Determine by Function
    - Tr.ID Tab. Type: (empty)
    - Tr. ID Code Set: (empty)
    - Trk.ID Function: ZSST\_GTT\_TID\_FO\_HDR (with a help icon)
  - Parameter Setup:
    - Cntl Data Function: ZSST\_GTT\_OTE\_FO\_HD (with a help icon)
    - Info Data Function: (empty)
    - Planned Event Function: ZSST\_GTT\_EE\_FO\_HD (with a help icon)
- Object Identification:**
  - Bus. Proc. Type: TMS\_TOR
  - Appl. Obj. Type: ZGTT\_SHP\_FU\_ACC
  - Text: Extract freight unit information to Global Track and Trace-Acc
- Global Track & Trace Relevance:**
  - GTT Rel. Method: Check Function (Function Module)
  - GTT Rel. Function: ZSST\_GTT\_FO\_HDR (with a help icon)
- Parameter Setup:**
  - Bus. Proc. Type: TMS\_TOR
  - Appl. Obj. Type: ZGTT\_SHP\_FU\_ACC
  - Text: Extract freight unit information to Global Track and Trace-Acc

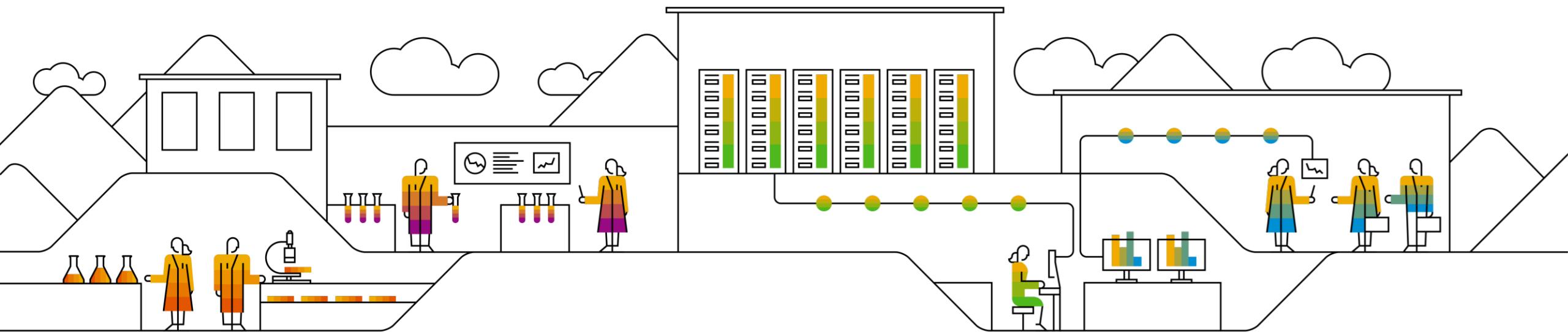
# C) Download ABAP Code from GitHub

C1. Initial Download ABAP code from GitHub(Only for TSO)

C2. Update ABAP code from GitHub(Only for TSO)

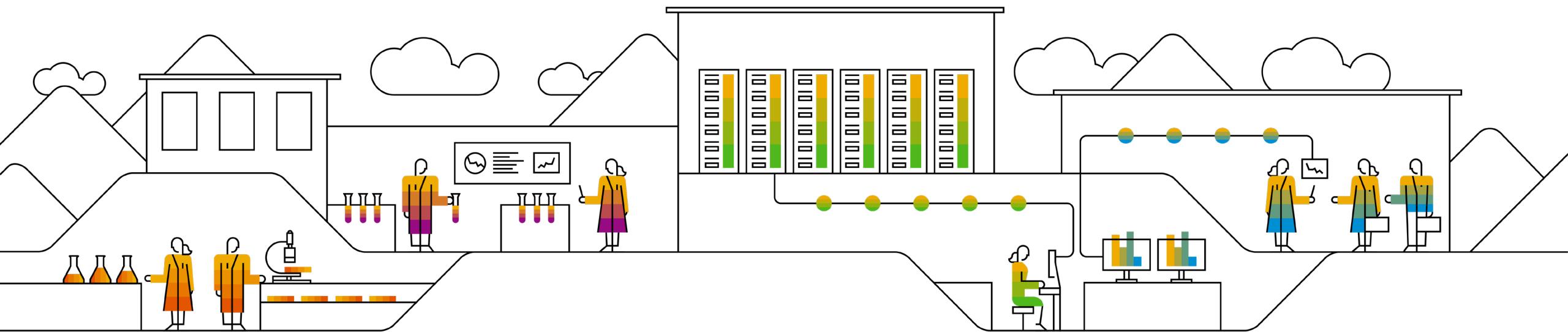
C3. Download Another ABAP code from GitHub(TPO)

C4. Initial Download ABAP code from GitHub(Include TSO/TPO/TS)



# C) Download ABAP Code from GitHub

## C1. Initial Download ABAP code from GitHub(Only for TSO)



# STEP 1: Install ABAPGit

You need to install ABAPGit before downloading codes from GitHub.

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

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

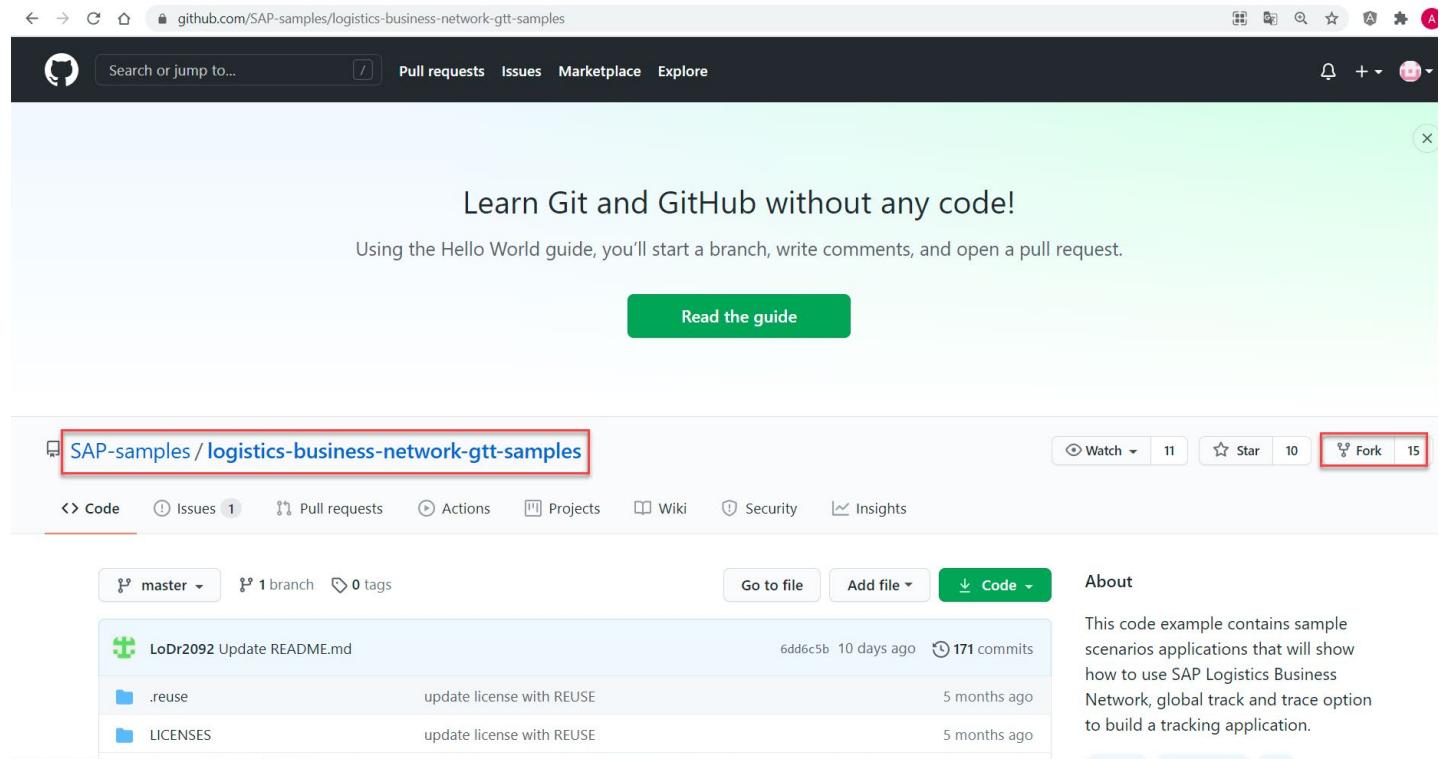
When installation is complete, a new report is created, **ZABAPGIT\_STANDALONE**.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar has sections like "Getting Started" (with links to Installation, Upgrading, Uninstalling, and UI features), "Setup" (with links to SSL setup, Proxy configuration, and Development version), "Online Projects" (with links to Installing online repo, Keeping code up to date, Uninstall repository, First project, Moving package into git, and Contributing to a project), "Offline Projects" (with links to Import zip and Export zip), and "Reference" (with links to Repo Settings (abapgit.xml), Supported object types, Icon Legend, User Exits, Authorizations, and Namespaces). The main content area starts with a "Summary" section stating that abapGit exists in two flavours: standalone or developer version. It then details the differences between the two versions. A "Prerequisites" section notes that SAP BASIS version 702 or higher is required. The "Install standalone version" section is highlighted with a red border and contains numbered steps: 1. Download the ABAP code (right click -> save-as) to a file, 2. Via SE38 or SE80, create a new report named ZABAPGIT\_STANDALONE (formerly ZABAPGIT\_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version, 3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload, 4. Activate. Below these steps, it says "Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. \$ZABAPGIT)". Finally, it states "Now you can use abapGit by executing the report in transaction SE38".

# STEP 2: Fork Sample code Repository

2-1.Navigate to sample code in  
<https://github.com/SAP-samples/logistics-business-network-gtt-samples>

2-2.Click the “Fork” button, it will copy the newest version of sample codes into the user’s account and meanwhile it will navigate to user’s own repository



# STEP 3: Change Configuration file ‘.abapgit.xml’

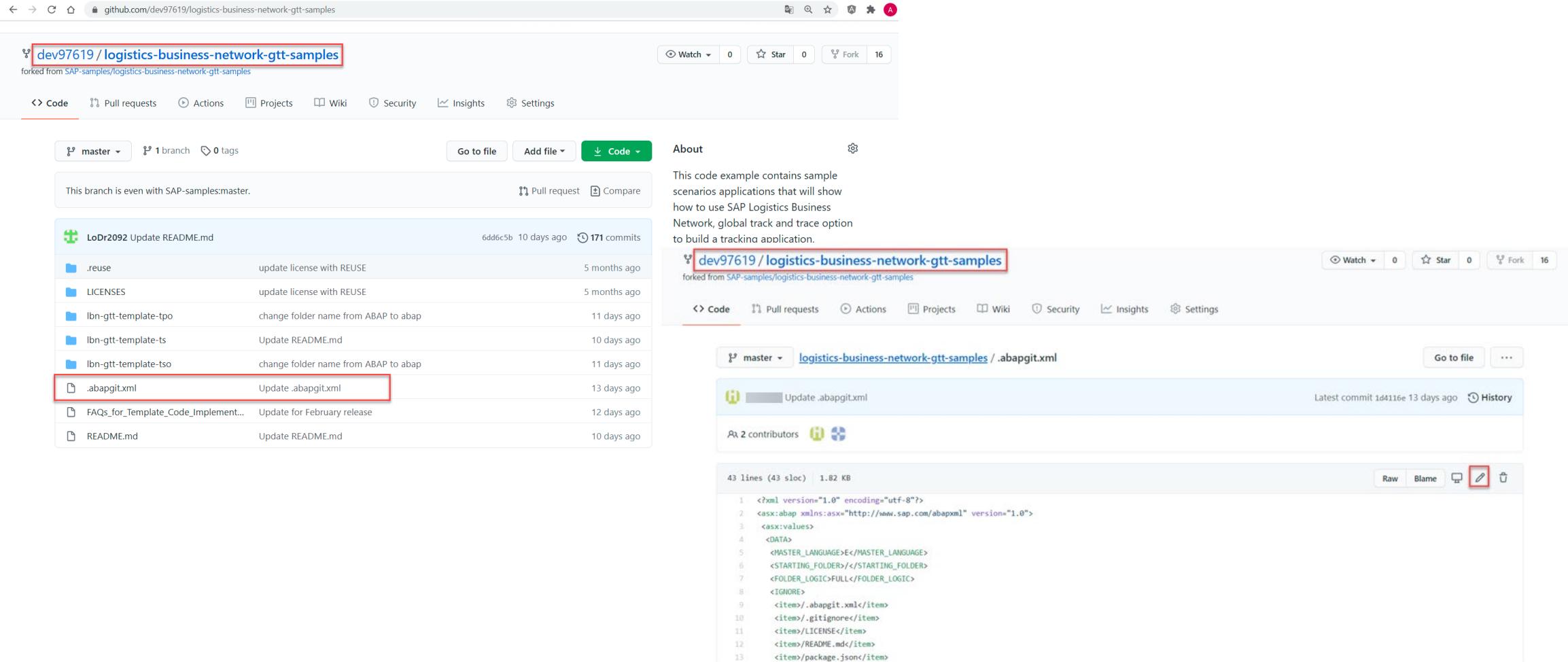
3-1: In the user’s account repository, click the file ‘.abapgit.xml’

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The main content area displays a list of commits. A commit by 'LoDr2092' titled 'Update README.md' is at the top. Below it, a commit by 'LoDr2092' titled 'Update .abapgit.xml' is highlighted with a red box. Other commits listed include updates to '.reuse', 'LICENSES', folder names, and README files.

File / Commit	Description	Date
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
<b>.abapgit.xml</b>	<b>Update .abapgit.xml</b>	13 days ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

# STEP 3: Change Configuration file '.abapgit.xml'

3-2: Click  button to edit the file



The screenshot shows two views of a GitHub repository. The top view is the repository page for 'dev97619 / logistics-business-network-gtt-samples'. The bottom view is the code editor for the '.abapgit.xml' file.

**Repository Page (Top):**

- URL: [github.com/dev97619/logistics-business-network-gtt-samples](https://github.com/dev97619/logistics-business-network-gtt-samples)
- Owner: dev97619
- Name: logistics-business-network-gtt-samples
- Description: This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.
- Branch: master
- Pull requests: 0
- Actions: 0
- Projects: 0
- Wiki: 0
- Security: 0
- Insights: 0
- Settings: 0
- Forked from SAP-samples/logistics-business-network-gtt-samples
- Last commit: 6dd6c5b 10 days ago
- Commits: 171
- Files: 1 branch, 0 tags
- Code: Go to file, Add file, Code dropdown

**Code Editor (Bottom):**

- Repository: dev97619 / logistics-business-network-gtt-samples
- File: .abapgit.xml
- Commit: Update .abapgit.xml
- Contributors: 2
- Latest commit: 1d4116e 13 days ago
- History: History link
- Code View:

```
43 lines (43 sloc) 1.82 KB
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER></STARTING_FOLDER>
7       <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8       <IGNORE>
9         <item>/.abapgit.xml</item>
10        <item>/.gitignore</item>
11        <item>/LICENSE</item>
12        <item>/README.md</item>
13        <item>/package.json</item>
```
- Actions: Raw, Blame, Copy, Edit (highlighted), Delete

## STEP 3: Change Configuration file '.abapgit.xml'

3-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with  
"<STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>" as follows.

