



# 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](#) chapter of *How to Send Documents from SAP ERP to SAP Business Network Global Track and Trace*. You can find this guide at <http://help.sap.com/gtt>.

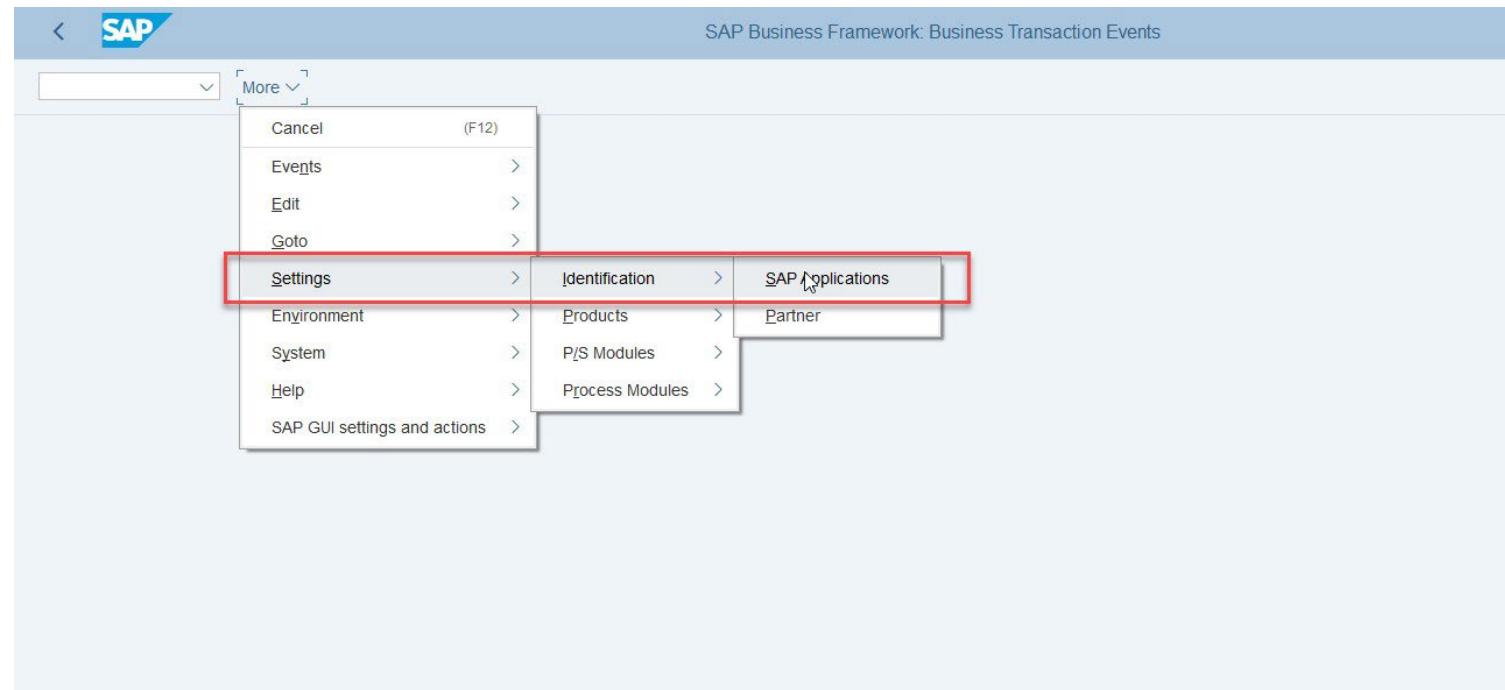
1-2: The ABAP codes on Github to support sample apps for SAP Business Network Global Track and Trace shall be implemented in SAP S/4HANA 1909 SP03 on premise or higher. Please note that the codes are not validated in its lower version or other ECC series of products, so you might need to do further adaptation 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 Fiori application interface titled 'Change View "BTE Application Indicator": Overview'. The main area is a table with two columns: 'Appl.' and 'Text'. The 'Appl.' column lists various application codes, and the 'Text' column provides a brief description of each. The row for 'PI-EM' is highlighted with a red border, and a checkmark is visible in the checkbox column next to it, indicating it is active. The table also contains other rows for applications like PM, PM-BW, PM-EQM, etc. At the bottom right of the table, there are 'Save' and 'Cancel' buttons.

Appl.	A	Text
PI-EM	<input checked="" type="checkbox"/>	SAP Event Manager Integration
PM	<input checked="" type="checkbox"/>	Instandhaltung
PM-BW	<input checked="" type="checkbox"/>	Instandhaltung-BW
PM-EQM	<input checked="" type="checkbox"/>	Instandhaltung, Equipment
PM-PAM	<input checked="" type="checkbox"/>	Instandhalt. Pool Asset Mgmt
PMA-PC	<input checked="" type="checkbox"/>	Product Compliance
PMAT	<input checked="" type="checkbox"/>	Produkt - Material
PMIPUR	<input type="checkbox"/>	PMI Anschluss Einkauf
PPUSH	<input type="checkbox"/>	MAM Push
PP-BD	<input checked="" type="checkbox"/>	Production Planning MasterData
PP-DD	<input checked="" type="checkbox"/>	Demand Driven Replenishment
PP-MRP	<input checked="" type="checkbox"/>	Material Requirements Planning
PRICAT	<input type="checkbox"/>	Preiskatalog
PS-REP	<input checked="" type="checkbox"/>	Projektsystem
PSRV	<input checked="" type="checkbox"/>	Produkt - Service
QBEXT	<input checked="" type="checkbox"/>	External Inspection Procurement
QBEXTP	<input checked="" type="checkbox"/>	External Inspection Production
QILPO	<input checked="" type="checkbox"/>	Inspection Lot Order Integr.
RDSVFI	<input type="checkbox"/>	Dgtl.Signature Validation FI
RDSVMD	<input checked="" type="checkbox"/>	Dgtl.Signature BP Check