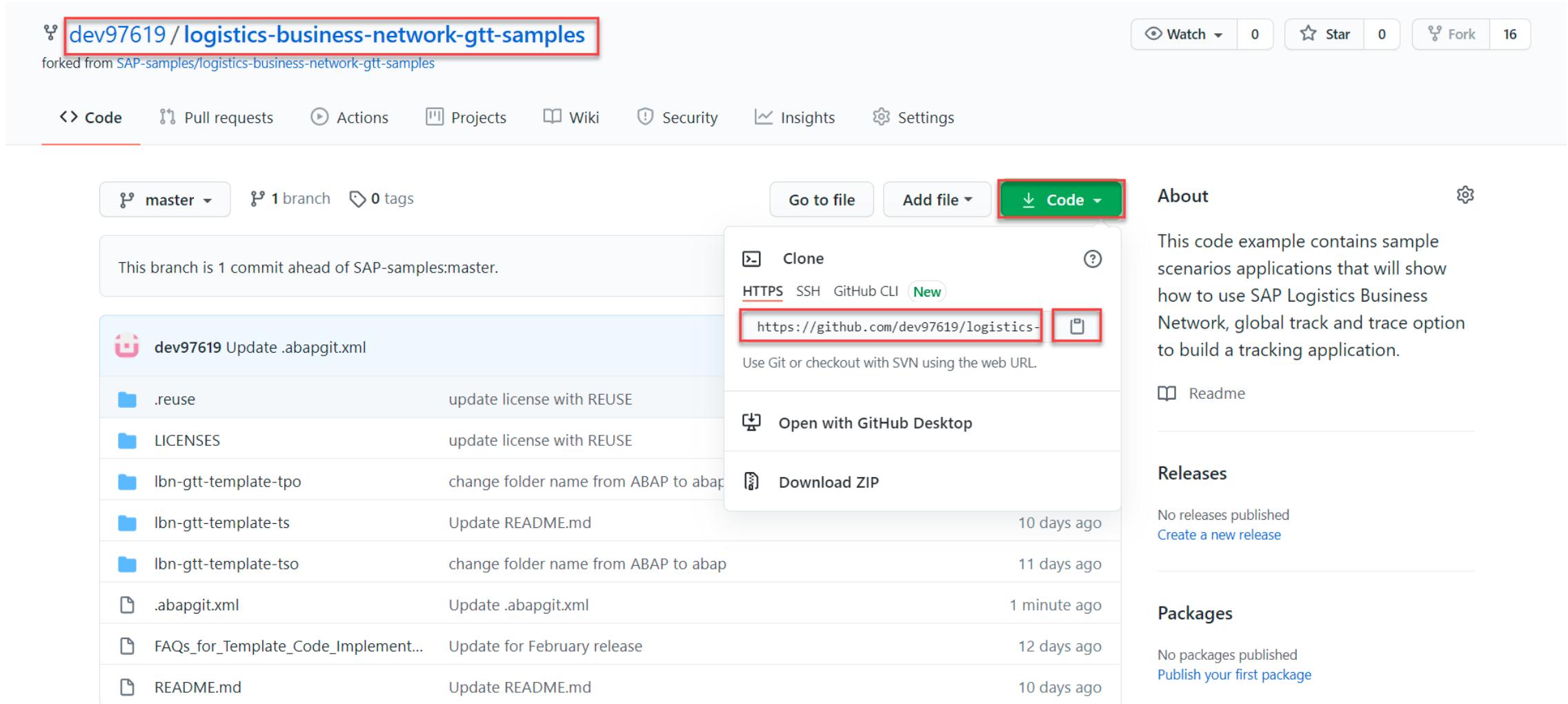
3-4: Commit change

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The '.abapgit.xml' file is open in the editor, showing the configuration code. A specific line has been highlighted with a red box: '<STARTING\_FOLDER>/</STARTING\_FOLDER>'. This line is being replaced by another line, also highlighted with a red box: '<STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>'. To the right of the editor, a 'Commit changes' dialog is displayed. It contains fields for 'Update .abapgit.xml' and an optional extended description. Below these fields are two radio button options: one selected for 'Commit directly to the master branch' and one for 'Create a new branch for this commit and start a pull request'. At the bottom of the dialog are 'Commit changes' and 'Cancel' buttons, with 'Commit changes' also highlighted with a red box.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3 <asx:values>
4 <DATA>
5 <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>
7 <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8 <IGNORE>
9 <item>/.abapgit.xml</item>
10 <item>/.gitignore</item>
```

# STEP 3: Change Configuration file '.abapgit.xml'

3-5: Go to the root and copy the repository URL by clicking  button



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository has been forked from SAP-samples/logistics-business-network-gtt-samples. The 'Code' tab is selected. A dropdown menu is open over the 'Clone' section, with the URL 'https://github.com/dev97619/logistics...' highlighted and a copy icon overlaid.

**Repository Information:**

- Owner: dev97619
- Name: logistics-business-network-gtt-samples
- Forked from: SAP-samples/logistics-business-network-gtt-samples
- Watchers: 0
- Stars: 0
- Forks: 16

**Branches:**

- master (selected)
- 1 branch
- 0 tags

**Commits:**

- dev97619 Update .abapgit.xml
- .reuse update license with REUSE
- LICENSES update license with REUSE
- Ibn-gtt-template-tpo change folder name from ABAP to abap
- Ibn-gtt-template-ts Update README.md
- Ibn-gtt-template-tso change folder name from ABAP to abap
- .abapgit.xml Update .abapgit.xml
- FAQs\_for\_Template\_Code\_Implement... Update for February release
- README.md Update README.md

**Clone Options:**

- Clone (dropdown menu open):
  - HTTPS (selected)
  - SSH
  - GitHub CLI (New)
- URL: [https://github.com/dev97619/logistics...](https://github.com/dev97619/logistics-business-network-gtt-samples) (highlighted with a red box and a copy icon)
- Use Git or checkout with SVN using the web URL.
- Open with GitHub Desktop
- Download ZIP

**About:**

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

**Readme:**

**Releases:**

No releases published  
[Create a new release](#)

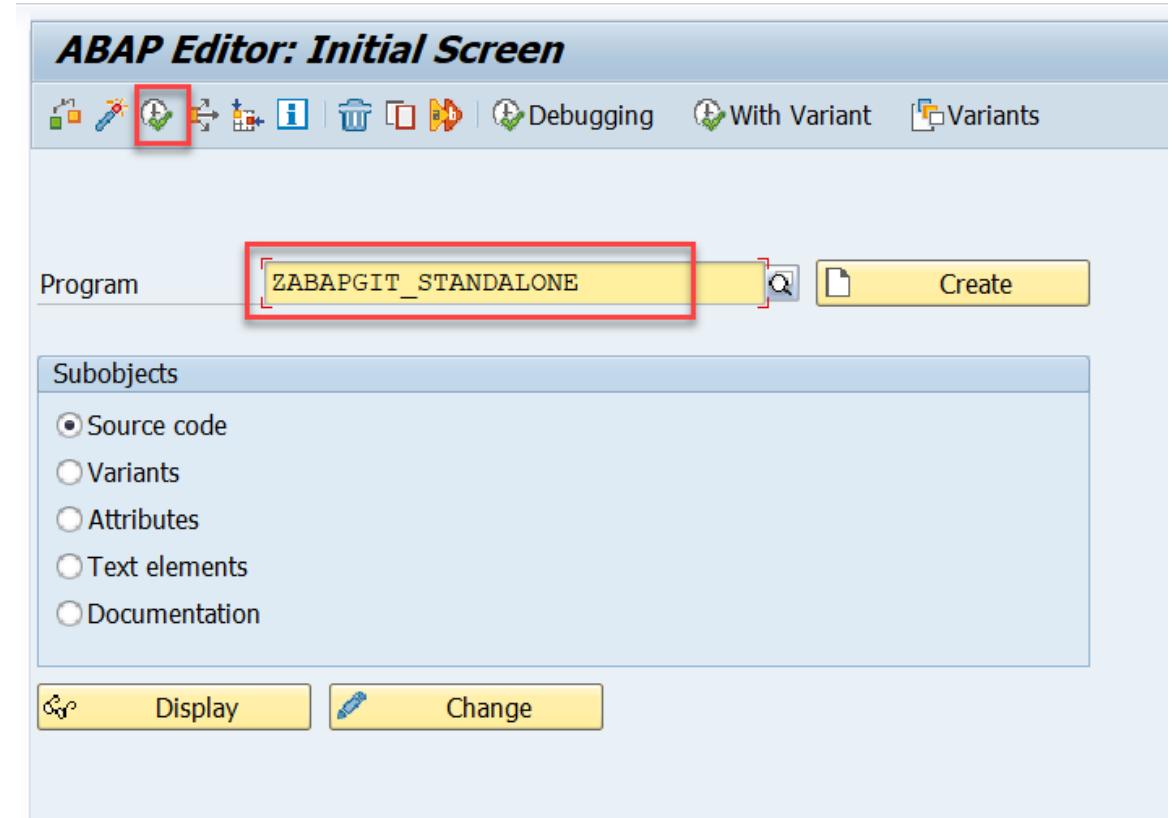
**Packages:**

No packages published  
[Publish your first package](#)

## STEP 4: Download ABAP code from GitHub

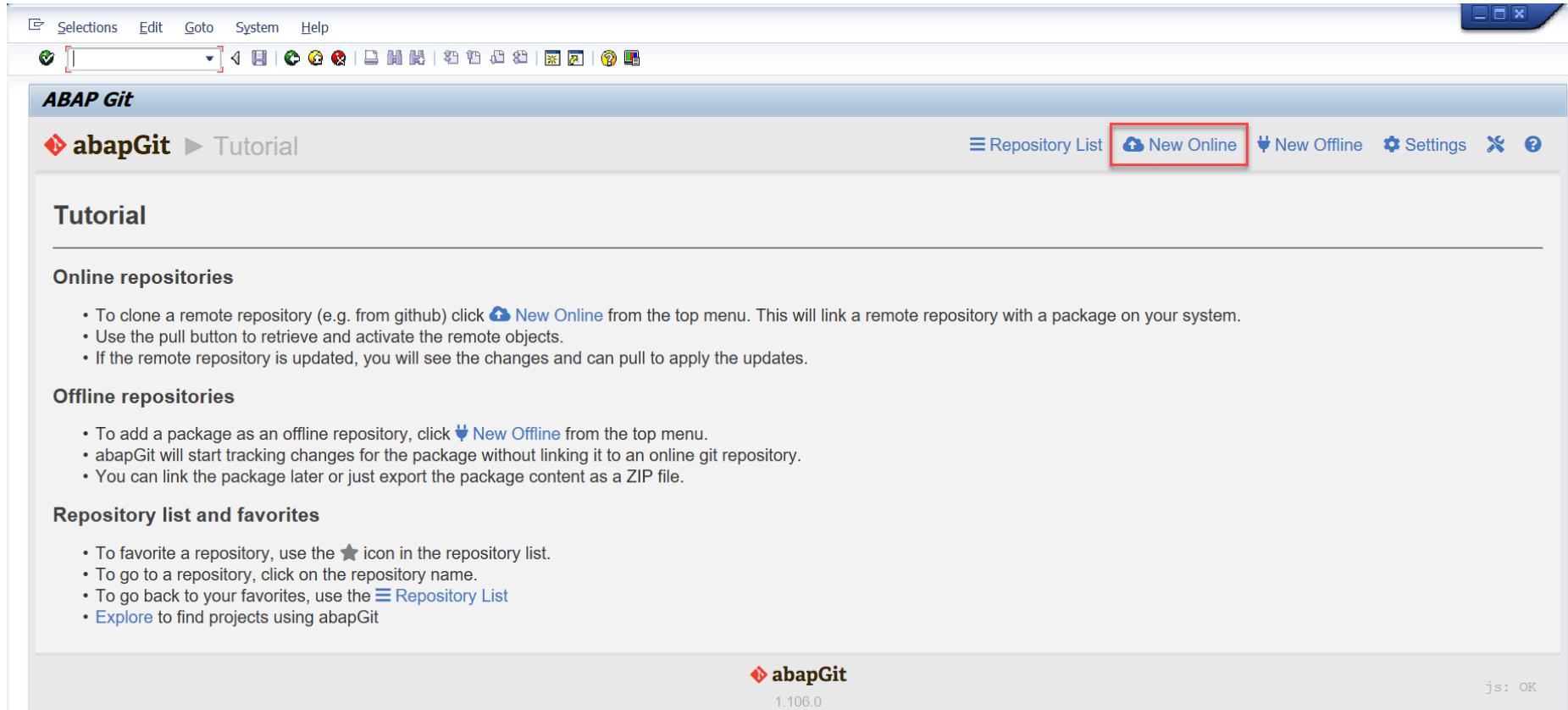
4-1: Enter T-code **SE38** and fill in the report name from STEP 1,  
**ZABAPGIT\_STANDALONE**

4-2: Click **Execute** to run the report



# STEP 4: Download ABAP code from GitHub

## 4-3: Click **New Online** to download the code



# STEP 4: Download ABAP code from GitHub

4-4: Fill in the **Git repository URL** in step 3-5:

<https://github.com/dev97619/logistics-business-network-gtt-samples.git>

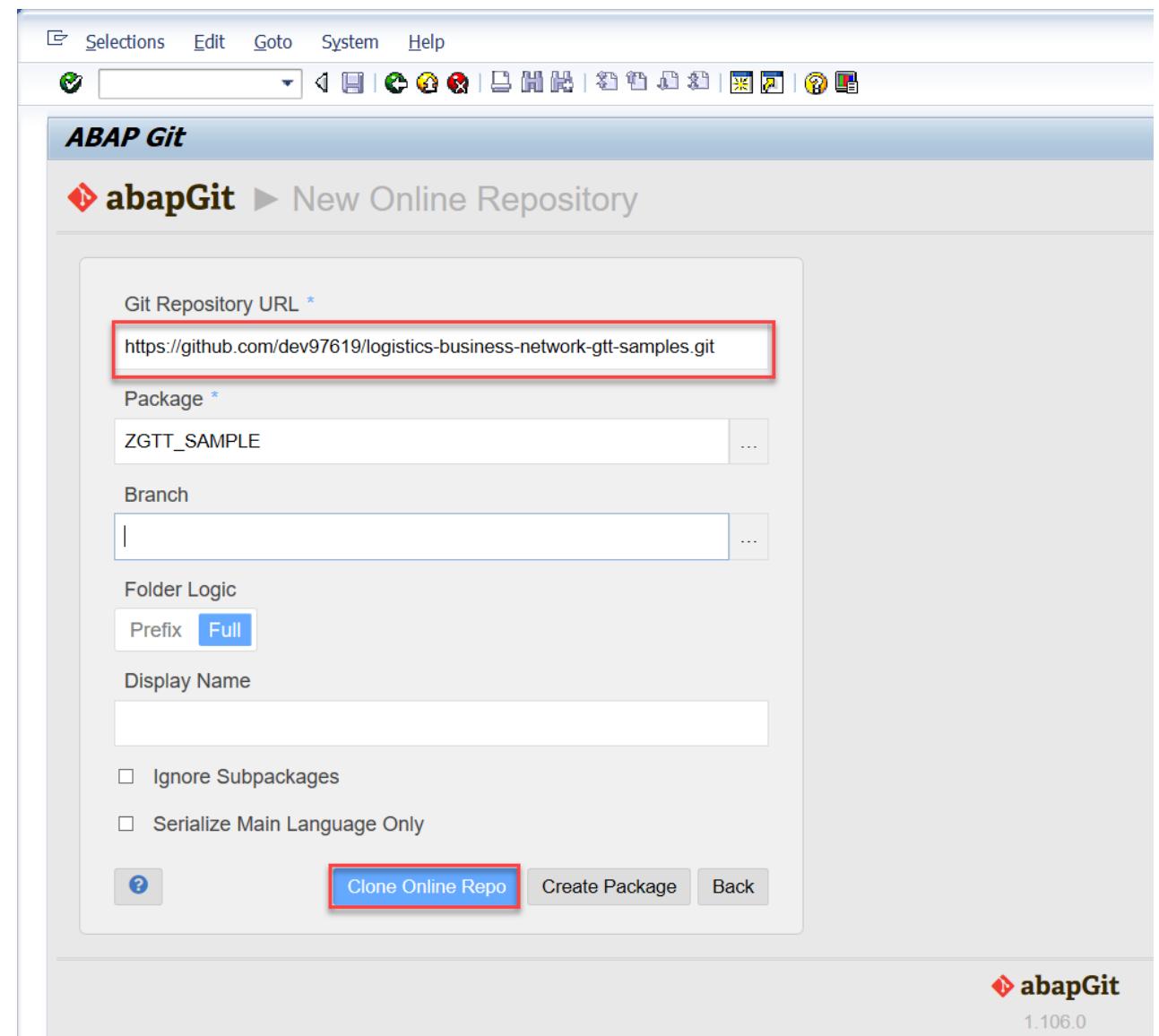
## Caution:

This URL is the user account's repository URL, not the public sample code's repository URL.

4-5: 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.

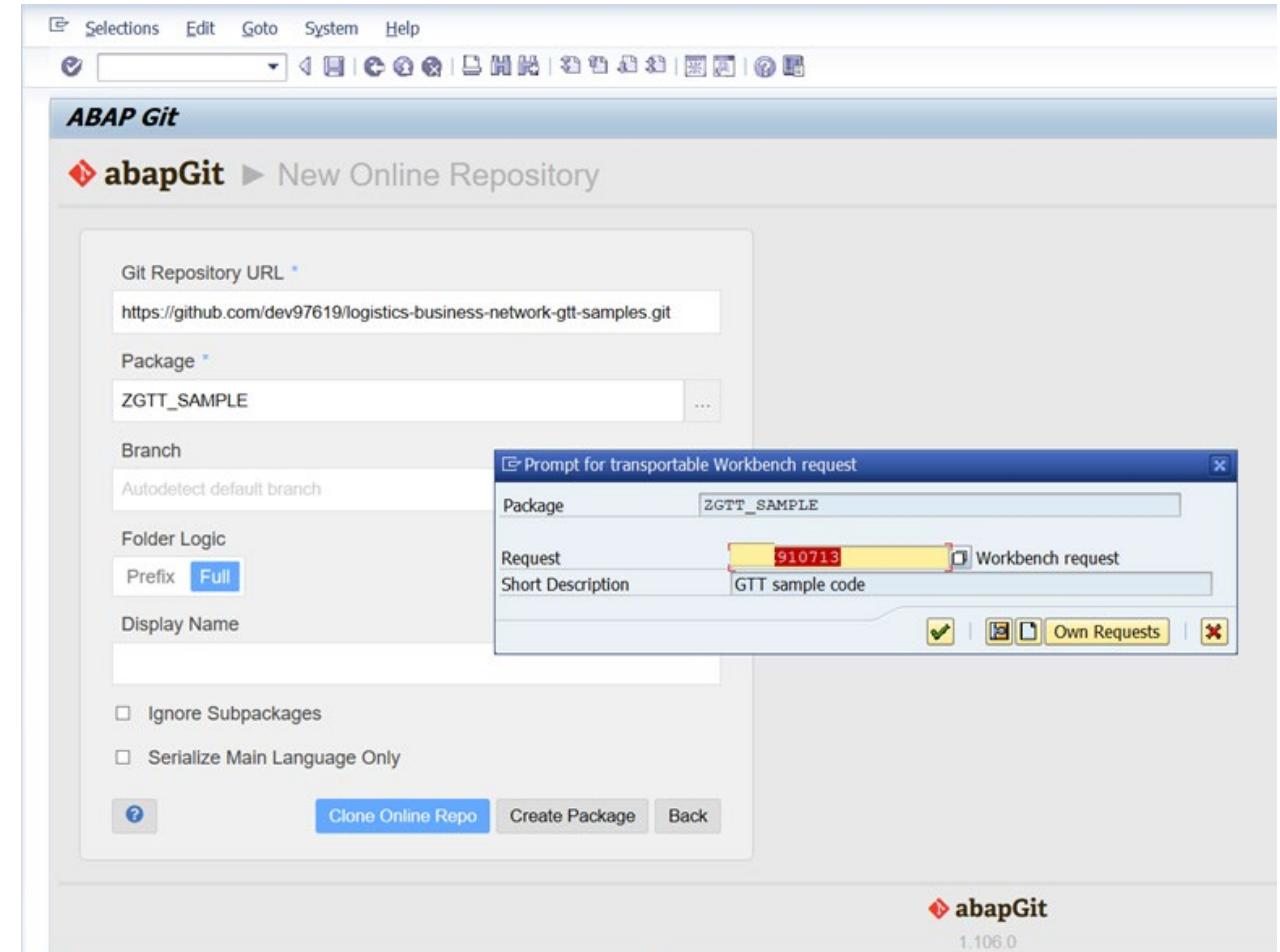
4-6: Set *Full* for **Folder Logic**

4-7: Click **Clone online repo** to download the code



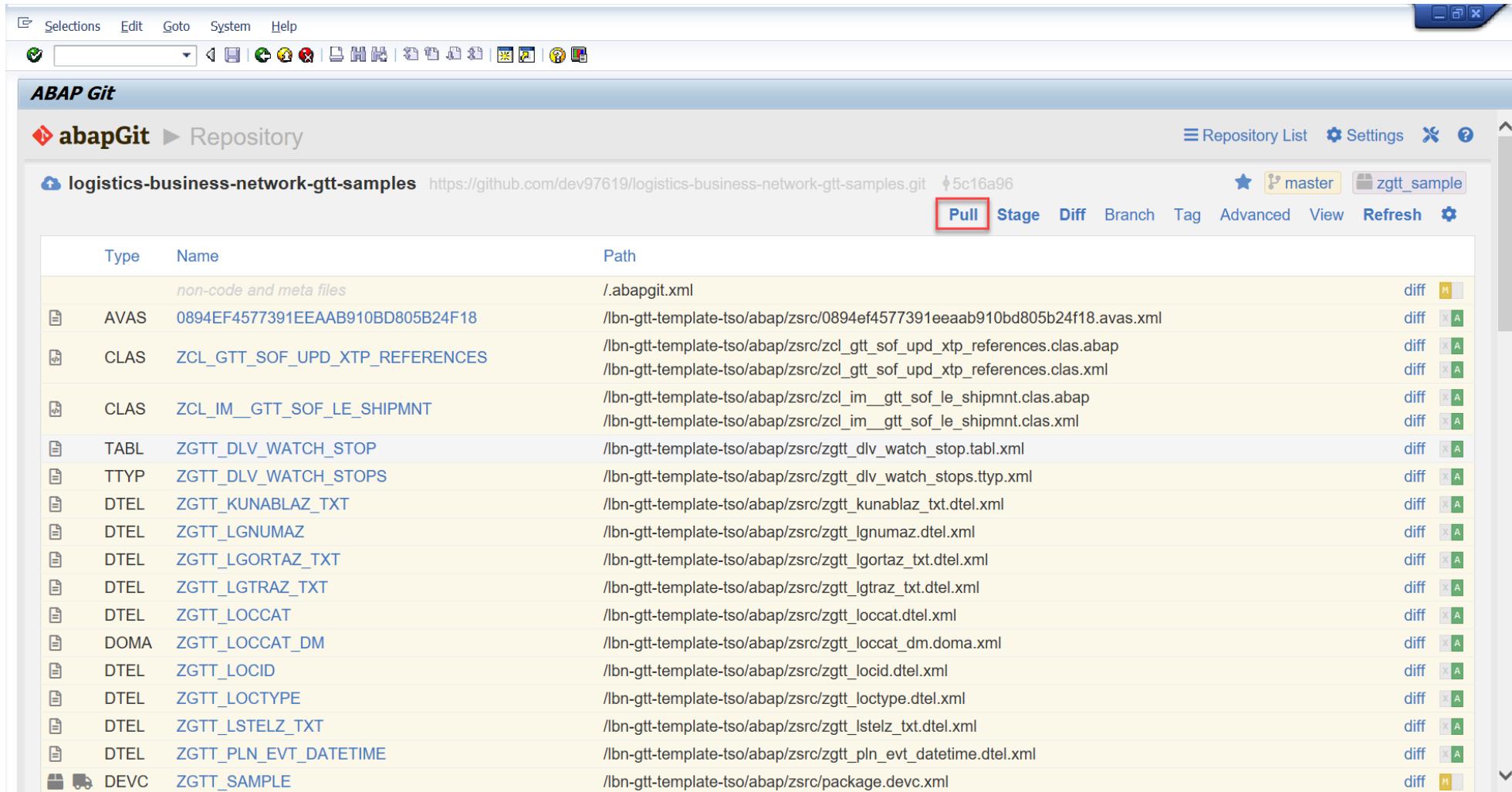
# STEP 4: Download ABAP code from GitHub

4-8: Assign the change to a change request. If you do not have any available change request, you need to create a new one.



# STEP 4: Download ABAP code from GitHub

4-9: Click Pull to pull down the latest version code

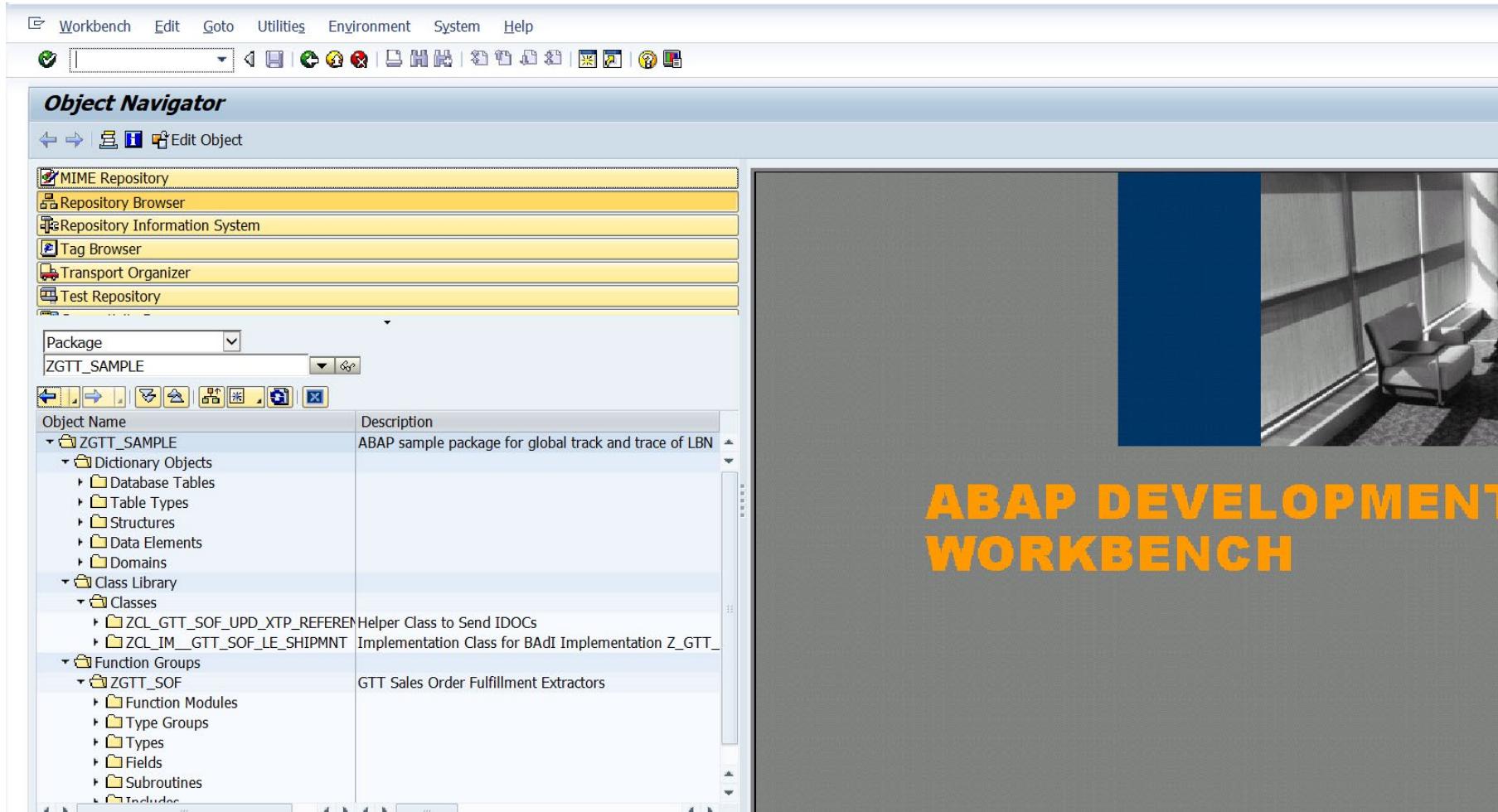


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows a repository named 'logistics-business-network-gtt-samples' with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples.git>. A commit hash '5c16a96' is displayed. The repository name is underlined. To the right of the repository name are buttons for 'master' and 'zgtt\_sample'. A navigation bar at the top right includes 'Repository List', 'Settings', and other options. Below the navigation bar is a toolbar with buttons for 'Pull' (highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a gear icon. The main content area is a table with columns 'Type', 'Name', and 'Path'. The 'Type' column shows file extensions like AVAS, CLAS, TABL, TTYP, DTEL, DOMA, and DEV. The 'Name' column lists file names such as '0894EF4577391EEAAB910BD805B24F18', 'ZCL\_GTT\_SOUPD\_XTP\_REFERENCES', 'ZCL\_IM\_GTT\_SOFL\_SHIPMNT', etc. The 'Path' column shows the full path for each file. On the far right of the table are 'diff' buttons and 'A' status indicators.

Type	Name	Path	
	non-code and meta files	./abapgit.xml	diff M
AVAS	0894EF4577391EEAAB910BD805B24F18	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	diff A
CLAS	ZCL_GTT_SOUPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	diff A diff A
CLAS	ZCL_IM_GTT_SOFL_SHIPMNT	/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	diff A diff A
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	diff A
TTYP	ZGTT_DLV_WATCH_STOPS	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	diff A
DTEL	ZGTT_KUNABLAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_kunablaz_txt.dtel.xml	diff A
DTEL	ZGTT_LGNUMAZ	/lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml	diff A
DTEL	ZGTT_LGORTAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml	diff A
DTEL	ZGTT_LGTRAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	diff A
DTEL	ZGTT_LOCCAT	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat.dtel.xml	diff A
DOMA	ZGTT_LOCCAT_DM	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat_dm.doma.xml	diff A
DTEL	ZGTT_LOCID	/lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	diff A
DTEL	ZGTT_LOCTYPE	/lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	diff A
DTEL	ZGTT_LSTELZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	diff A
DTEL	ZGTT_PLN_EVT_DATETIME	/lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml	diff A
DEV	ZGTT_SAMPLE	/lbn-gtt-template-tso/abap/zsrc/package.devc.xml	M

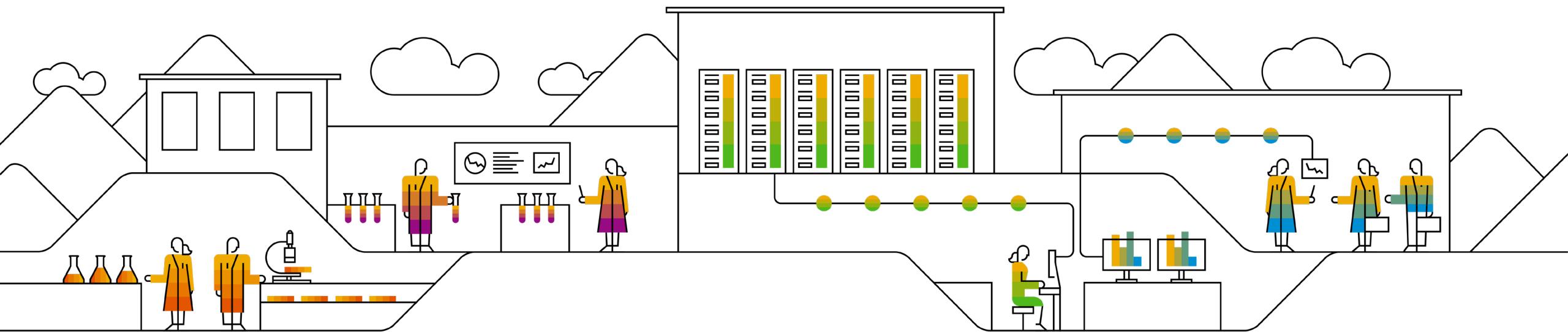
# STEP 4: Download ABAP code from GitHub

4-10: After you download the code, you can check them with T-code *SE80*.



# C) Download ABAP Code from GitHub

## C2. Update ABAP code from GitHub(Only for TSO)



# STEP 1: Delete the user's Account Repository

1-1: Assume you've already installed the sample code of TSO to your local SAP system with the version of the previous release, in the latest release, there will be some code changes in public sample code, you need to update the local code according to the latest public sample code.