# B) Configuration and Implementation

## - Basic

### B1. IDOC Configuration



# STEP 1: Define RFC Connection for 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

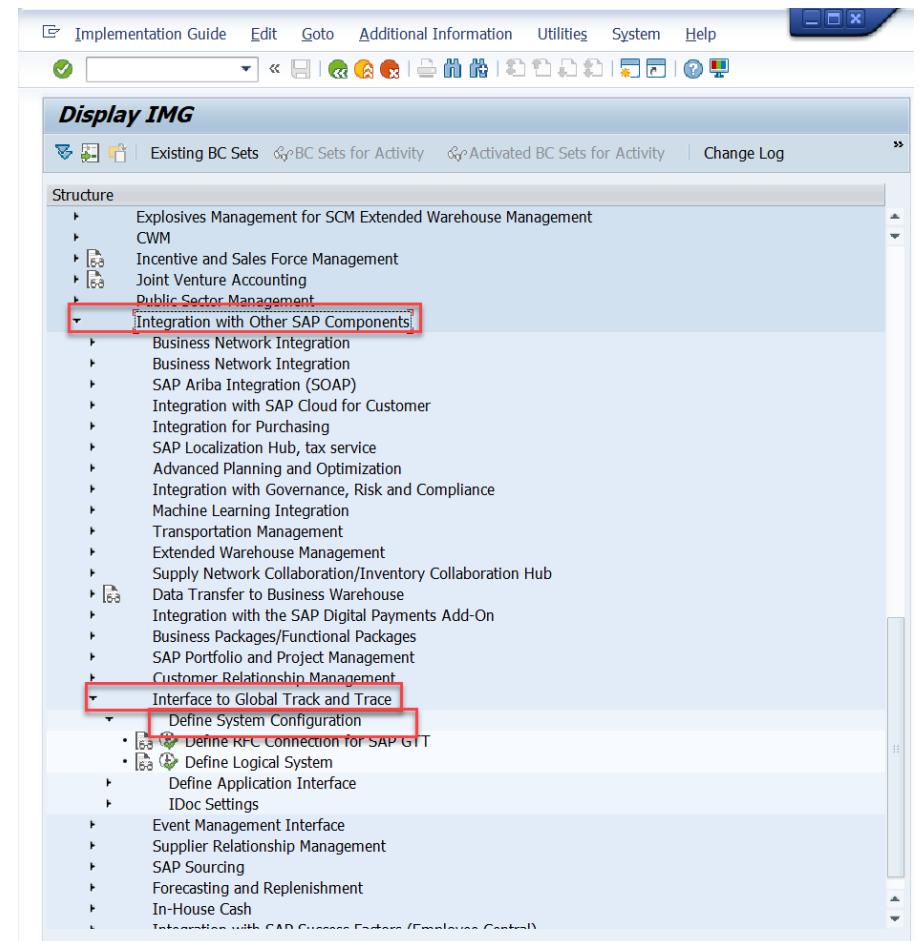
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

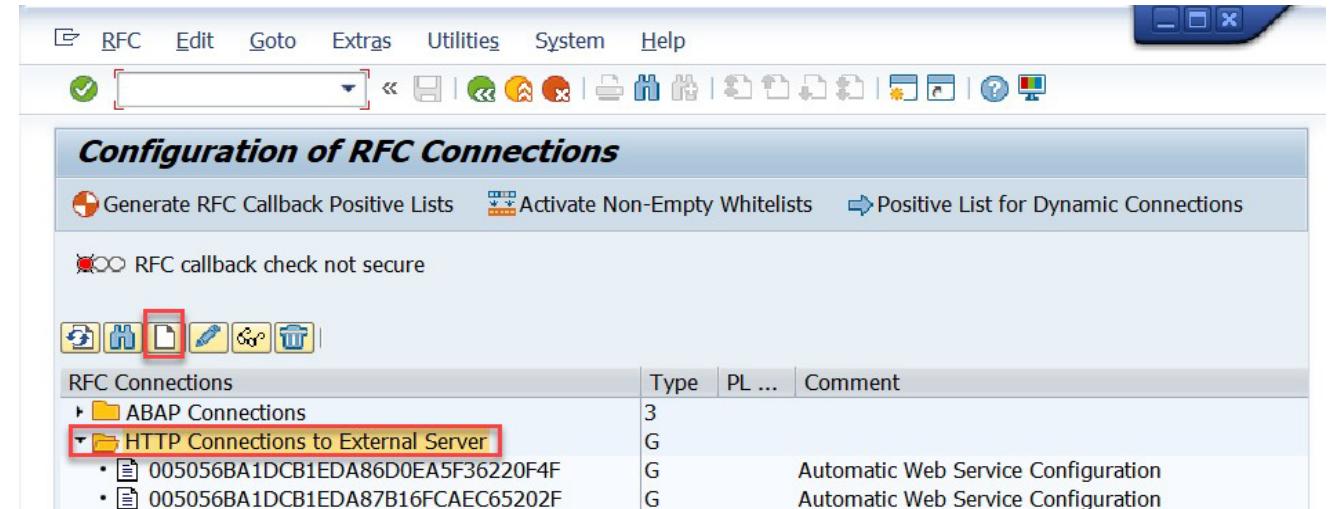
**Define RFC Connection for SAP GTT**



# 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	
Description 1	RFC for Events of SST Sample Application to Acceptance
Description 2	
Description 3	

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.

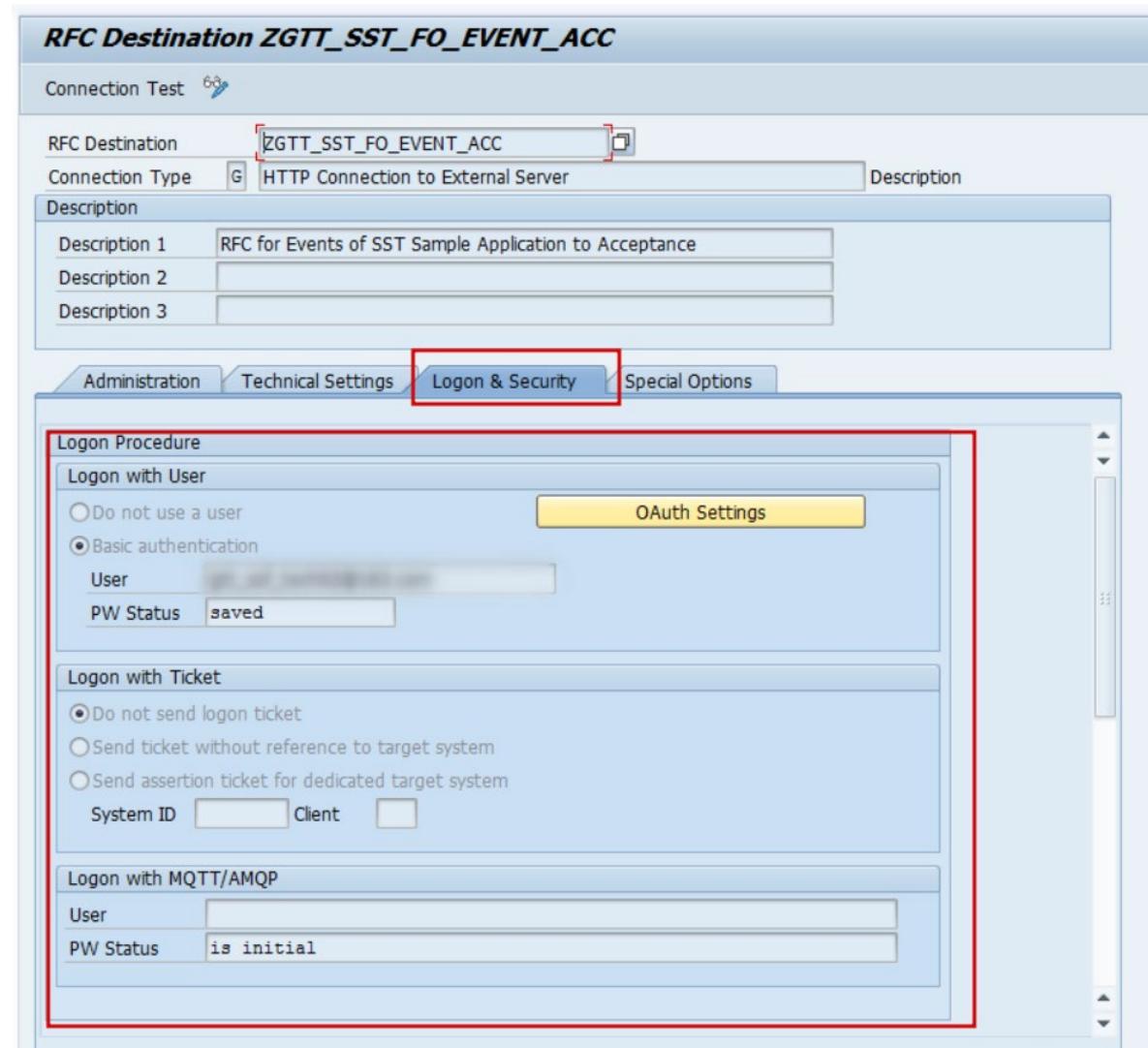
Also, SSL must be *Active*.

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

1-10: Save the configuration

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

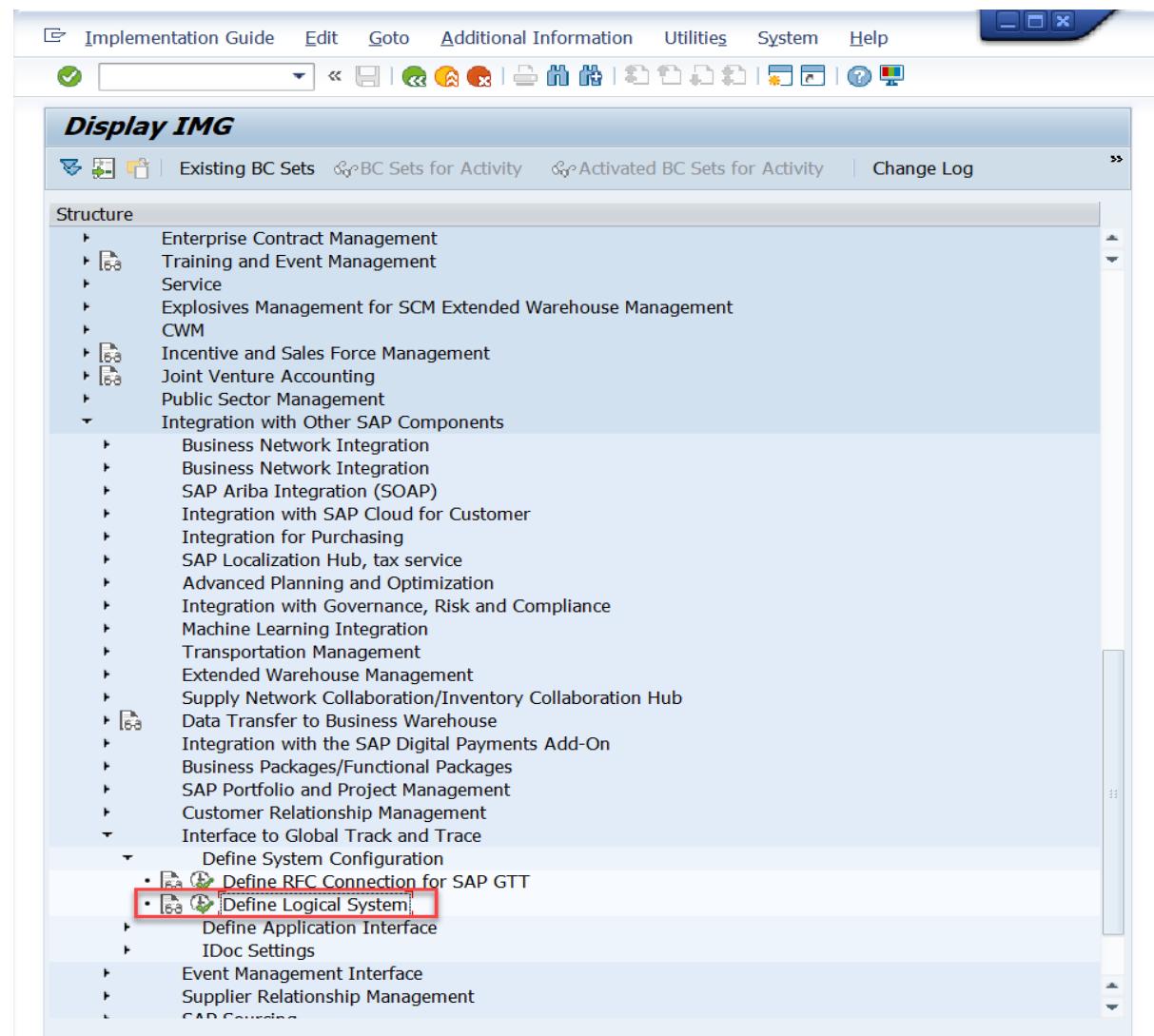
- one for event and
- the other for tracked process.