1-2: Navigate to the user's account's repository, click "settings"

The screenshot shows a GitHub repository page for the user 'dev97619' with the repository name 'logistics-business-network-gtt-samples'. The 'Settings' tab is highlighted with a red box. The repository has 0 stars, 0 forks, and 16 issues. The description of the repository states: 'This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.' The sidebar includes sections for About, Releases, and Packages.

This branch is 2 commits ahead of SAP-samples:master.

Commit	Message	Time Ago
dev97619 Update .abapgit.xml	update .abapgit.xml	29 minutes ago
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
lbn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
lbn-gtt-template-ts	Update README.md	10 days ago
lbn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
.abapgit.xml	Update .abapgit.xml	29 minutes ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

# STEP 1: Delete the user's Account Repository

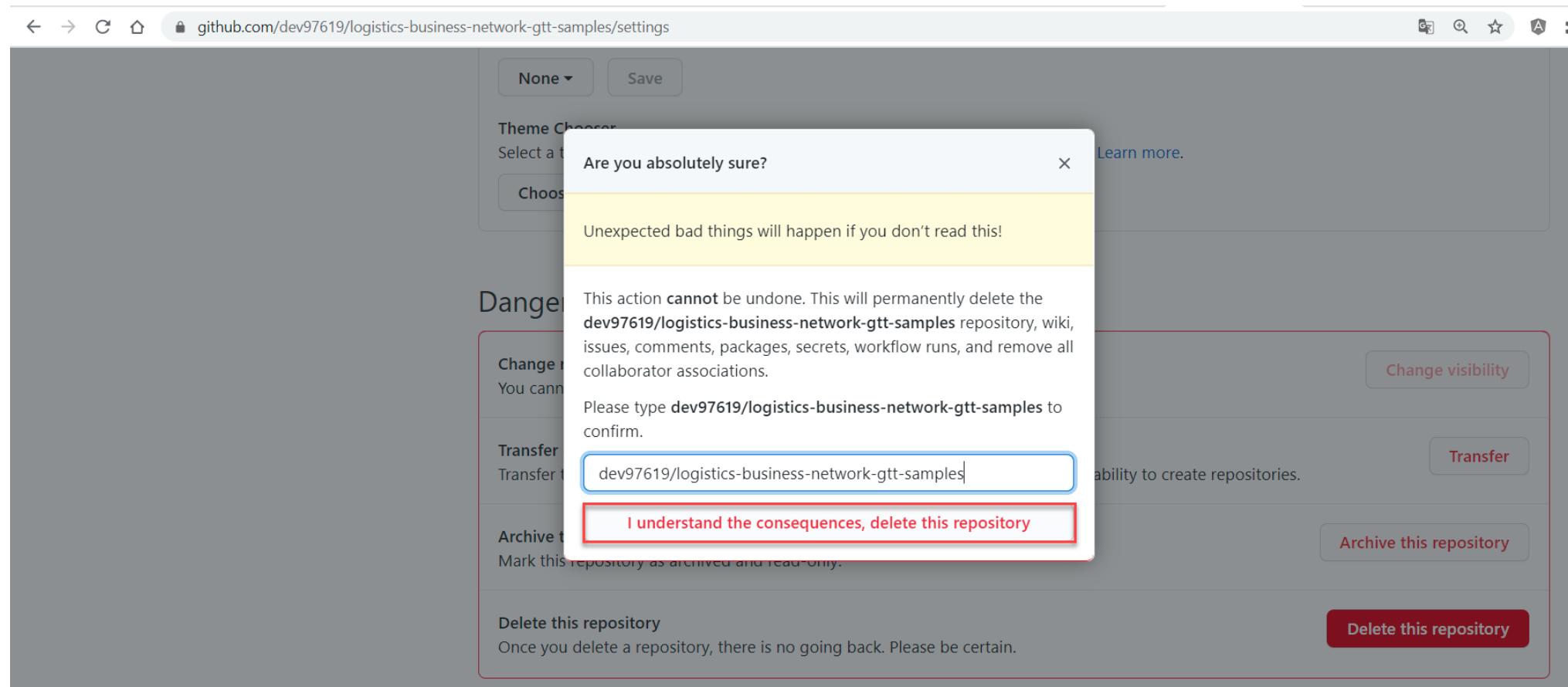
1-3: Scrolling down and find the button “Delete this repository” and click it.

The screenshot shows a GitHub repository settings page for the URL `github.com/dev97619/logistics-business-network-gtt-samples/settings`. At the top, there is a "Theme Chooser" section with a "None" dropdown and a "Save" button. Below this is a "Danger Zone" section with four options:

- Change repository visibility**: You cannot change the visibility of a fork. Please [duplicate the repository](#). A "Change visibility" button is present.
- Transfer ownership**: Transfer this repository to another user or to an organization where you have the ability to create repositories. A "Transfer" button is present.
- Archive this repository**: Mark this repository as archived and read-only. A "Archive this repository" button is present.
- Delete this repository**: Once you delete a repository, there is no going back. Please be certain. This button is highlighted with a red border.

# STEP 1: Delete the user's Account Repository

1-4: The popup shows some warning messages and follow the step it mentioned, click the button “I understand the consequences, delete this repository”



# STEP 1: Delete the user's Account Repository

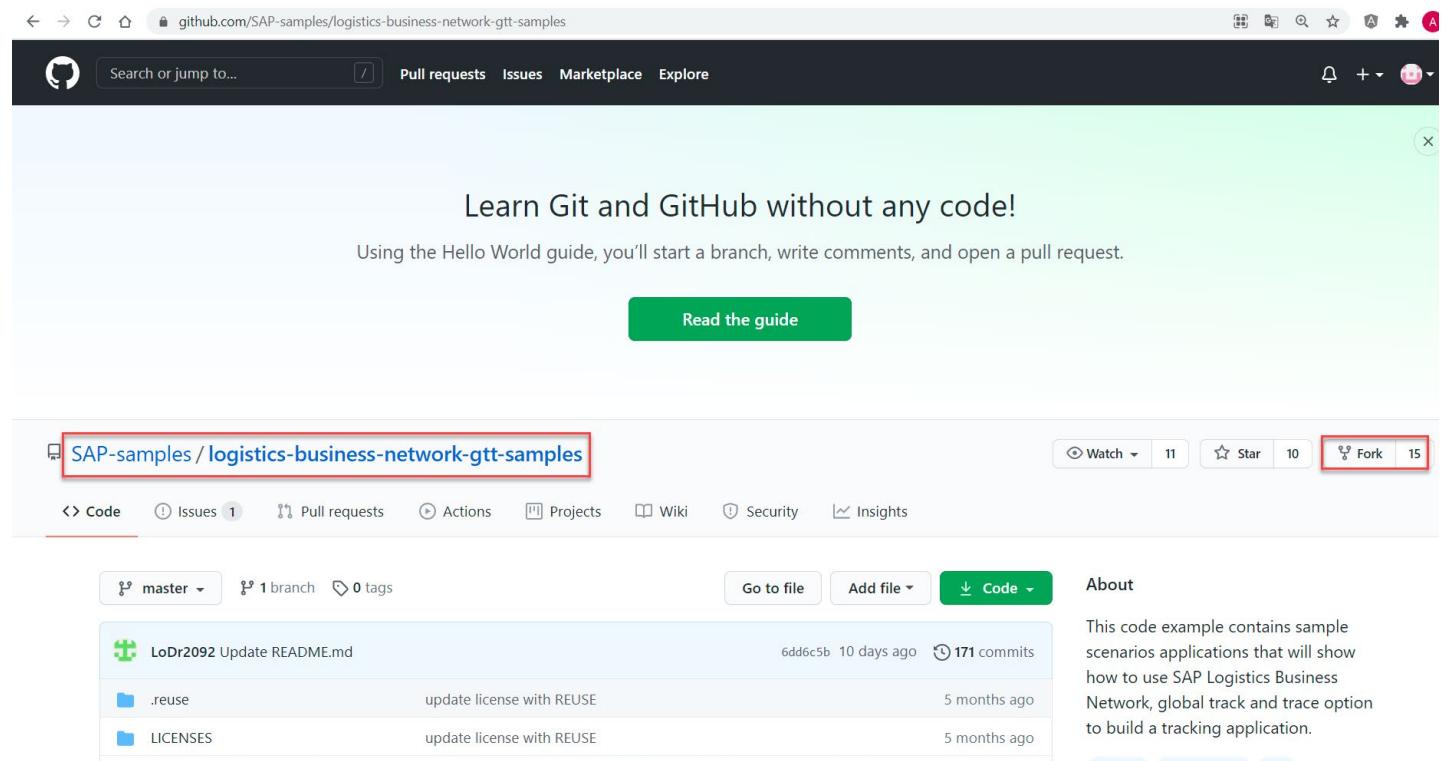
1-5: The user account's repository is deleted.

The screenshot shows a dark-themed GitHub interface. At the top, there is a navigation bar with a search bar, a pull requests button, issues, marketplace, and explore links. On the right side of the header are notifications, a plus sign for creating new items, and a profile icon. A prominent red rectangular box highlights a success message in a light blue banner: "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." Below this, the main content area features a "Create your first project" section with a "Create repository" button (which is green) and an "Import repository" link. To the right, a large callout box titled "Learn Git and GitHub without any code!" encourages users to use the Hello World guide to learn basic GitHub operations. It includes a "Read the guide" button (green) and a "Start a project" button (white). On the far left, there is a "Working with a team?" section and a "Create an organization" link.

# STEP 2: Fork Sample code Repository

2-1: Navigate to sample code in  
<https://github.com/SAP-samples/logistics-business-network-gtt-samples>

2-2: Click the “Fork” button, it will copy the newest version of sample codes into the user’s account and meanwhile it will navigate to user’s own repository



# STEP 2: Fork Sample code Repository

2-3: The newest version of sample codes copied to the user's account

github.com/dev97619/logistics-business-network-gtt-samples

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with SAP-samples:master.

Go to file Add file Code

Pull request Compare

Commit	Message	Date
LoDr2092 Update README.md	update license with REUSE	6dd6c5b 10 days ago
.reuse	update license with REUSE	5 months ago
LICENSES	change folder name from ABAP to abap	5 months ago
Ibn-gtt-template-tpo	Update README.md	11 days ago
Ibn-gtt-template-ts	change folder name from ABAP to abap	10 days ago
Ibn-gtt-template-tso	Update .abapgit.xml	11 days ago
.abapgit.xml	Update README.md	13 days ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

About

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

Readme

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

# STEP 3: Change Configuration file ‘.abapgit.xml’

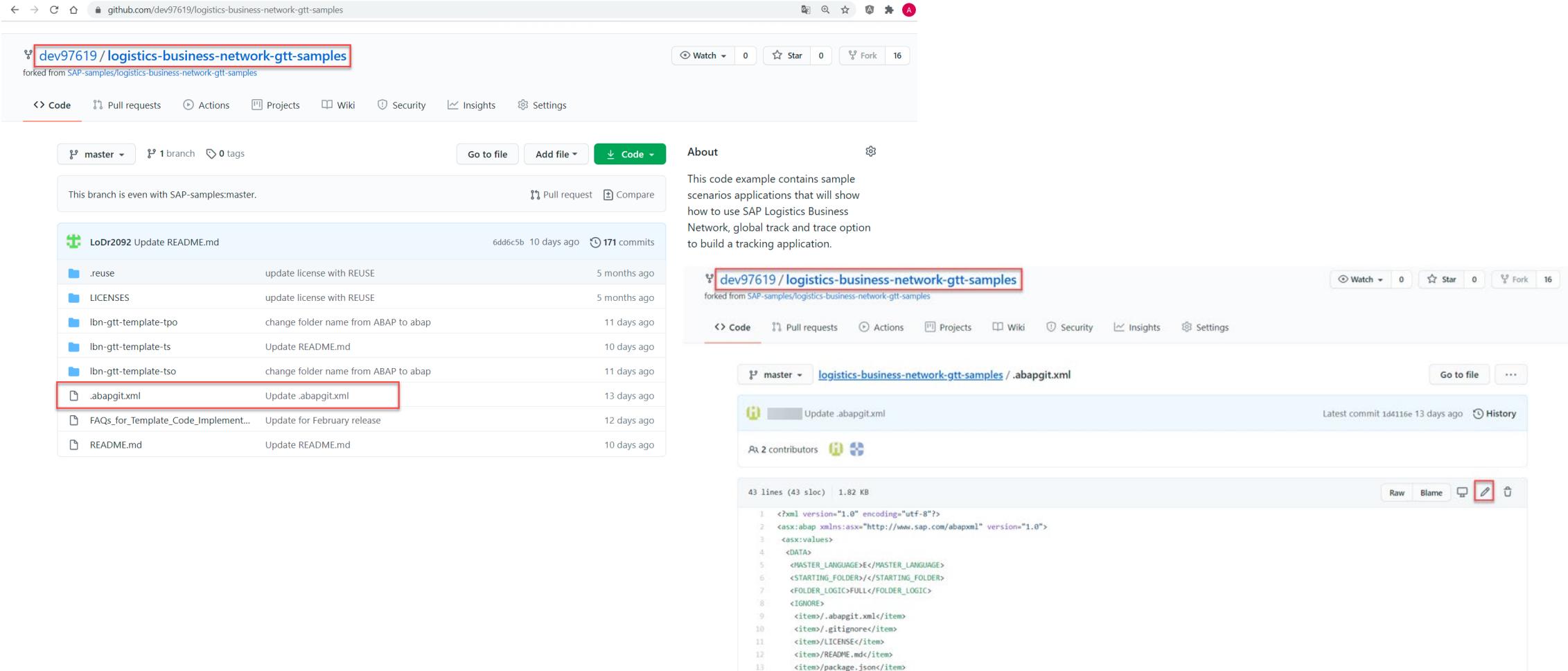
3-1: In the user account’s repository, click the file ‘.abapgit.xml’

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. A pull request button and a compare link are present. The commit history lists several changes, including one by LoDr2092 that updates the README.md file. A specific commit to '.abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgit.xml'. The commit was made 13 days ago by LoDr2092. Other commits listed include updates to LICENSES, folder names, and README files. To the right of the commit list, there is an 'About' section describing the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and a 'Readme' link. Below that is a 'Releases' section stating 'No releases published' and a 'Create a new release' link. At the bottom is a 'Packages' section stating 'No packages published' and a 'Publish your first package' link.

Commit	Message	Time Ago
LoDr2092 Update README.md	update license with REUSE	5 months ago
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
<b>.abapgit.xml</b>	<b>Update .abapgit.xml</b>	<b>13 days ago</b>
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

# STEP 3: Change Configuration file ‘.abapgit.xml’

3-2: Click  button to edit the file



The screenshot shows two GitHub repository pages. The top page is for the repository `dev97619 / logistics-business-network-gtt-samples`. The bottom page is for the specific file `logistics-business-network-gtt-samples/.abapgit.xml`.

**Top Repository Page:**

- Branch: master
- Commits: 171 commits
- Latest commit: LoDr2092 Update README.md (6dd6c5b, 10 days ago)
- File: .abapgit.xml (highlighted with a red box)

**Bottom File Page:**

- File: .abapgit.xml
- Commit: Update .abapgit.xml (1d4116e, 13 days ago)
- Contributors: 2 contributors
- Code Preview:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER></STARTING_FOLDER>
7       <FOLDER_LOGICFULL></FOLDER_LOGIC>
8       <IGNORE>
9         <item>/.abapgit.xml</item>
10        <item>/.gitignore</item>
11        <item>/LICENSE</item>
12        <item>/README.md</item>
13        <item>/package.json</item>
```