# 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

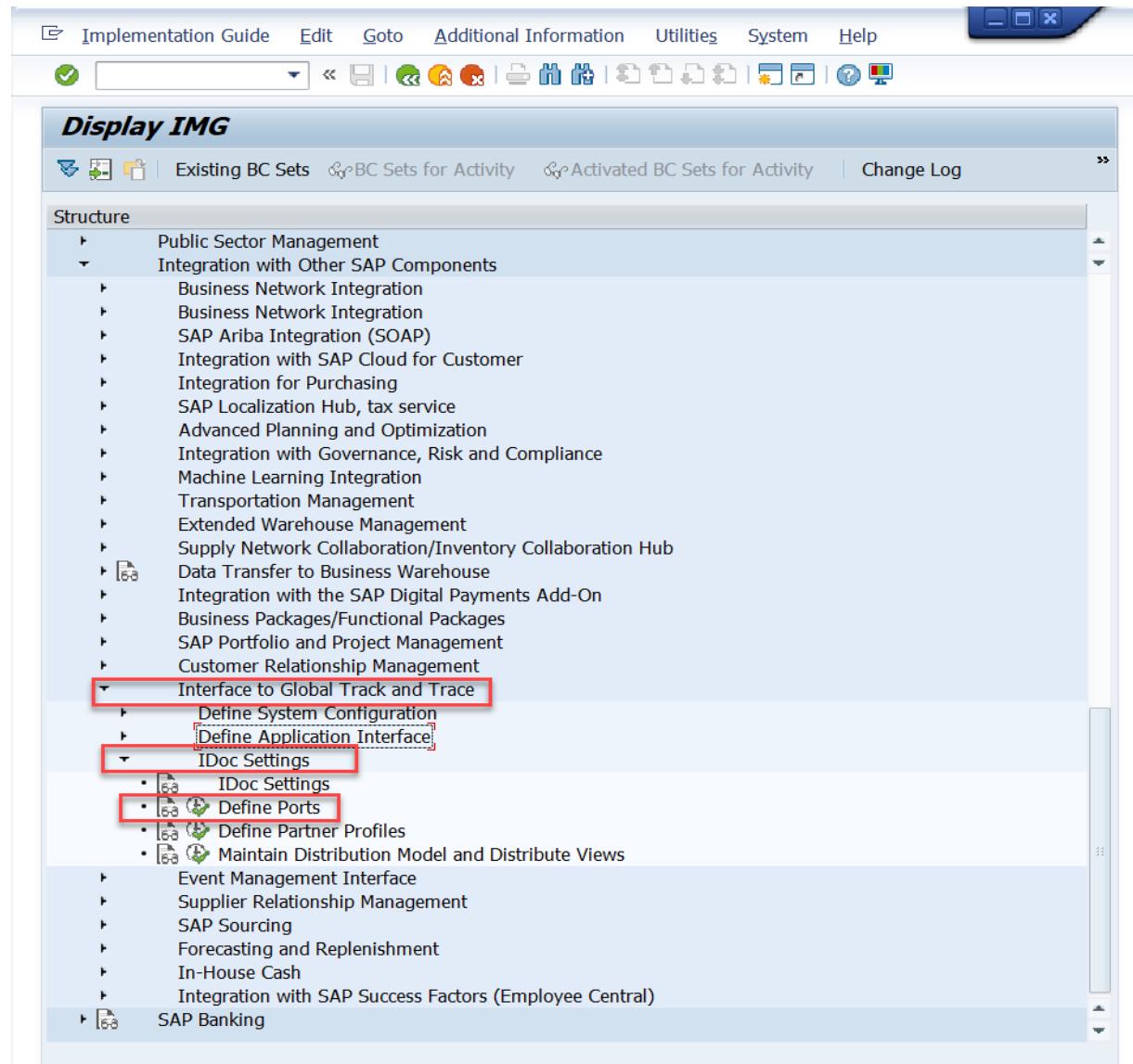
2-4: Save the configuration

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

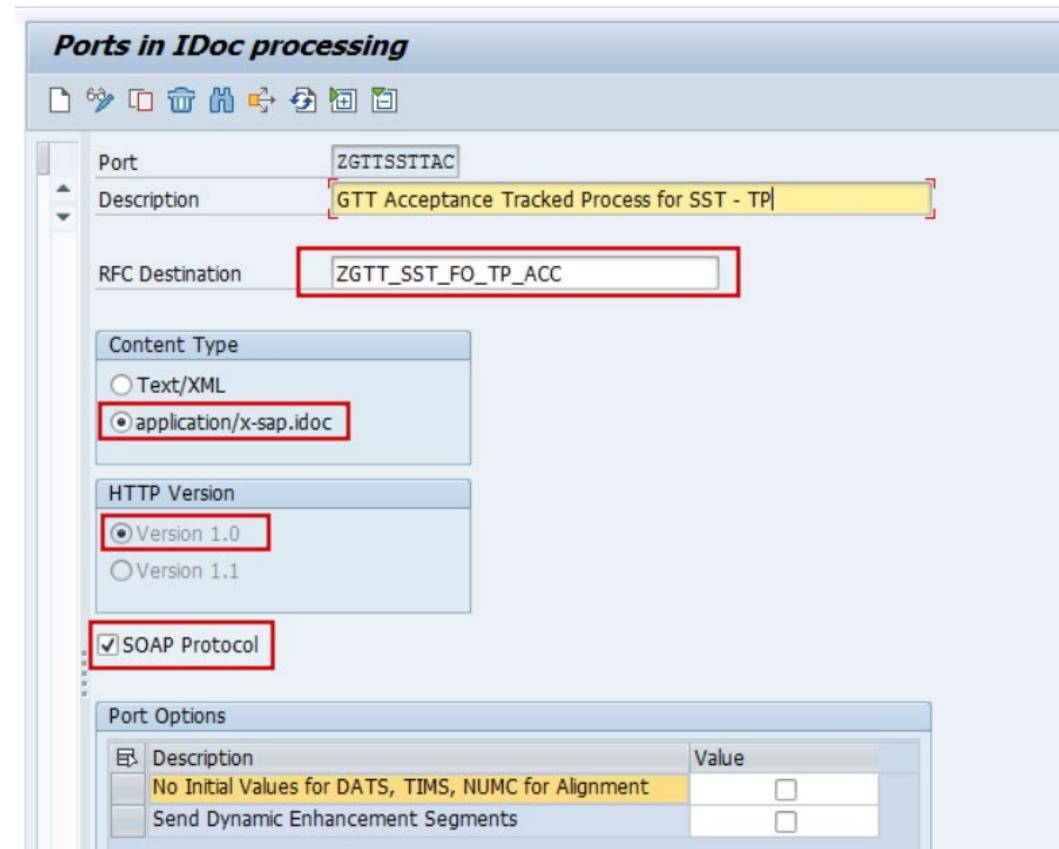
3-5: Choose **Content Type** as *application/x-sap.idoc*

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.