## STEP 3: Change Configuration file '.abapgit.xml'

3-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with  
"<STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>" as follows.

3-4: Commit change

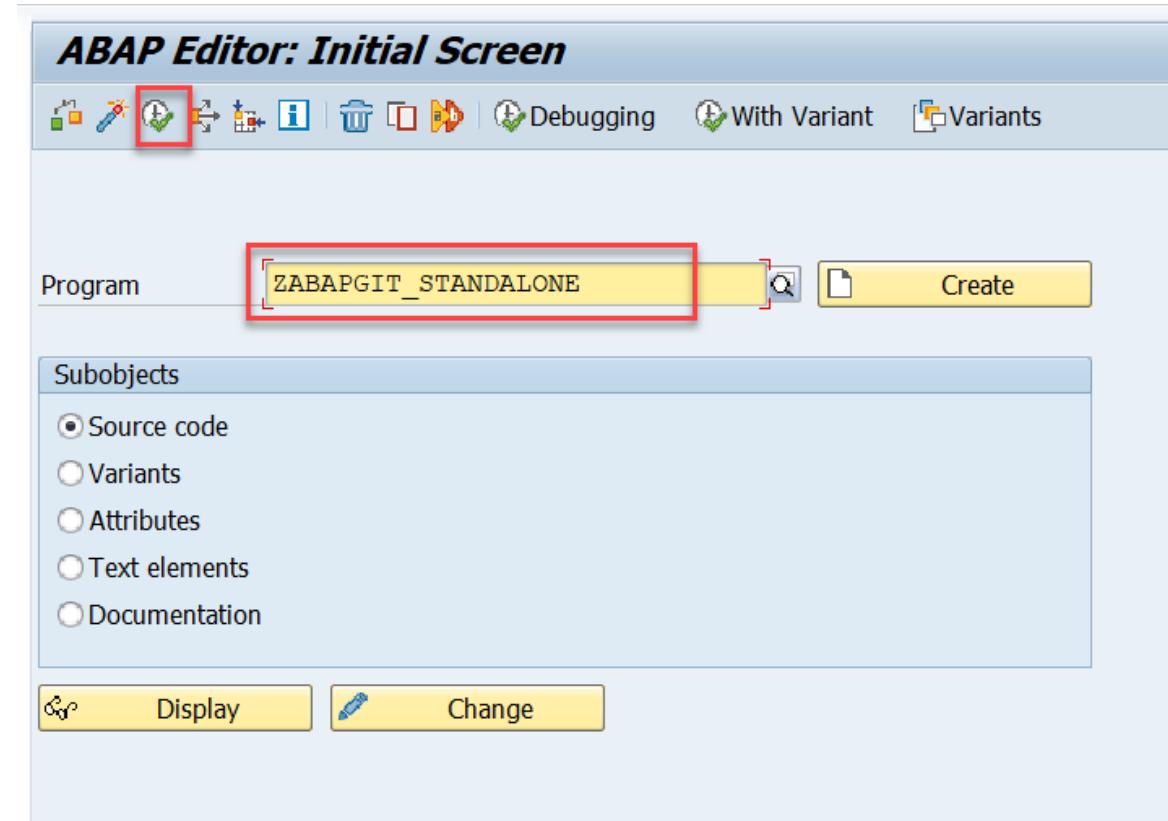
The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. On the right, a 'Commit changes' dialog is open for the file '.abapgit.xml'. The file content is displayed in the editor, showing XML code. A specific line is highlighted with a red box: '<STARTING\_FOLDER>/</STARTING\_FOLDER>'. In the commit message input field, the text 'Update .abapgit.xml' is entered. Below the message, there is an optional extended description field and two branch selection options: 'Commit directly to the master branch' (selected) and 'Create a new branch for this commit and start a pull request'. At the bottom of the dialog are 'Commit changes' and 'Cancel' buttons.

```
<?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-tso/abap/zsrc/</STARTING_FOLDER>
<FOLDER_LOGIC>FULL</FOLDER_LOGIC>
<IGNORE>
<item>/.abapgit.xml</item>
<item>/.gitignore</item>
```

## STEP 4: Update ABAP code from GitHub

4-1: Enter T-code *SE38* and fill in the report name **ZABAPGIT\_STANDALONE**

4-2: Click **Execute** to run the report



## STEP 4: Update ABAP code from GitHub

4-3: Check the URL is not changed after your recreation of repository copy. Access the TSO Repository by clicking button.

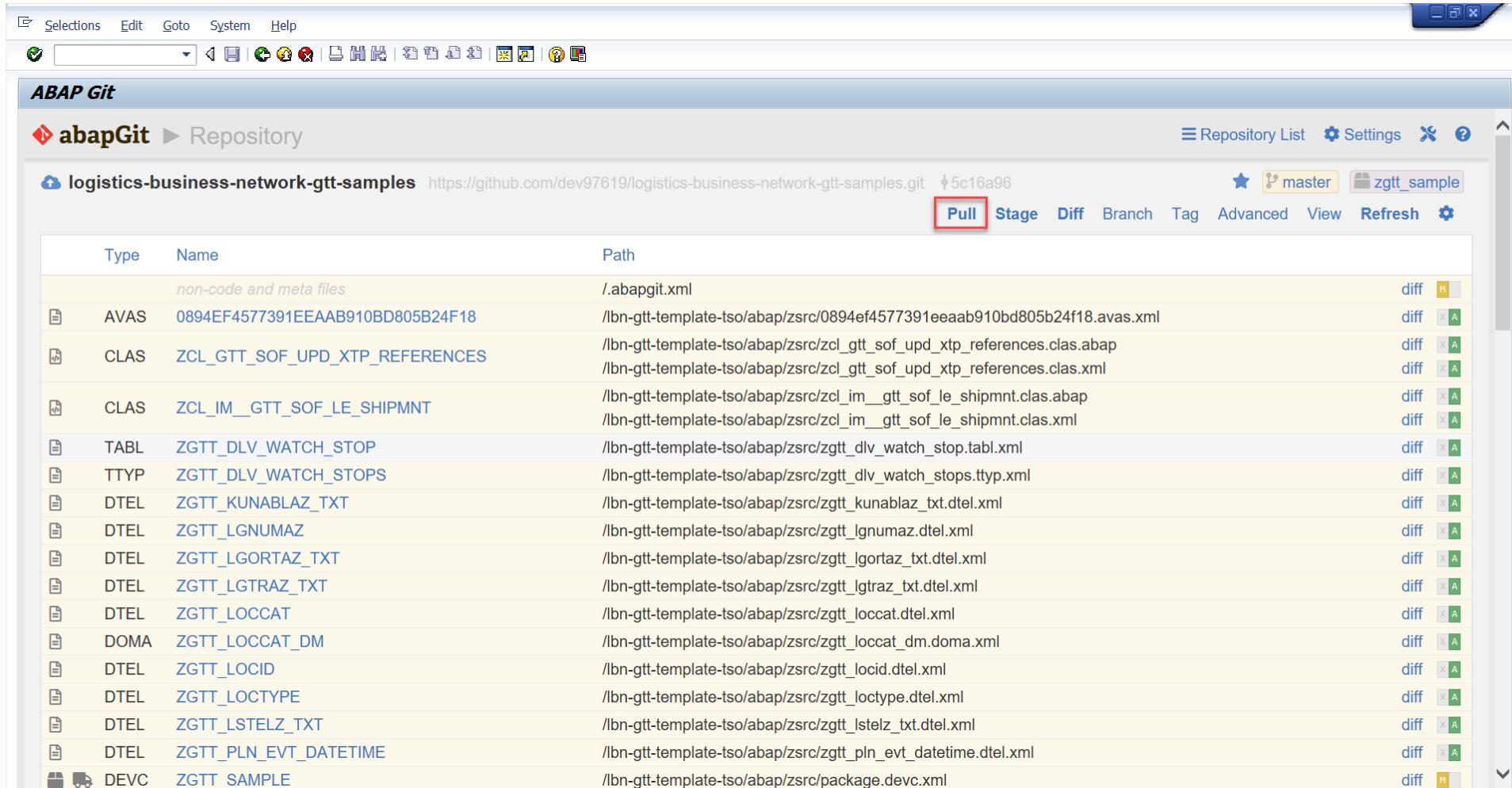
The screenshot shows the ABAP Git interface within a SAP application window. The title bar includes standard SAP menu items: Selections, Edit, Goto, System, Help, along with various toolbar icons. The main area is titled "ABAP Git" and displays a "Repository List". A header bar contains the "abapGit" logo, a "Repository List" link, and buttons for "New Online", "New Offline", "Settings", and help. Below this is a filter bar with a "Filter:" input field, "Only Favorites" checked, and "Detail" selected. The main table lists a single repository:

Name	Url	Package	Branch	Action
logistics-business-network-gtt-samples	github.com/dev97619/logistics-business-network-gtt-samples.git	zgtt_sample	master	Check   Stage   Patch   Settings

At the bottom center is the "abapGit" logo and version "1.106.0". On the far right, a status message "js: OK" is visible.