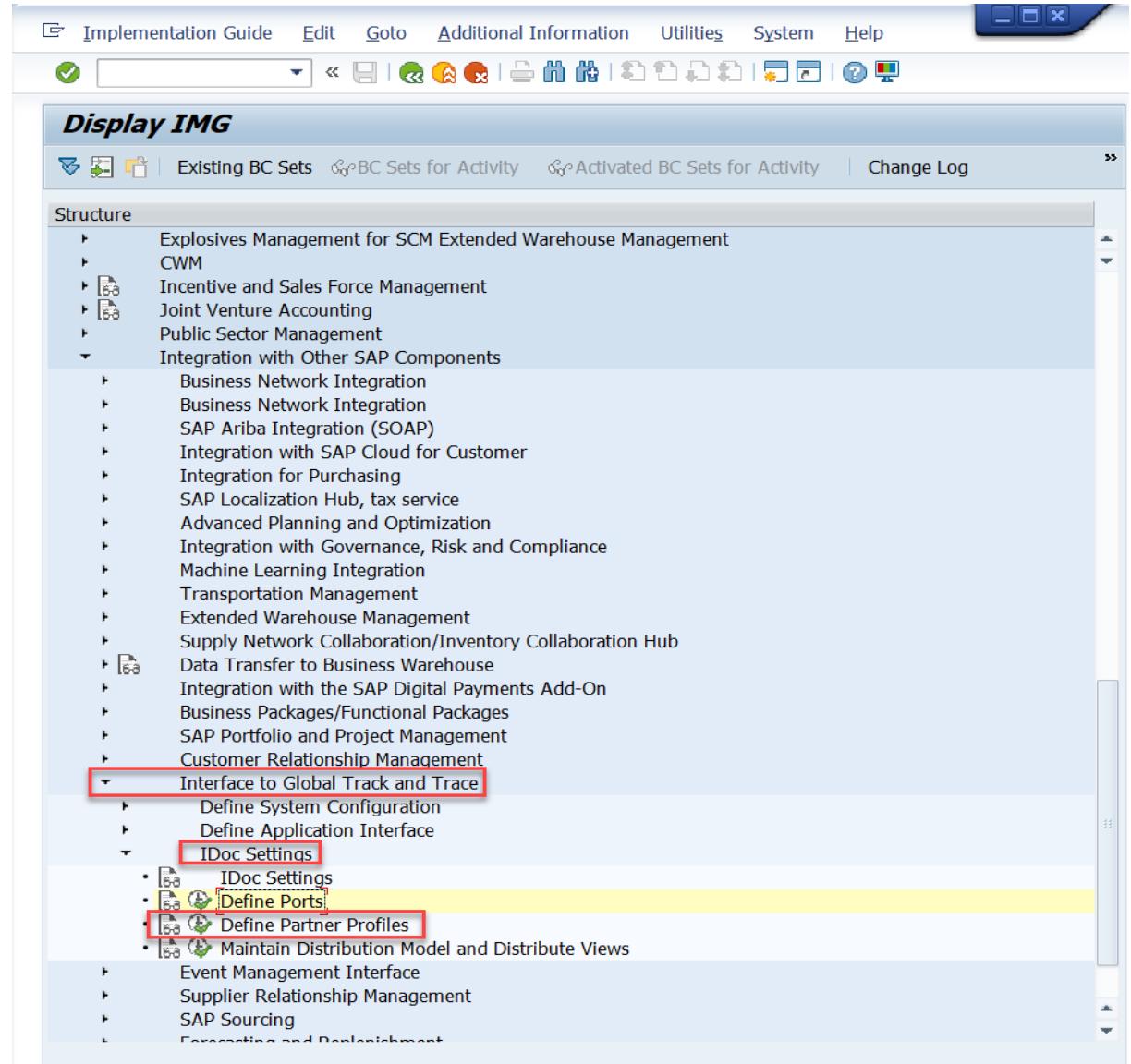


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSSSTEAC	GTT Acceptance Event for SST	ZGTT_SST_FO_EVENT_ACC	application/x-sap.idoc	Version 1.0	Checked
ZGTTSSTTAC	GTT Acceptance Tracked Process for SST	ZGTT_SST_FO_TP_ACC	application/x-sap.idoc	Version 1.0	Checked

# STEP 4: Define Partner Profiles

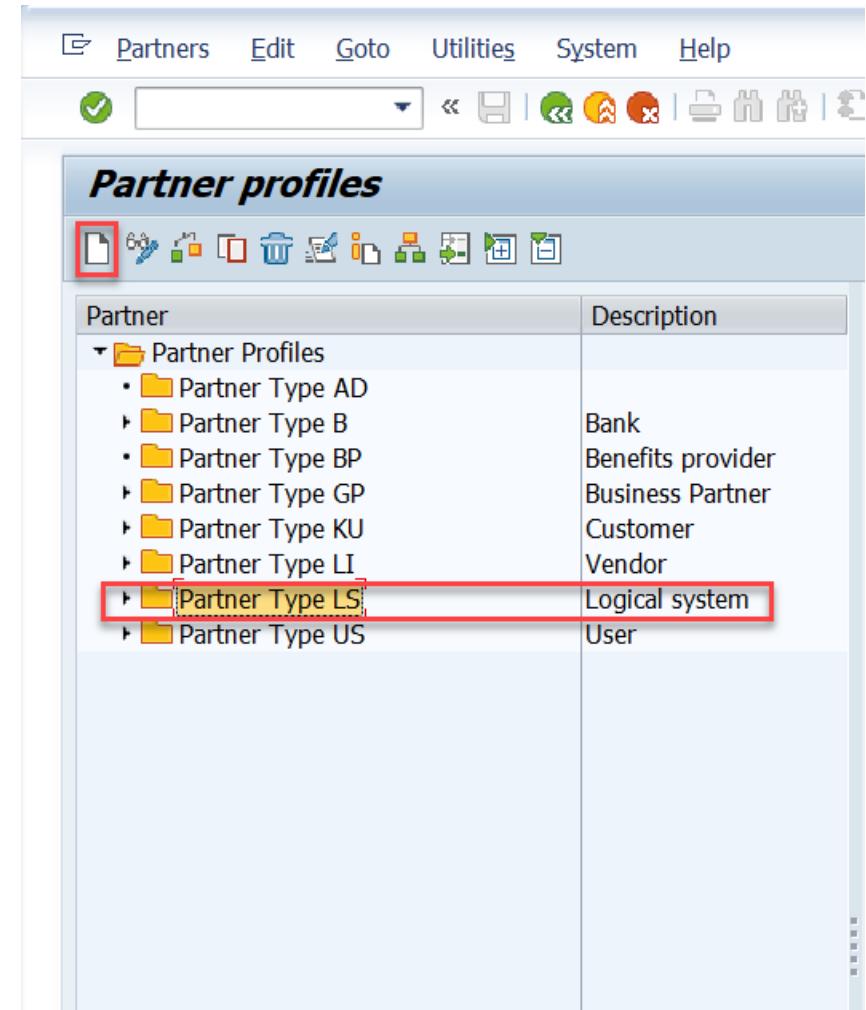
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -  
> **Interface to Global Track and Trace** -  
> **IDoc Settings**

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 Partner profiles interface. The top section displays basic partner information: Partner No. ZGTTSSSTAC (Logical System For GTT SST - Accept) and Type LS (Logical system). The 'Processor' field is highlighted with a red box, indicating it is the current step. Below this, the 'Ty.' field is set to US and the 'Lang.' field is set to EN. The 'Processor' field contains the value 'User'. The 'Outbound' section lists two message types: AOPOST and EVMSTA, each associated with a receiver partner (ZGTTSSSTAC and ZGTTSTEAC respectively) and a basic type (EHPOST01 and EVMSTA02). The 'Inbound' section is currently empty.

## 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 partner details are displayed: Partner No. ZGTTSSITAC, Type LS, Logical System For GTT SST - Accept, Logical system. There are tabs for Post Processing: Valid Processors, Classification, and Telephony. Under the Post Processing tab, there are fields for Ty. (US), Processor (User), and Lang. (EN). The English language is selected.

In the main area, there are two tables: Outbound and Inbound. The Outbound table has columns: Partner Role, Message Type, Message Va..., Function, Test, Receiver P..., I... Pa..., and Basic Type. It contains rows for AOPOST and EVMSTA, each associated with a receiver partner (ZGTTSSITAC or ZGTTSSSTEAC) and basic types EHPOST01 or EVMSTA02. The Inbound table has columns: Partner Role, Message Type, Message Va..., Function, Test, P.., and Process Code. It currently has no data.