# STEP 4: Update ABAP code from GitHub

4-4: Click **Pull** to pull down the latest version code

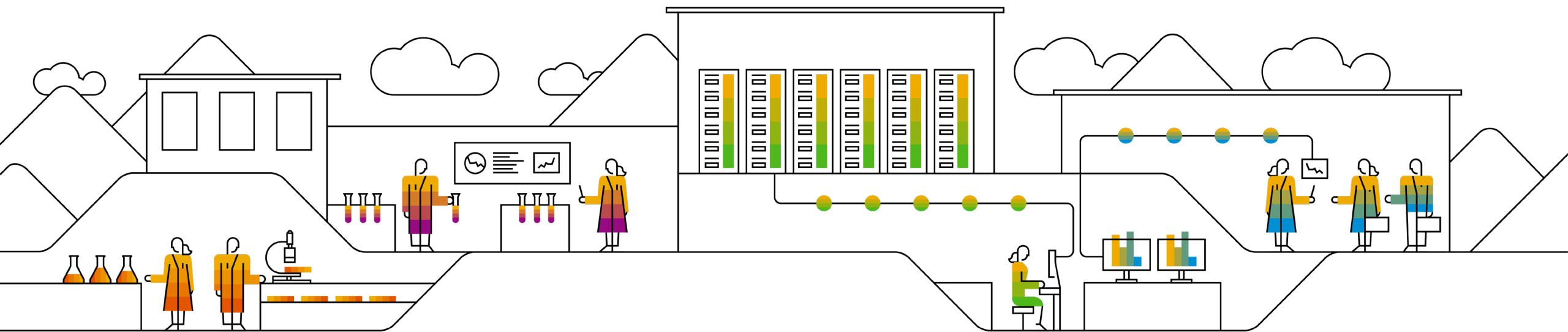


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git". Underneath, the repository path "abapGit > Repository" is shown, along with the URL "logistics-business-network-gtt-samples" and the commit hash "5c16a96". To the right of the URL, there are buttons for "master" and "zgtt\_sample". A navigation bar below the URL includes "Pull", "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The main area is a table with columns "Type", "Name", and "Path". The "Pull" button is highlighted with a red box. The table lists several files and their paths, such as "AVAS", "ZCL\_GTT\_SOF\_UPD\_XTP\_REFERENCES", "ZCL\_IM\_GTT\_SOF\_LE\_SHIPMNT", etc.

Type	Name	Path	diff	M
	non-code and meta files	/abapgit.xml	diff	M
AVAS	0894EF4577391EEAAB910BD805B24F18	//lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	diff	X A
CLAS	ZCL_GTT_SOF_UPD_XTP_REFERENCES	//lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	diff	X A
CLAS	ZCL_IM_GTT_SOF_LE_SHIPMNT	//lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	diff	X A
TABL	ZGTT_DLV_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	diff	X A
TTYP	ZGTT_DLV_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	diff	X A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_kunablaz_txt.dtel.xml	diff	X A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zggt_lgnuzaz.dtel.xml	diff	X A
DTEL	ZGTT_LGORAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgoraz_txt.dtel.xml	diff	X A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	diff	X A
DTEL	ZGTT_LOCAT	//lbn-gtt-template-tso/abap/zsrc/zggt_locat.dtel.xml	diff	X A
DOMA	ZGTT_LOCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zggt_locat_dm.doma.xml	diff	X A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	diff	X A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	diff	X A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	diff	X A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.datetime.dtel.xml	diff	X A
DEV	ZGTT_SAMPLE	//lbn-gtt-template-tso/abap/zsrc/package.devic.xml	diff	M

# C) Download ABAP Code from GitHub

## C3. Download Another ABAP code from GitHub(TPO)

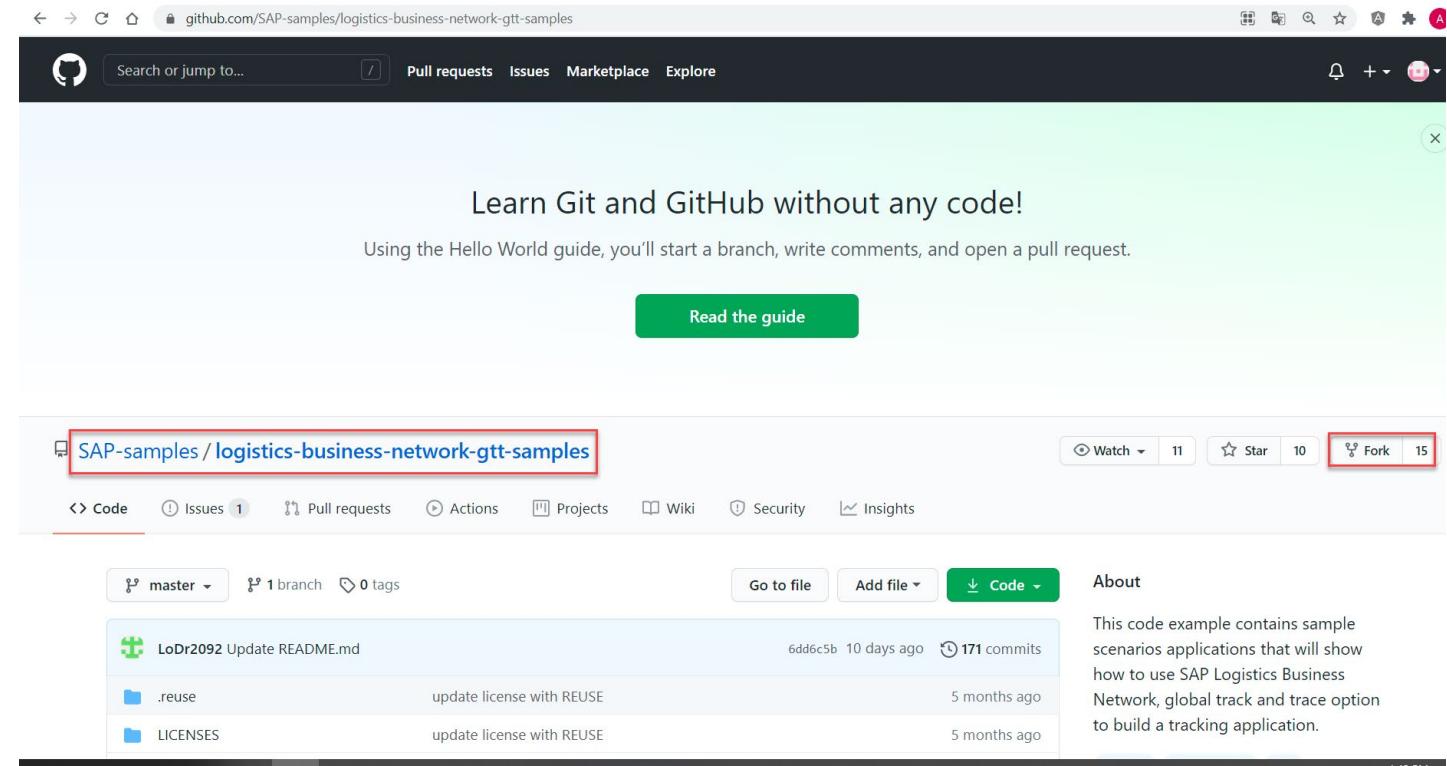


# STEP 1: Fork Sample code Repository

1-1. Assume that using ABAPGit, you have already installed the sample code of TSO to your local SAP system and meanwhile you also want to install sample code of TPO to your local SAP system.

1-2. Navigate to sample code in  
<https://github.com/SAP-samples/logistics-business-network-gtt-samples>

1-3. Click the “Fork” button, it will copy the newest version of sample codes into the user’s account and meanwhile it will navigate to user’s own repository



# STEP 2: Change Configuration file ‘.abapgit.xml’

2-1: In the user’s account repository, click the file ‘.abapgit.xml’

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one by LoDr2092 that updates the README.md file. A specific commit to '.abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgit.xml'. The repository has 0 stars, 0 forks, and 16 issues. The 'About' section describes the code example as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options, and building a tracking application. It includes links to 'Readme', 'Releases', and 'Packages'.

Commit	Message	Date
LoDr2092 Update README.md	update license with REUSE	6dd6c5b 10 days ago
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
<b>.abapgit.xml</b>	<b>Update .abapgit.xml</b>	13 days ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

# STEP 2: Change Configuration file ‘.abapgit.xml’

2-2: Click button to edit the file

The screenshot shows two GitHub repository pages. The top page is for the repository `dev97619 / logistics-business-network-gtt-samples`. The bottom page is for the file `.abapgit.xml` within that repository. The file content is as follows:

```
<?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>/</STARTING_FOLDER>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
      <IGNORE>
        <item>/abapgit.xml</item>
        <item>.gitignore</item>
        <item>LICENSES</item>
        <item>README.md</item>
        <item>package.json</item>
      </IGNORE>
    </DATA>
  </asx:values>
</asx:abap>
```

## STEP 2: Change Configuration file '.abapgit.xml'

2-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with "<STARTING\_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING\_FOLDER>" as follows.

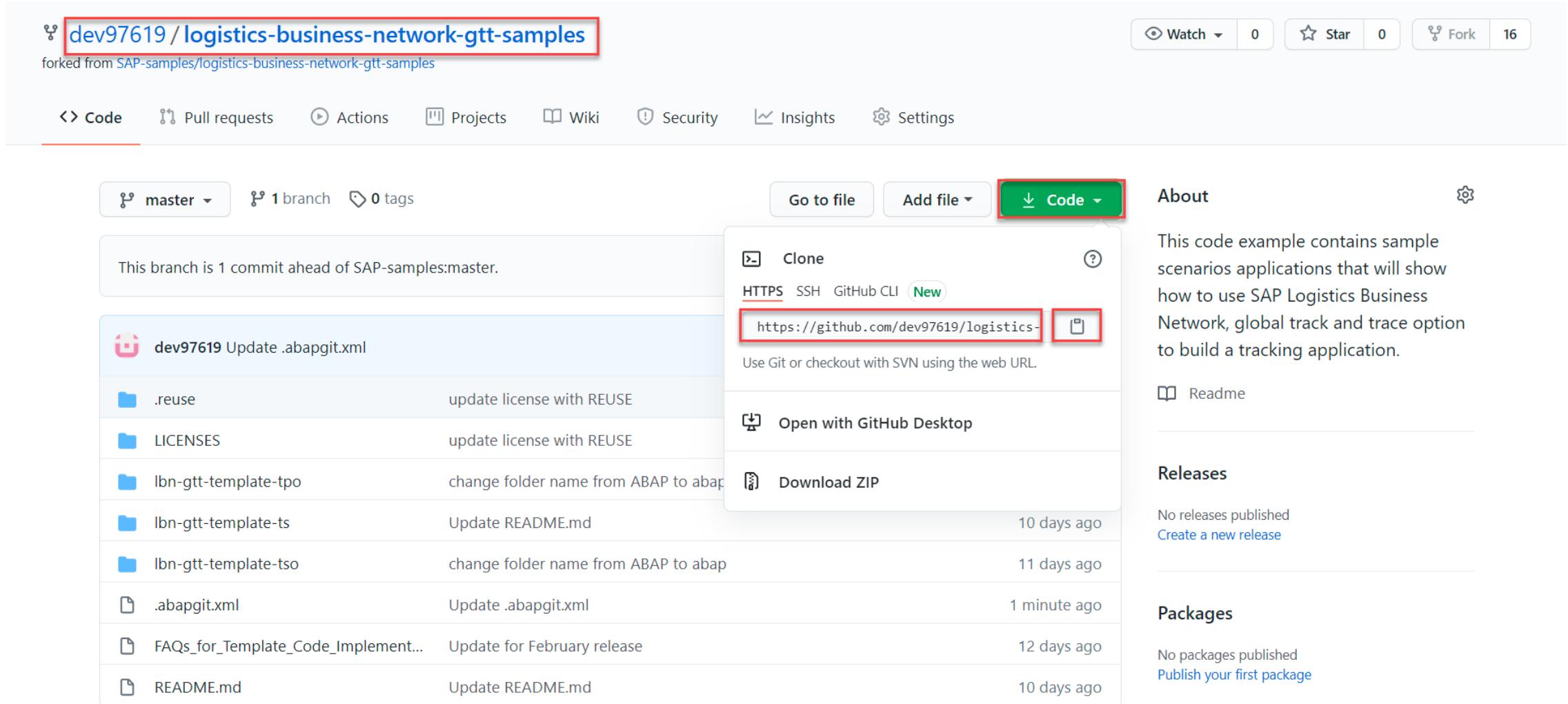
2-4: Commit change

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. In the code editor, the file '.abapgit.xml' is open, showing XML code. A red box highlights the line '<STARTING\_FOLDER>/</STARTING\_FOLDER>'. To the right, a 'Commit changes' dialog is open, containing fields for a commit message ('Update .abapgit.xml'), a description, and two radio button options for committing: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request.' At the bottom are 'Commit changes' and 'Cancel' buttons, with 'Commit changes' also highlighted by a red box.

```
<?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>/.abapgit.xml</item>
<item>/.gitignore</item>
```

## STEP 2: Change Configuration file '.abapgit.xml'

2-5: Go to the root and copy the repository URL by clicking  button



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository has 0 stars and 16 forks. The 'Code' tab is selected. The repository has 1 branch and 0 tags. The master branch is 1 commit ahead of SAP-samples:master. A dropdown menu is open over the 'Code' button, with the 'Clone' section highlighted. The URL 'https://github.com/dev97619/logistics...' is copied to the clipboard. The repository description states: 'This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.' It includes links to 'Readme', 'Releases', and 'Packages'.

dev97619 / logistics-business-network-gtt-samples

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is 1 commit ahead of SAP-samples:master.

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

Ibn-gtt-template-tpo change folder name from ABAP to abap

Ibn-gtt-template-ts Update README.md

Ibn-gtt-template-tso change folder name from ABAP to abap

.abapgit.xml Update .abapgit.xml

FAQs\_for\_Template\_Code\_Implement... Update for February release

README.md Update README.md

Code ▾

Clone

HTTPS SSH GitHub CLI New

https://github.com/dev97619/logistics- 

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published

Create a new release

Packages

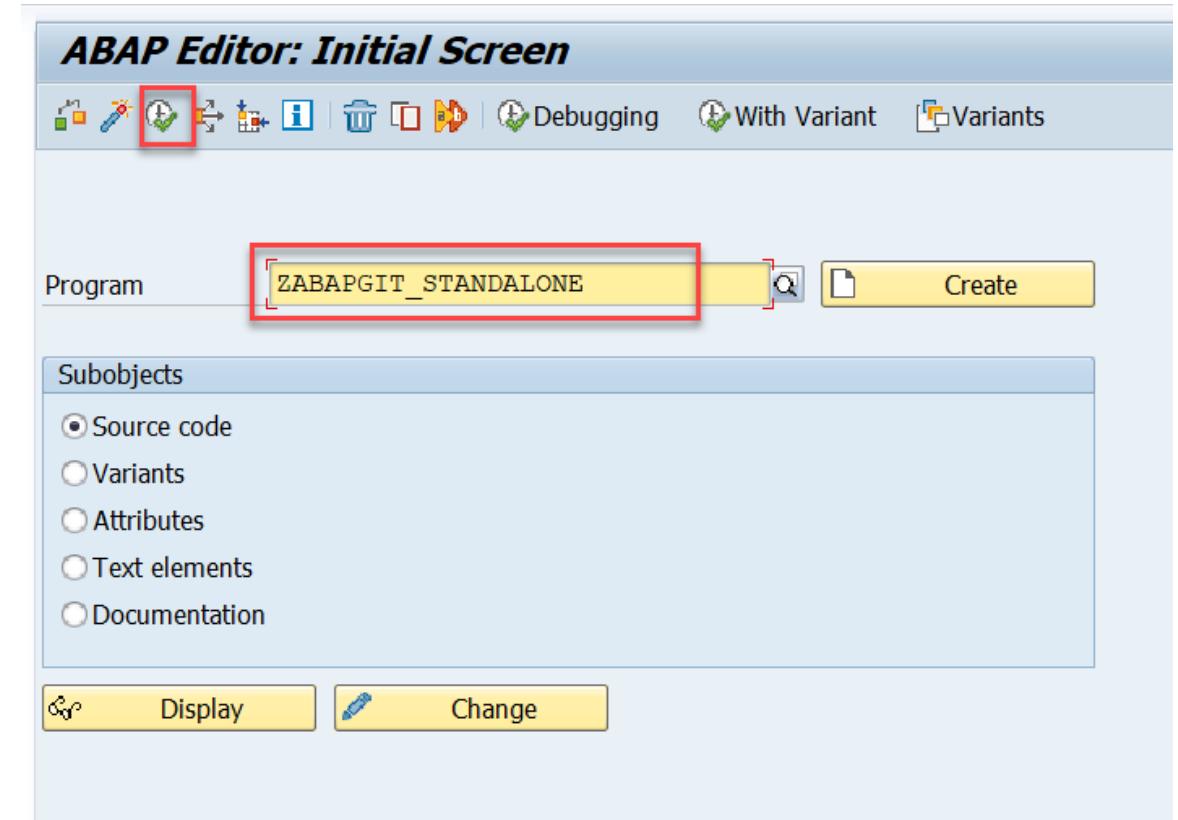
No packages published

Publish your first package

## STEP 3: Remove TSO Repository in ABAPGit

3-1: Enter T-code **SE38** and fill in the report name **ZABAPGIT\_STANDALONE**

3-2: Click **Execute** to run the report



# STEP 3: Remove TSO Repository in ABAPGit

3-3: Access the TSO Repository by clicking button

The screenshot shows the ABAP Git interface with the following details:

- Toolbar:** Selections, Edit, Goto, System, Help.
- Buttons:** Checkmark, Refresh, Save, Undo, Redo, etc.
- Title Bar:** ABAP Git
- Header:** abapGit ► Repository List, New Online, New Offline, Settings, ?
- Filter:** Filter: [ ] Only Favorites | Detail
- Table:** Repository List with columns: Name, Url, Package, Branch, Action.
- Rows:**
  - Name: logistics-business-network-gtt-samples, Url: github.com/dev97619/logistics-business-network-gtt-samples.git, Package: zgtt\_sample, Branch: master, Action: Check | Stage | Patch | Settings,
- Footer:** abapGit 1.106.0, js: OK

# STEP 3: Remove TSO Repository in ABAPGit

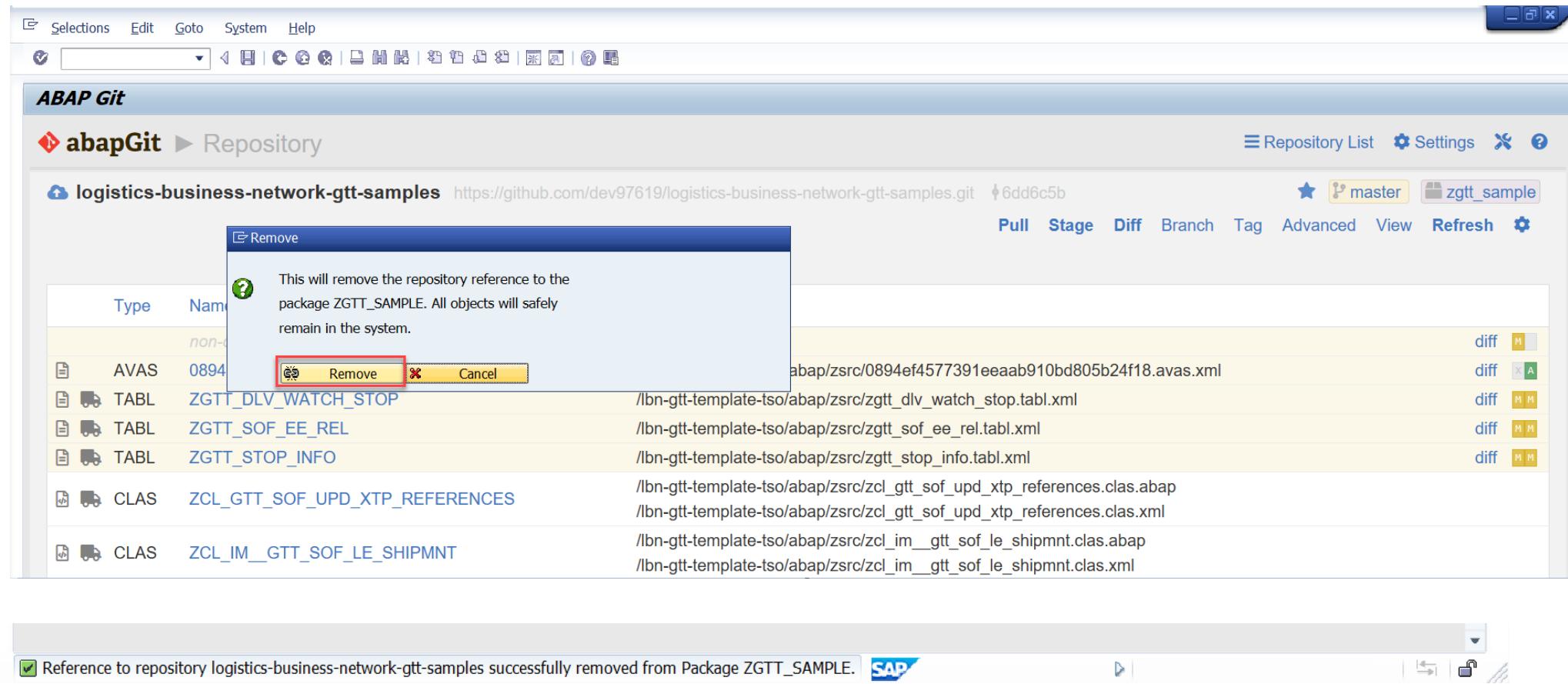
3-4: Choose sub menu “Remove” under the “Advanced” menu, click it.

The screenshot shows the ABAPGit interface within SAP. The top navigation bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the bar is a toolbar with various icons. The main title is 'ABAP Git' with a 'Repository' link. The repository details are shown: 'logistics-business-network-gtt-samples' at 'https://github.com/dev97619/logistics-business-network-gtt-samples.git' with commit hash '6dd6c5b'. The master branch is selected. A context menu is open over the repository name, with the 'Advanced' tab highlighted by a red box. The 'Advanced' tab has a submenu with several options: 'Reset Local (Force Pull)', 'Checkout commit', 'Background Mode', 'Change Remote', 'Make Off-line', 'Force Stage', 'Transport to Branch', 'Add all objects to transport request', 'Syntax Check', 'Run Code Inspector', 'Update Local Checksums', 'Beta - Data', 'Remove' (which is also highlighted by a red box), and 'Uninstall'.

Type	Name	Path
non-code and meta files		
AVAS	0894EF4577391EEAAB910BD805B24F18	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml
TABL	ZGTT_SOF_EE_REL	/lbn-gtt-template-tso/abap/zsrc/zggt_sof_ee_rel.tabl.xml
TABL	ZGTT_STOP_INFO	/lbn-gtt-template-tso/abap/zsrc/zggt_stop_info.tabl.xml
CLAS	ZCL_GTT_SOF_UPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen
CLAS	ZCL_IM_GTT_SOF_LE_SHIPMNT	/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt. /lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.
TTYP	ZGTT_DLV_WATCH_STOPS	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xr
DTEL	ZGTT_KUNABL陛_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_kunabl陛_txt.dtel.xml
DTEL	ZGTT_LGNUMAZ	/lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml
DTEL	ZGTT_LGORTAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml

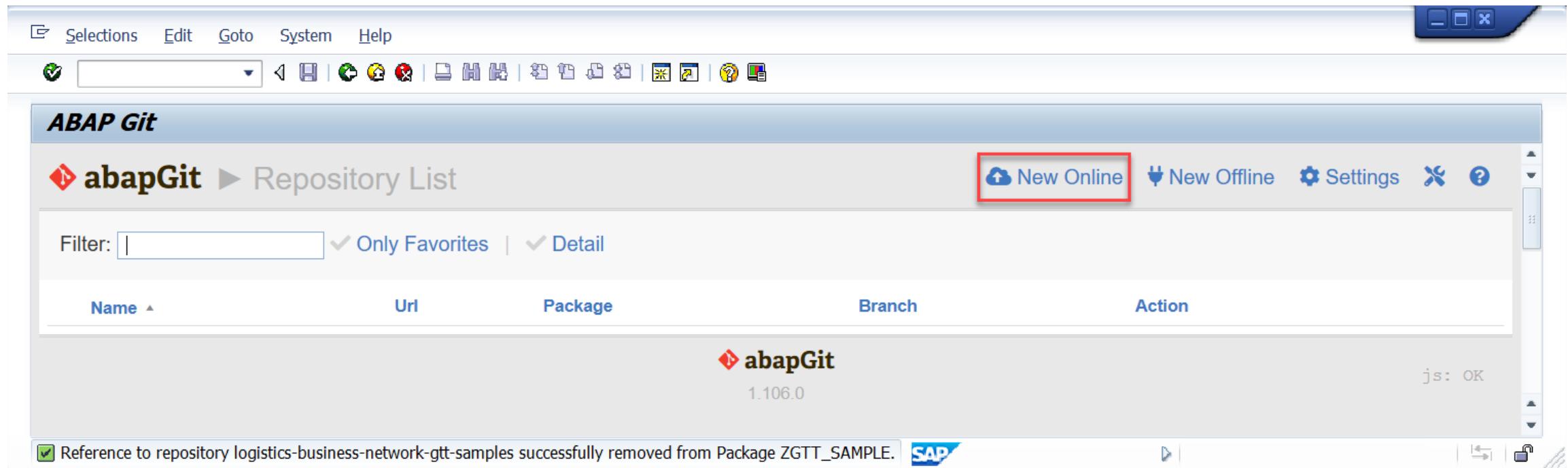
# STEP 3: Remove TSO Repository in ABAPGit

3-5: Click “Remove” button in the popup window, the reference to TSO repository will be removed.



# STEP 4: Download TPO code from GitHub

4-1: Click **New Online** to download the code



# STEP 4: Download TPO code from GitHub

4-2: Fill in the **Git repository URL** in step 2-5:

<https://github.com/dev97619/logistics-business-network-gtt-samples.git>

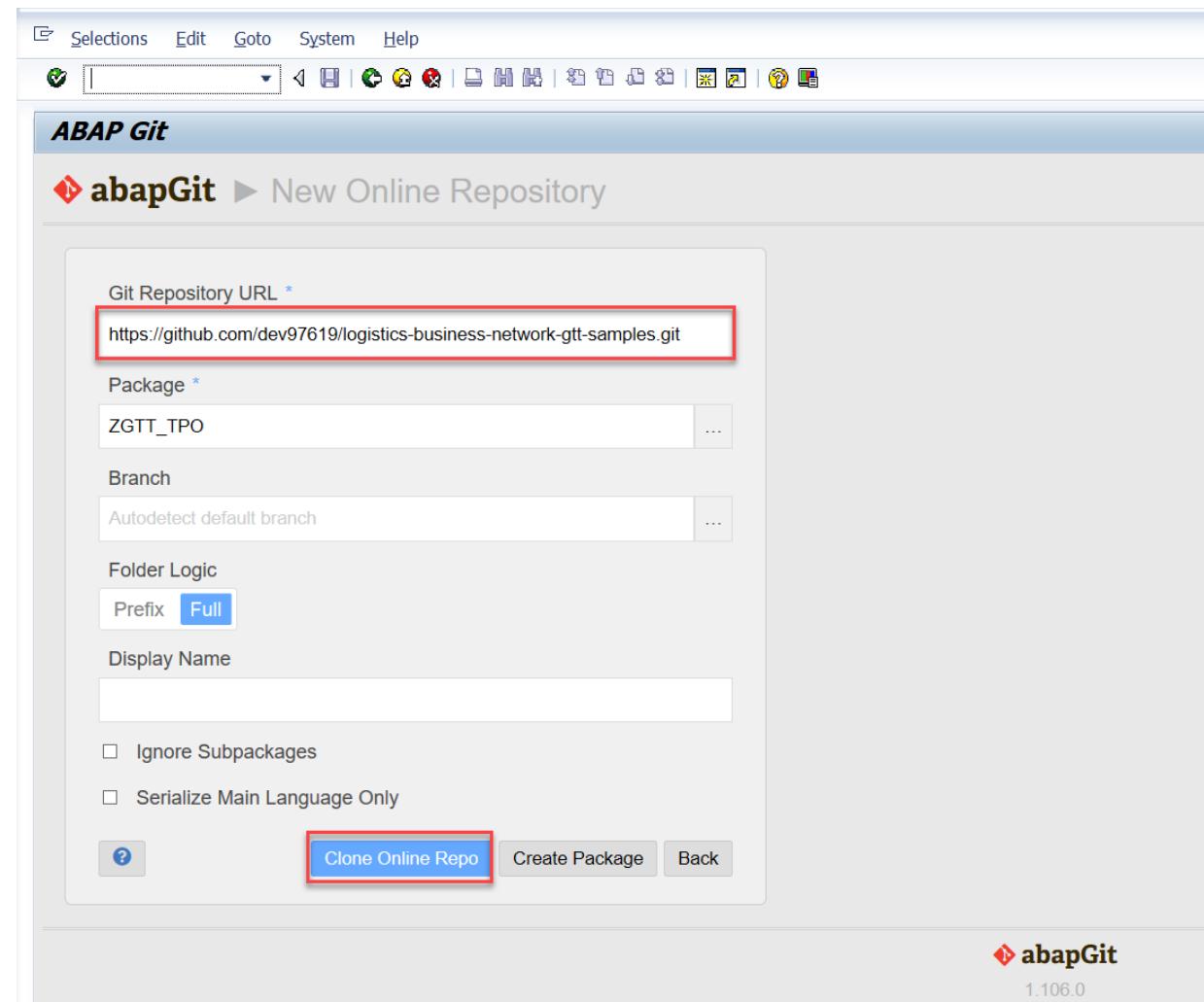
## Caution:

This URL is the user account's repository URL, not the public sample code's repository URL.

4-3: 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.

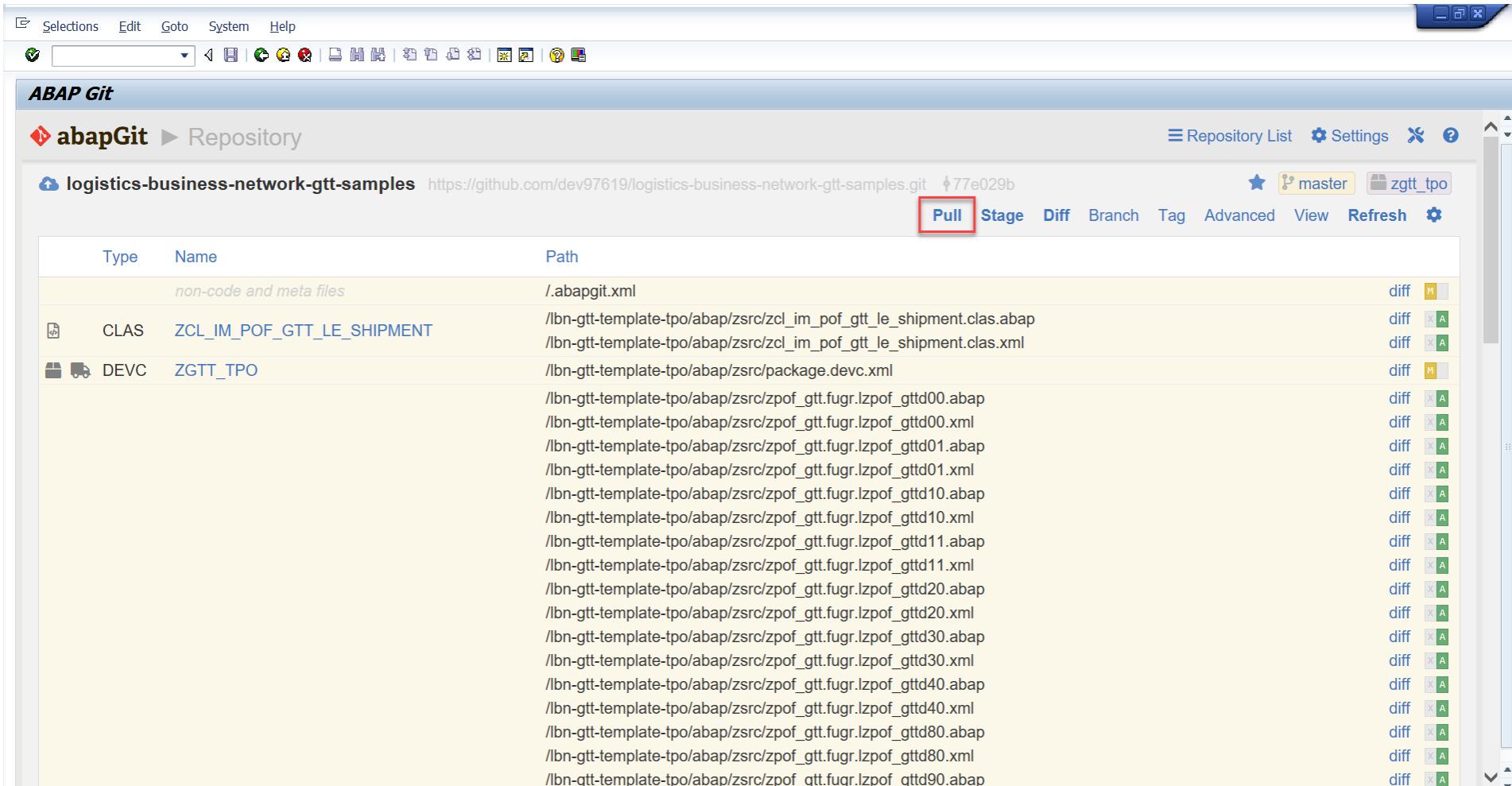
4-4: Set **Full** for **Folder Logic**

4-5: Click **Clone online repo** to download the code



# STEP 4: Download ABAP code from GitHub

4-6: Click **Pull** to pull down the latest version code



The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation "abapGit > Repository". The main area displays a list of files under the repository "logistics-business-network-gtt-samples" with the URL "https://github.com/dev97619/logistics-business-network-gtt-samples.git". The commit hash "77e029b" is shown. On the right side, there are buttons for "Repository List", "Settings", and "Refresh". Below these buttons, the "Pull" button is highlighted with a red box. The file list includes:

Type	Name	Path	diff
non-code and meta files			
		/.abapgit.xml	diff H
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	diff A
DEVC	ZGTT_TPO	/lbn-gtt-template-tpo/abap/zsrc/package.devc.xml	diff H
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt40.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt40.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt80.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt80.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt90.abap	diff A

# C) Download ABAP Code from GitHub

C4. Initial Download ABAP code from GitHub(include TSO/TPO/TS)



# STEP 1: Install ABAPGit

You need to install ABAPGit before downloading the code from GitHub.

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

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

When installation is complete, a new report is created, **ZABAPGIT\_STANDALONE**.

 abapGit › documentation

**Getting Started**

- Installation
- Upgrading
- Uninstalling
- UI features

**Setup**

- SSL setup
- Proxy configuration
- Development version

**Online Projects**

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

**Offline Projects**

- Import zip
- Export zip

**Reference**

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

**Installation**

 [Improve this page](#)

**Summary #**

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

- The standalone version is targeted at users. It 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

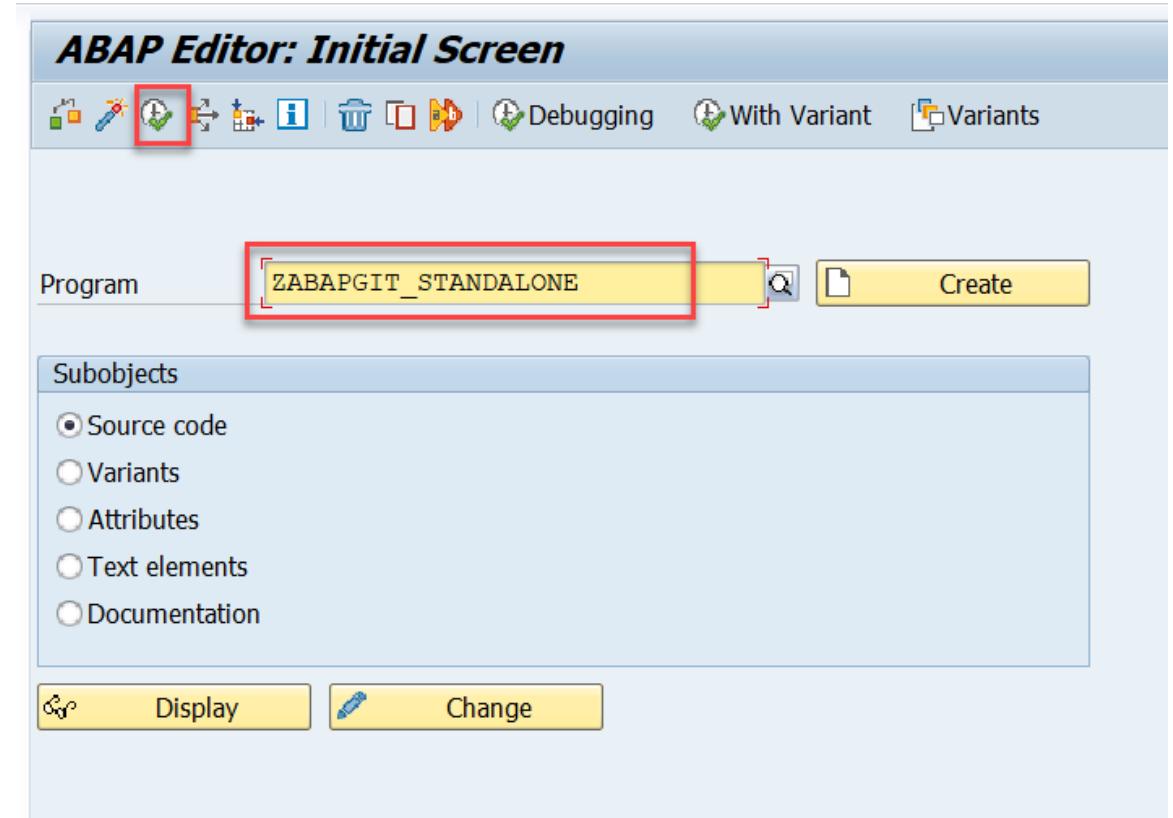
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `.$ZABAPGIT`).

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

## STEP 2: Download ABAP Code

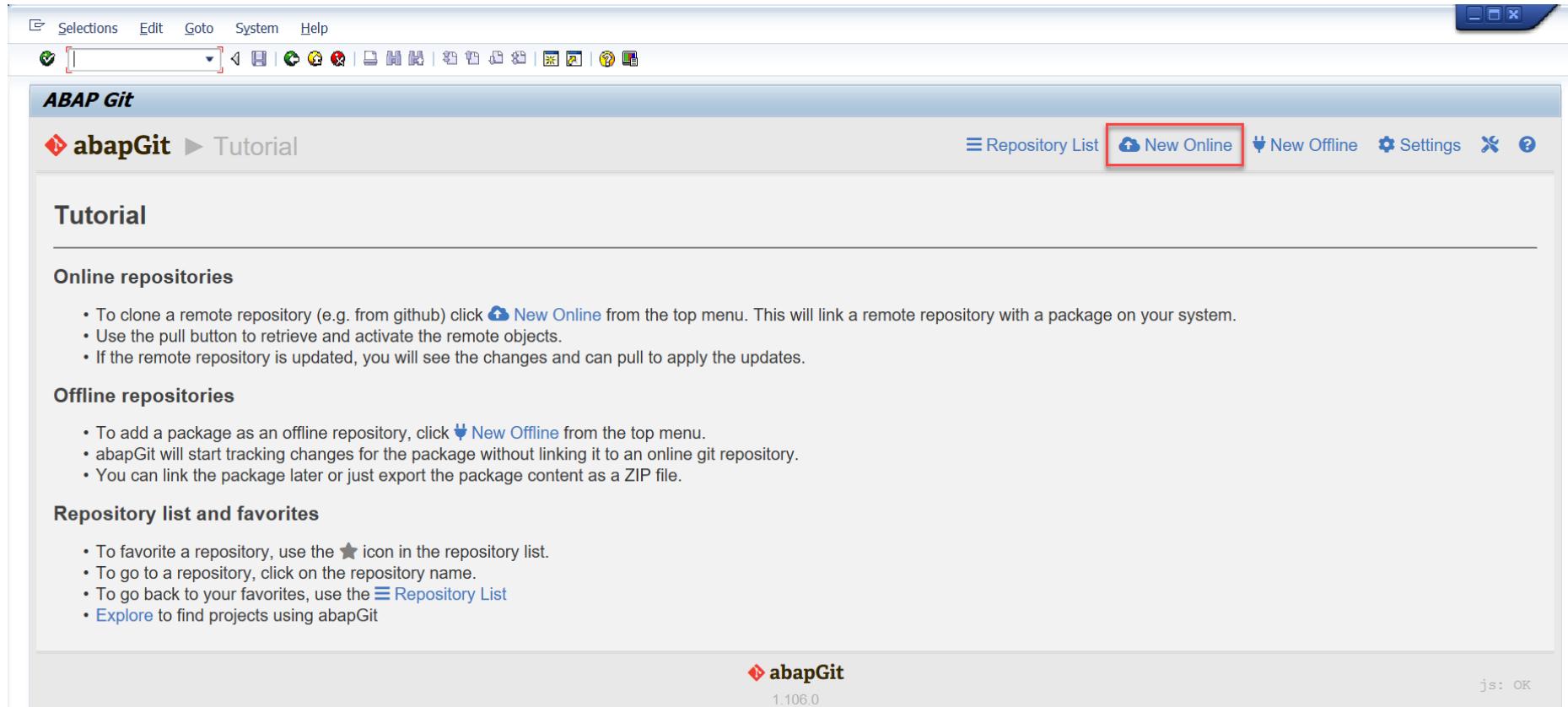
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,  
**ZABAPGIT\_STANDALONE**

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

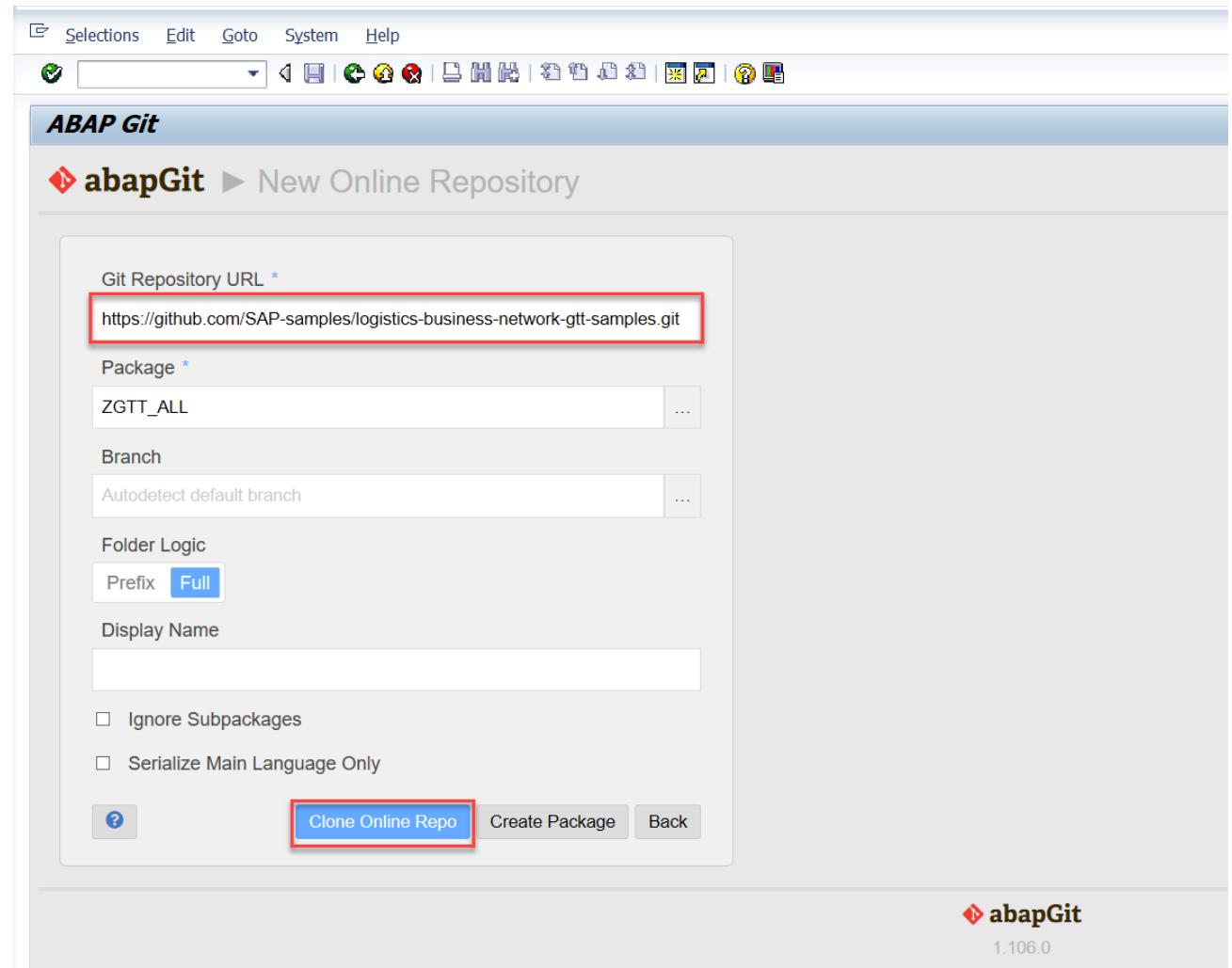
2-4: Fill in the **Git repository URL**:

<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.

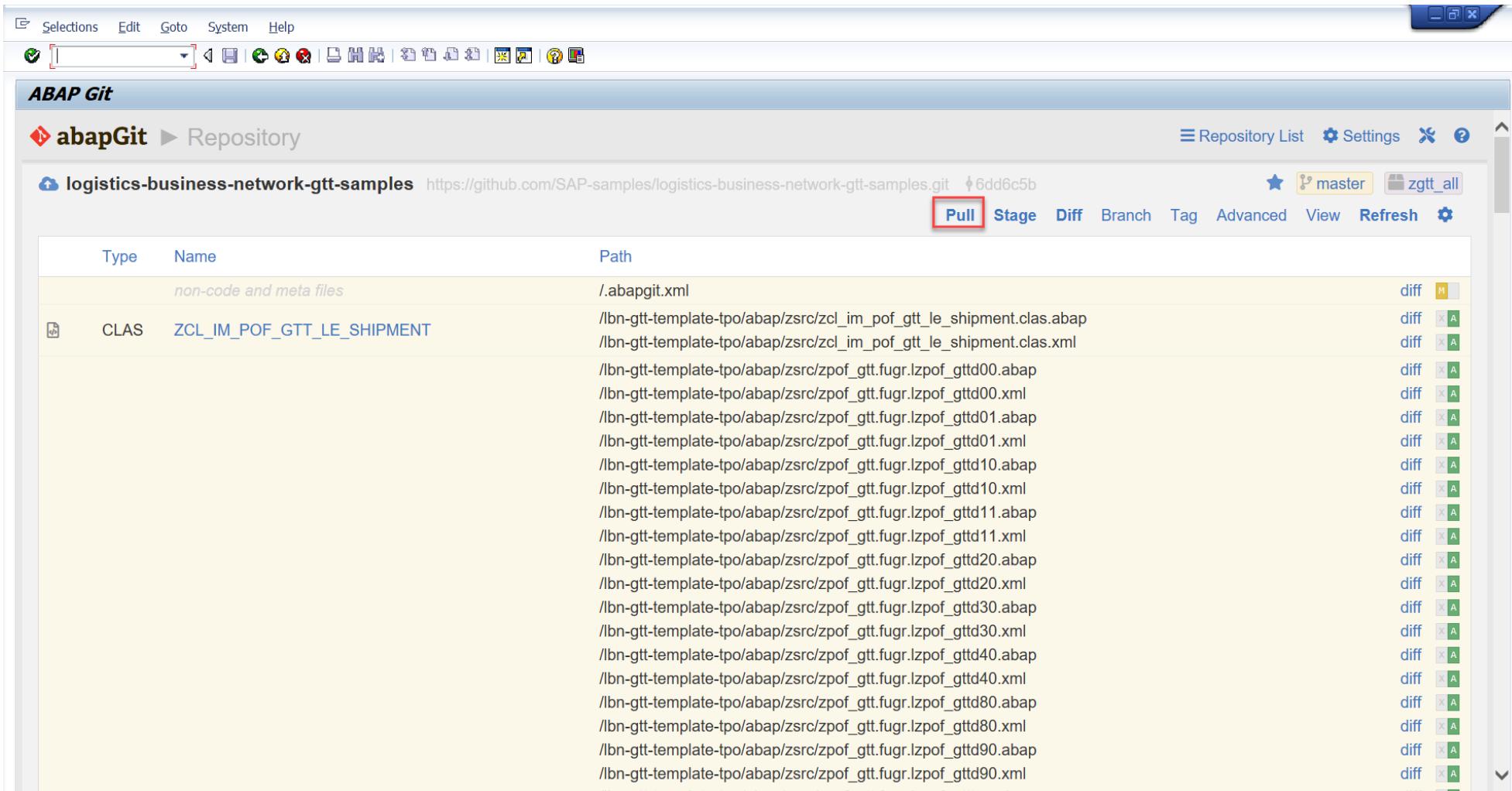
2-6: Set *Full* for **Folder Logic**

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



The screenshot shows the ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and the repository name "abapGit Repository". Underneath is another header with the repository URL "logistics-business-network-gtt-samples" and a commit hash "6dd6c5b". To the right of this are buttons for "master" and "zgtt\_all". The main area is a table with columns "Type", "Name", and "Path". The "Type" column shows mostly "non-code and meta files" and one entry for "CLAS". The "Name" column lists file names like ".abapgit.xml", "ZCL\_IM\_POF\_GTT\_LE\_SHIPMENT", and various ABAP and XML files under the path "/bn-gtt-template-tpo/abap/zsrc/". The "Path" column shows the full file paths. On the far right of each row are "diff" buttons with small colored squares indicating changes (yellow for M, green for A, grey for X).

Type	Name	Path	diff
	non-code and meta files	/.abapgit.xml	M
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/bn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.xml	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.abap	A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.xml	A

# Known Issue: Remote Deleted Object Cannot be Synchronized to the Local Object

**Symptom:** If the user update the ABAP code by report ZABAPGIT\_STANDALONE, there will be a code difference as below: because the enhancement implementation Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC is already obsolete and removed from the GitHub, the report ZABAPGIT\_STANDALONE cannot remove the object which was already deleted in GitHub.

The screenshot shows the SAP abapGit interface. The title bar says "GTT Installation". The main area displays a repository named "logistics-business-network-gtt-samples" with the URL "https://github.com/PALGTT/logistics-business-network-gtt-samples.git" and commit hash "cdeaaaf9". The repository is set to the "master" branch. A red box highlights the table showing differences between local and remote files. The table has columns "Type", "Name", and "Path". It lists three entries: a non-code file ".abapgit.xml", a class "ZCL\_GTT\_SOFTWARE\_SHIPPING" (type CLAS), and an enhancement "Z\_GTT\_SOFTWARE\_SHIPPING\_DELIVERY\_PROC" (type ENHO). The "Z\_GTT\_SOFTWARE\_SHIPPING\_DELIVERY\_PROC" entry is fully highlighted with a red border, indicating it is the focus of the issue described in the text above.

Type	Name	Path	diff	
non-code and meta files			/.abapgit.xml	diff
CLAS	ZCL_GTT_SOFTWARE_SHIPPING	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_im_le_shipping.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_im_le_shipping.clas.xml	diff diff	
ENHO	Z_GTT_SOFTWARE_SHIPPING_DELIVERY_PROC	/lbn-gtt-template-tso/abap/zsrc/z_gtt_sof_le_shp_delivery_proc.enho.xml	diff	

# **Known Issue: Remote Deleted Object Cannot be Synchronized to the Local Object**

## **Solution:**

### **Option 1)**

1-2) Deactivate the BADI implementation

### **Option 2)**

2-1) Deactivate the enhancement implementation Z\_GTT\_SOF\_LE\_SHIP\_DELIVERY\_PROC

2-2) Delete the BADI implementation class ZCL\_GTT\_SOF\_IM\_LE\_SHIPPING

## **Notes:**

Option 1: Objects deactivated and can be used after activation in the future.

Option 2: Objects deleted completely and would not be shown in the ABAPGit during code download.

# Known Issue: Remote Deleted Object Cannot be Synchronized to the Local Object

For option 1: Go to Transaction code SE19 and deactivate the BADI implementation.

The screenshot shows the SAP BAdI Builder interface. At the top, it says "BAdI Builder: Initial Screen for Implementations". Below that, there's a toolbar with "Check", "Delete implementation", "Copy implementation", "Rename implementation", "Application help", and "More". The main area is titled "Edit Implementation" and has two tabs: "New BAdI" (selected) and "Classic BAdI". Under "New BAdI", the "Enhancement implementation" field contains "Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC", which is highlighted with a red box. Under "Classic BAdI", the "Implementation" field is empty. At the bottom of this section are "Display" and "Change" buttons. In the background, a second window titled "Enhancement Implementation Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC Display" is open. It shows the "Properties" tab selected, with "Implementation Elements" sub-tab active. The "BAdI Implementations" table lists "Z\_GTT\_SOF\_IM\_LE\_SHIPPING" as the implementation, with "Implementing Class" checked. On the right side of the properties window, there are sections for "BAdI Implementation" (set to "Z\_GTT\_SOF\_IM\_LE\_SHIPPING"), "Description" (set to "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list"), and "Runtime Behavior". The "Implementation is active" checkbox is unchecked, and the status message "The implementation will not be called" is displayed.