At the bottom of the interface, there are several icons for search, add, and other functions. The 'Add' icon 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		
<input type="checkbox"/> 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

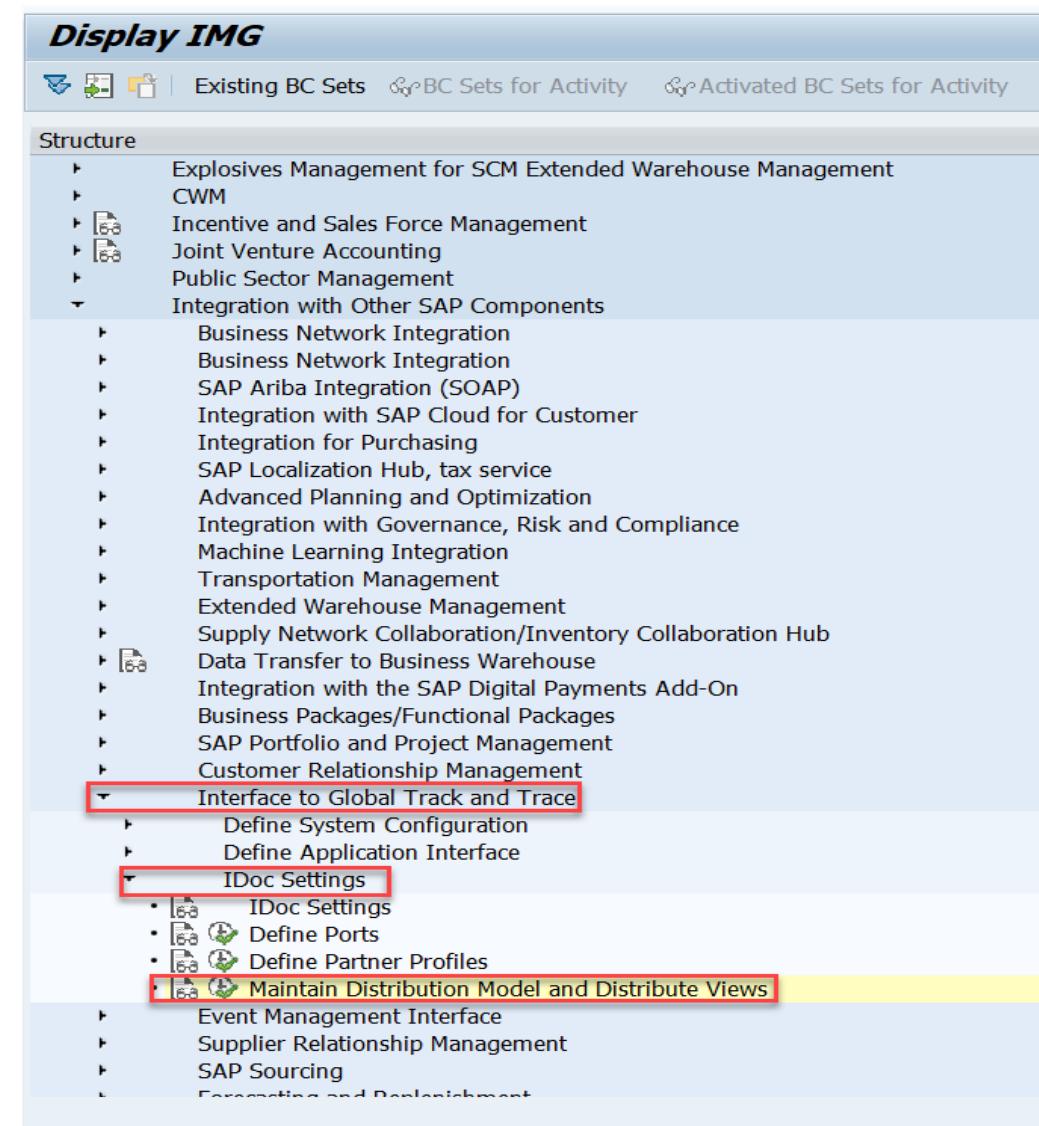
**Partner profiles: Outbound parameters**

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		
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	<input type="checkbox"/>	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.

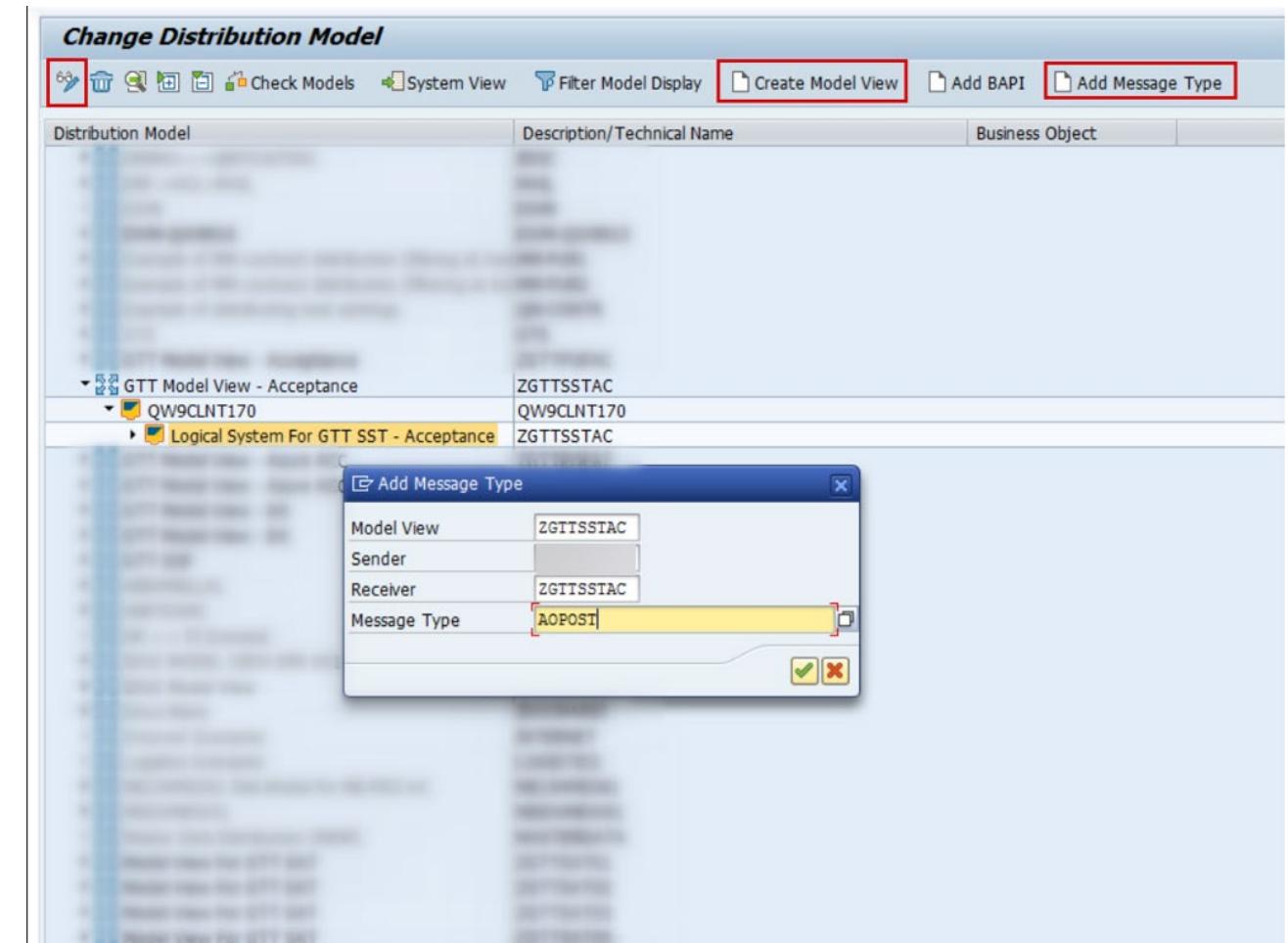
For the event:

**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**

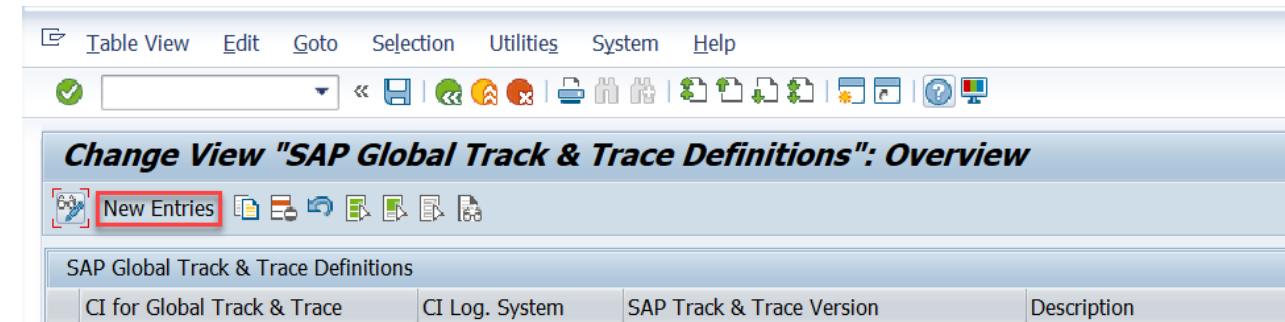
- 6-2: Choose activity  
**Define CI Tenant for SAP GTT**



# 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.



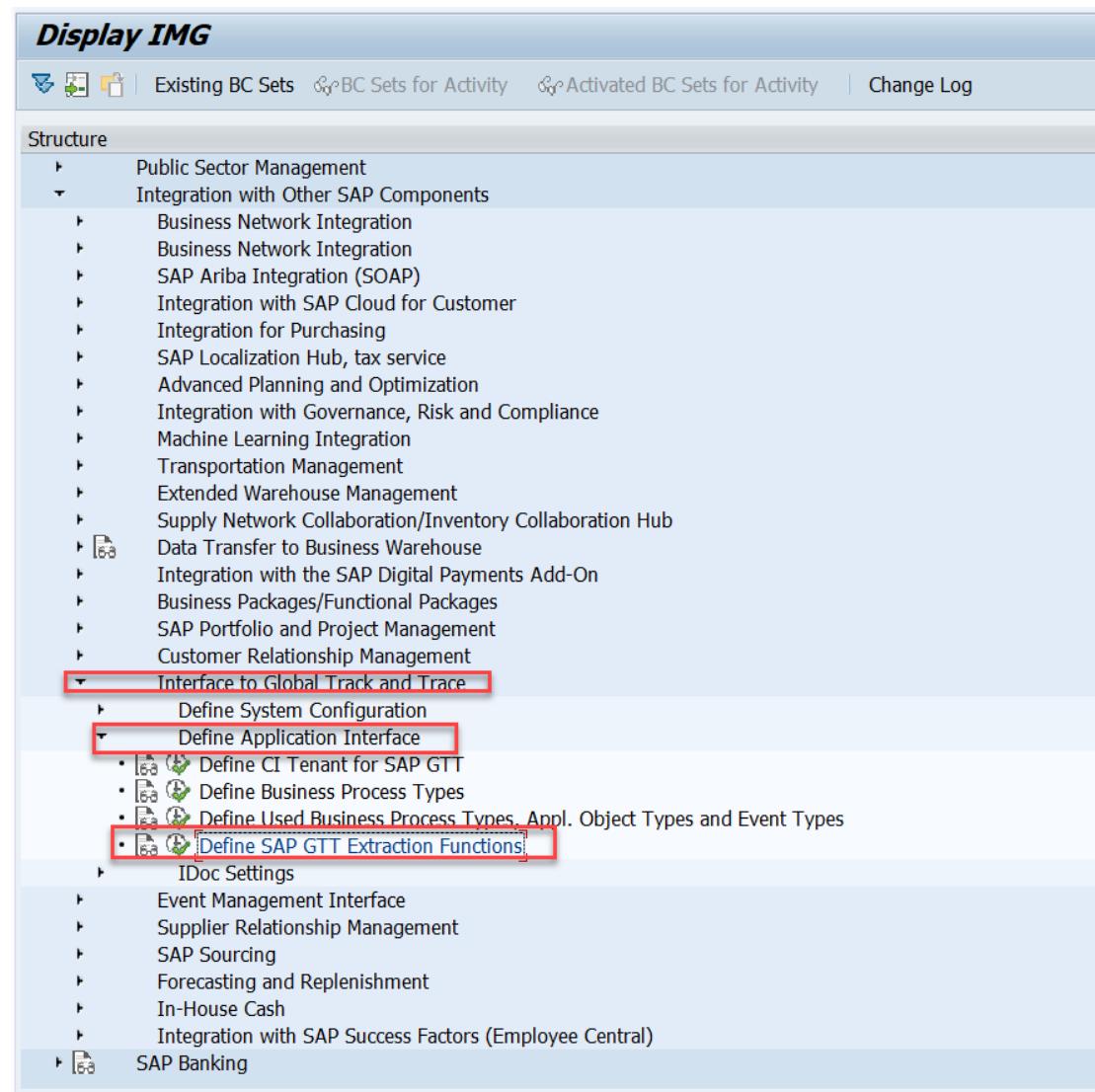
The screenshot shows the SAP Global Track & Trace Definitions table view. At the top left, there's a toolbar with icons for New Entries, Copy, Paste, etc. The main area is titled "Change View 'SAP Global Track & Trace Definitions': Overview". Below it, the table has columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. One row is visible, showing "ZGTTSSSTAC" in the CI for Global Track & Trace column, "ZGTTSSSTAC" in the CI Log. System column, "Global Track & Trace" in the SAP Track & Trace Version column, and "CI For GTT Freight Order Sample APP - Acceptance" in the Description column.