# D) Configuration and Coding Guide

## - Advanced



# 1: Maintain AOT Type

When you are creating Application Object Type for one Business Process Type, make sure the AOT name must be as same as the name that is defined in the corresponding model in the Manage Models application in SAP Business Network Global Track and Trace Version 2.

The screenshot illustrates the process of maintaining Application Object Types (AOT) in SAP Business Network Global Track and Trace Version 2. It shows two main screens: the 'Define Application Object Types' screen and the 'Model Details' screen.

**Define Application Object Types Screen:**

- Bus. Proc. Type:** TMS\_TOR
- Appl. Obj. Type:** ZGTT\_SHP\_ACC\_HD (highlighted with a red box)
- Description:** Extract freight order header information to Global Track and Trace-Acc
- Text:** (empty)
- General Data:** Sequencing / Destination (Seq. No.: 10, CI for GTT: ZGTTISSTAC), Business Object Reference (Object Type: [empty], BO Setup Fnct: [empty]), Behavior (checkboxes: GTT Relevant, Stop AO Determ., Appl. Log Deact, Alt. BusProcType: [empty]).

**Model Details Screen:**

- SAP Model Details:** Internal - Test
- SOF:** Active
- Sales Order Fulfillment:** Namespace: com.lbngttsamples.gtt.app.sof Correlation Level: 5
- Tracked Process:** Shipment
- IDOC Integration:** Tracked Process Mapping (ERP Object Type: Others, Application Object Type: ZGTT\_SHP\_ACC\_HD highlighted with a red box)
- Fields:** (Table)

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPGP	YN_SHP_NO
serviceAgentLbnId	E1EHPGP	YN_SHP_SA_LBN_ID
transportationMode	E1EHPGP	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPGP	YN_SHP_CONTAIN_DGOODS

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

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

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

Function	Function Module	Description
ZSST_GTT_FO_HDR	ZSST_GTT_OTE_FO_HDR_REL	Extractor for relevance determination for Freight Order and Freight Booking

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.

### 3: Sample Codes for the Track Shipments Template App

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

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

# 4: Available Contexts for the Extractors Modules

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

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

4-3: Select the Business Process Types to find all the context tables and their structure info

The image consists of two side-by-side SAP screenshots. The left screenshot shows the 'Display IMG' interface with the 'Structure' tree expanded. A path is highlighted: 'Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface -> Define Business Process Types'. The 'Define Business Process Types' node is specifically highlighted with a red box. The right screenshot shows the 'Display View "Define Available Application Tables": Overview' screen. It has a 'Business Process Type' field set to 'HMS\_TOR'. Below it is a table titled 'Define Available Application Tables' with columns: Structure/Table, DDIC Definition, DB Struc. Name, Bus..., Updt... Fld Name, No C..., Inser..., Upda..., Delete..., Key St..., and Key Length. The table lists various context tables such as 'TOR\_PARTY', 'TOR\_REQ\_ROOT', 'TOR\_REQ\_ROOT\_BEFORE', 'TOR\_REQ\_STOP', 'TOR\_REQ\_STOP\_BEFORE', 'TOR\_REQ\_STOP\_TU', 'TOR\_REQ\_STOP\_TU\_BEFORE', 'TOR\_REQ\_STOP\_TU\_STOP', 'TOR\_REQ\_STOP\_TU\_STOP\_BEFORE', 'TOR\_ROOT', 'TOR\_ROOT\_BEFORE', 'TOR\_STOP', 'TOR\_STOP\_ADDR', 'TOR\_STOP\_BEFORE', 'TOR\_STOP\_SUCCESSOR', 'TOR\_STOP\_SUCCESSOR\_BEF', 'TOR\_TENDERING', 'TRQ\_ROOT', and 'TRQ\_ROOT\_BEFORE'. The first column 'Structure/Table' for all rows is highlighted with a red box.

Structure/Table	DDIC Definition	DB Struc. Name	Bus...	Updt... Fld Name	No C...	Inser...	Upda...	Delete...	Key St...	Key Length
TOR_PARTY	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_REQ_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...		/SCMTMS/S_EM_BO...					0	0
TOR_REQ_STOP_TU	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_STOP_TU_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_STOP_TU_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_REQ_STOP_TU_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...		/SCMTMS/S_EM_BO...					0	0
TOR_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...		CHANGE_MODE	C	U	D		70	70
TOR_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_ADDR	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_SUCCESSOR	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_SUCCESSOR_BEF	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_TENDERING	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TRQ_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TRQ_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70

# 5: Coding Tips in the GTT Relevance Function Modules

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

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

See sample code of function module: **ZSST\_GTT\_OTE\_FO\_HDR\_REL**

The screenshot displays two SAP ABAP development environments side-by-side.

**Left Window (Function Builder):**

- Title Bar:** Function Builder: Display ZSST\_GTT\_OTE\_FO\_HDR\_REL
- Toolbar:** Standard SAP tool icons.
- Repository Browser:** Shows the package ZSST\_GTT and lists various function modules like ZSST\_GTT\_EE\_FO\_ARRIVAL\_REL, ZSST\_GTT\_EE\_FO\_COUPLING, etc.
- Function Module Tab:** Active tab is ZSST\_GTT\_OTE\_FO\_HDR\_REL.
- Source Code View:** Displays the ABAP code for the function module, including declarations, imports, exports, and a TRY-CATCH block.

**Right Window (ABAP Editor):**

- Title Bar:** ABAP Editor: Display Include LZSST\_GTT\_D20
- Toolbar:** Standard SAP tool icons.
- Repository Browser:** Shows the package ZSST\_GTT and the include LZSST\_GTT\_D20.
- Source Code View:** Displays the ABAP code for the LZSST\_GTT\_D20 include module, specifically the method lif\_bo\_reader~check\_relevance.

# 6: Coding Tips in the Tracking ID Function Modules

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

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

See sample code of function module: *ZSST\_GTT\_OTE\_FO\_HEADER\_TID*. Main logic for Freight Order and Freight Booking Tracking ID: *LCL\_BO\_FREIGHT\_ORDER\_READER* and *LCL\_BO\_FREIGHT\_BOOKING\_READER*, method *LIF\_BO\_READER~GET\_TRACK\_ID\_DATA*

The image shows two SAP IDE screens side-by-side. The left screen is the 'Function Builder: Display ZSST\_GTT\_OTE\_FO\_HEADER\_TID' showing the source code for the function module. The right screen is the 'ABAP Editor: Display Include LCL\_BO\_FREIGHT\_D20' showing the source code for the include module. Both screens show parts of the code related to tracking ID handling, including imports, data definitions, and method implementations.

**Function Builder: Display ZSST\_GTT\_OTE\_FO\_HEADER\_TID**

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

TRY.
  lcl_ef_performer->get_track_id_data(
    EXPORTING
      is_definition = VALUE #(          maintab = lif_sst_constants->cs_tabledesc-fo_header_new )
      io_bo_factory = NEW lcl_bo_factory( )          i_appsyst
      iv_appsyst
      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
    IMPORTING
      et_track_id_data = e_trackidata[])

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

**ABAP Editor: Display Include LCL\_BO\_FREIGHT\_D20**

```
METHOD lif_bo_reader~get_track_id_data.

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

  FIELD-SYMBOLS: <lt_item> TYPE ANY_TABLE,
                  <lt_item_old> TYPE ANY_TABLE,
                  <ls_root> TYPE /smmtms/s_em_bo_tor_root,
                  <lt_root_new> TYPE /smmtms/t_em_bo_tor_root,
                  <lt_root_old> TYPE /smmtms/t_em_bo_tor_root.

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

  lr_root_new = mo_ef_parameters->get_appl_table(
    iv_tabledesc = lif_sst_constants->cs_tabledesc-fo_header_new ).

  lr_root_old = mo_ef_parameters->get_appl_table(
    iv_tabledesc = lif_sst_constants->cs_tabledesc-fo_header_old ).

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

# 7: Coding Tips in the Control Parameter Function Modules

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

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

See sample code of function module: *ZSST\_GTT\_OTE\_FO\_HDR*. Main logic for Freight Order and Freight Booking Control parameters: *LCL\_BO\_FREIGHT\_ORDER\_READER* and *LCL\_BO\_FREIGHT\_BOOKING\_READER*, method *LIF\_BO\_READER~GET\_DATA*

The screenshot shows the SAP Business Network Global Track and Trace v2 interface. At the top, there are tabs for 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below these, there is a 'Integration Switch' toggle switch set to 'ON'. The main area displays an 'Application Object Type: ZGTT\_SHP\_ACC\_HD'. A red box highlights a table titled 'Fields' under the 'IDOC Integration' tab. The table has columns for 'Field', 'IDOC Segment', and 'IDOC Field'. The data in the table is as follows:

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

# 7: Coding Tips in the Control Parameter Function Modules

**ABAP Editor: Display Include LZSST\_GTT\_D20**

```

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

```

**Function Builder: Display ZSST\_GTT\_OTE\_FO\_HDR**

```

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_sst_co-
      io_bo_factory     = NEW lcl_tor_factory( )
      iv_appsyst         = i_appsyst
      is_app_obj_types   = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_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_appsyst     = i_appsyst
      IMPORTING
        es_bapiret     = ls_bapiret ).

    APPEND ls_bapiret TO e_logtable.

    CASE lo_udm_message->textid.

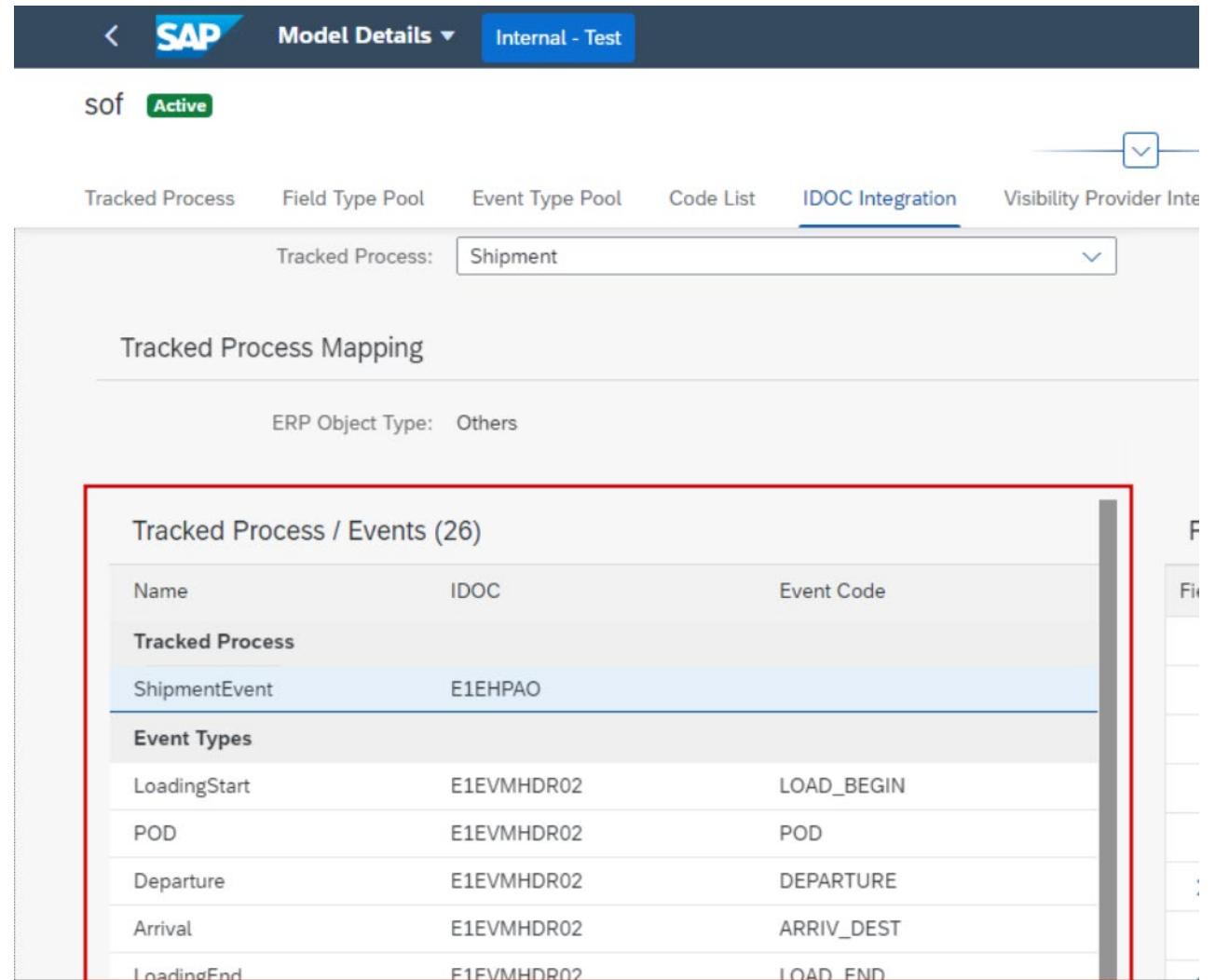
```

# 8: Coding Tips in the Planned Event Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E\_EXPEVENTDATA*.
3. SAP Business Network Global Track and Trace v2 asks for full transport for all the planned events, which means that all the events need 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 needs 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 needs to be filled with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* application in SAP Business Network Global Track and Trace V2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function module: *ZSST\_GTT\_EE\_FO\_HDR*, Main logic for Freight Order and Freight Booking Control parameters:  
*LCL\_PE\_FILLER\_FO\_HEADER*, method  
*LIF\_PE\_FILLER~GET\_PLANNED\_EVENTS*



The screenshot shows the SAP Model Details interface for a model named 'sof' (status: Active). The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process' dropdown is set to 'Shipment'. The 'Tracked Process / Events' section displays a table with 26 entries, each row containing a tracked process name, its IDOC, and the corresponding event code. The table is highlighted with a red border.

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

# 8: Coding Tips in the Planned Event Function Modules

**ABAP Editor: Display Include LZSST\_GTT\_D30**

```

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

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

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

  SHIFT lv_tor_id LEFT DELETING LEADING '0'.

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

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

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

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

```

Scope: \CLASS lc\_pe\_filler\_fo\_header\METHOD lif\_pe\_filler~get\_planned\_events | ABAP | Ln 581 Col 67

**Function Builder: Display ZSST\_GTT\_EE\_FO\_HDR**

```

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

CLEAR e_logtable[].

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

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

```

Scope: \FUNCTION ZSST\_GTT\_EE\_FO\_HDR\TRY | ABAP | Ln 37 Col 19

# 8: Coding Tips in the Planned Event Function Modules

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

The screenshot shows the SAP ABAP code editor with the title bar "LZSST\_GTTD10" and status bar "Active". The code is part of an include module and defines a method `get_postal_address`. The code uses several data objects and associations between objects like `lo_tor_srv_mgr`, `lo_loc_srv_mgr`, `lr_bo_conf`, and `lv_dummy`. The logic involves multiple IF statements to handle different key links and configurations. The code is color-coded for syntax highlighting.

```
Include LZSST_GTTD10 Active

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

# 9: Coding Tips in the Event Data Function Modules

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

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

See sample code of function module: *ZSST\_GTT\_EE\_FO\_ARRIVAL*.  
Relevance function module: *ZSST\_GTT\_EE\_FO\_ARRIVAL\_REL*.

The screenshot shows the SAP Model Details interface for a tracked process named 'Shipment'. The 'Tracked Process' tab is selected, showing a mapping table for 'ShipmentEvent'. The table has three columns: Name, IDOC, and Event Code. It lists five entries under 'Event Types': LoadingStart (IDOC: E1EVMHDR02, Event Code: LOAD\_BEGIN), POD (IDOC: E1EVMHDR02, Event Code: POD), Departure (IDOC: E1EVMHDR02, Event Code: DEPARTURE), Arrival (IDOC: E1EVMHDR02, Event Code: ARRIV\_DEST), and LoadingEnd (IDOC: E1EVMHDR02, Event Code: LOAD\_END). A red box highlights the entire table.

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

# 9: Coding Tips in the Event Data Function Modules

**Function Builder: Display ZSST\_GTT\_EE\_FO\_ARRIVAL**

```

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

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

```

**Function Builder: Display ZSST\_GTT\_EE\_FO\_ARRIVAL\_REL**

```

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

1  FUNCTION zsst_gtt_ee_fo_arrival_rel.
2
3      *"-> Local Interface:
4      IMPORTING
5          i_all_appl_tables TYPE /SAPTRX/APPLSYSTEM
6          i_event_code        TYPE /SCMTMS/IF_TOR_CONST=>SC_TOR_EVENT-ARRIV_DEST
7          i_event              TYPE /SCMTMS/IF_TOR_CONST=>SC_TOR_EVENT-ARRIV_DEST
8          i_relevance          TYPE /SCMTMS/IF_TOR_CONST=>SC_TOR_EVENT-RELEVANCE_DETERM_ERROR
9
10     EXPORTING
11         e_result            TYPE SY-BINPT
12
13     TABLES
14         c_logtable_structure BAPIRET2 OPTIONAL
15
16     EXCEPTIONS
17         parameter_error
18         relevance_determine_error
19         stop_processing
20
21     TRY.
22         lcl_actual_event->get_tor_actual_event_class( i_event )->check_event_relevance(
23             EXPORTING
24                 i_all_appl_tables = i_all_appl_tables
25                 iv_event_code    = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
26                 i_event           = i_event
27             IMPORTING
28                 e_result         = e_result .
29             CATCH cx_udm_message INTO DATA(lo_udm_message).
30
31     Scope: \FUNCTION zsst_gtt_ee_fo_arrival_rel\ TRY ABAP Ln 27 Col 52

```

# Thank you.

Contact information:

**Eva Hu**  
Product Management  
[e.hu@sap.com](mailto:e.hu@sap.com)



# Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

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, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.