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

# STEP 7: Define GTT Extraction Functions

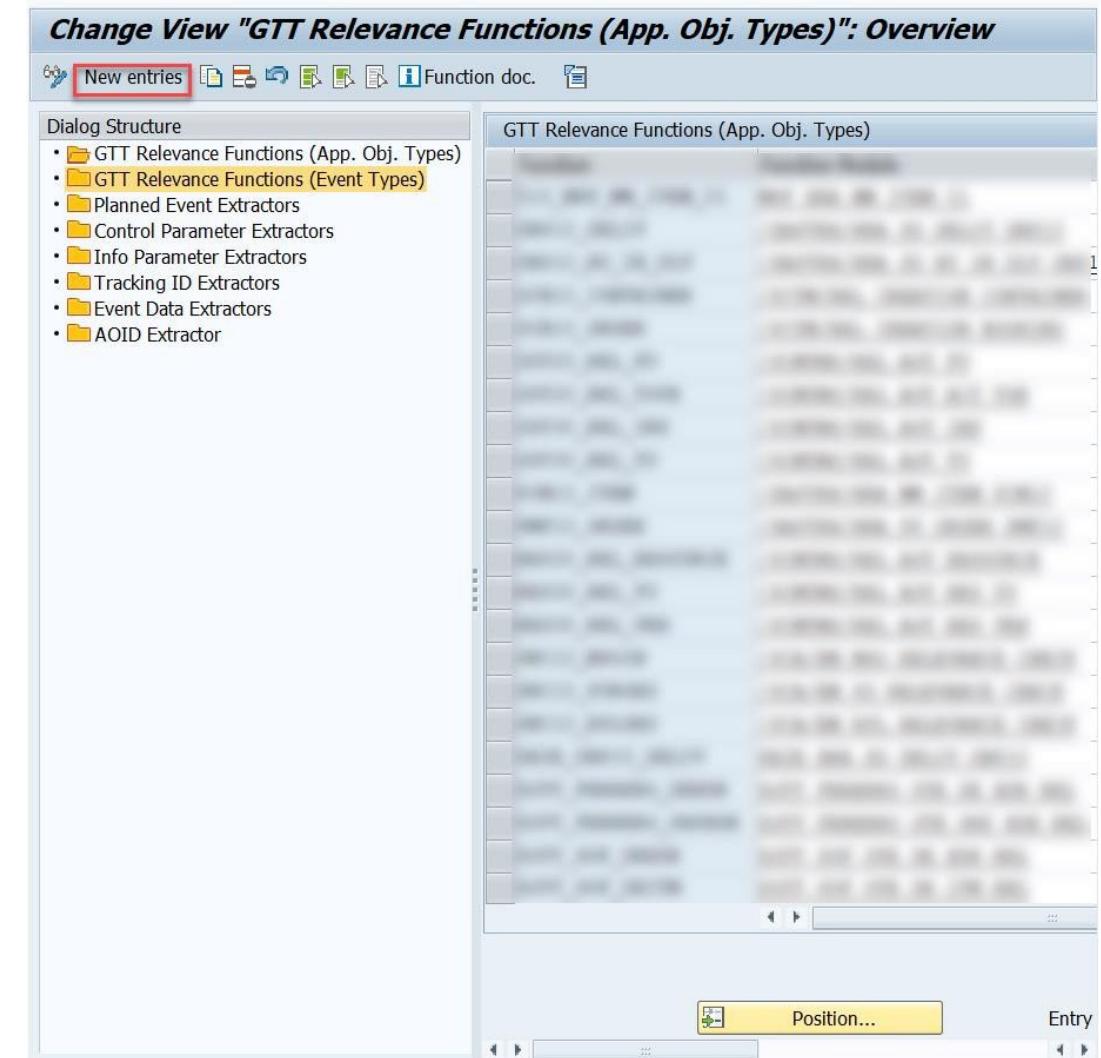
7-1: In **Display IMG** page, click  
**Integration with Other SAP Components -**  
-> **Interface to Global Track and Trace -**  
-> **Define Application Interface**

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



# STEP 7: Define GTT Extraction Functions

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



# STEP 7: Define GTT Extraction Functions

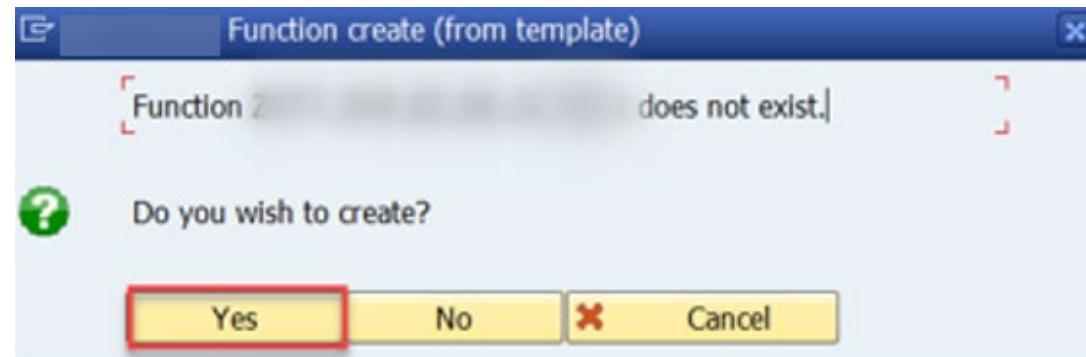
7-4: Input the **Function name** and **Function Module** for the newly created extraction function

7-5: Click **Save**

Change View "GTT Relevance Functions (App. Obj. Types)": Overview		
Dialog Structure	GTT Relevance Functions (App. Obj. Types)	
	Function	Function Module
<ul style="list-style-type: none"><li>•  GTT Relevance Function</li><li>•  GTT Relevance Function</li><li>•  Planned Event Extractor</li><li>•  Control Parameter Extractor</li><li>•  Info Parameter Extractor</li><li>•  Tracking ID Extractors</li><li>•  Event Data Extractors</li><li>•  AOID Extractor</li></ul>	ZSST_GTT_FO_HDR	ZSST_GTT_OTE_FO_HDR_REL
		Appl. Object Type Relevance for Freight Order Header

## STEP 7: Define GTT Extraction Functions

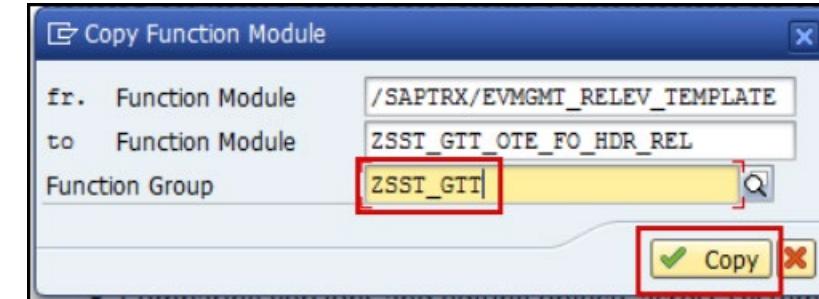
7-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



# 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.

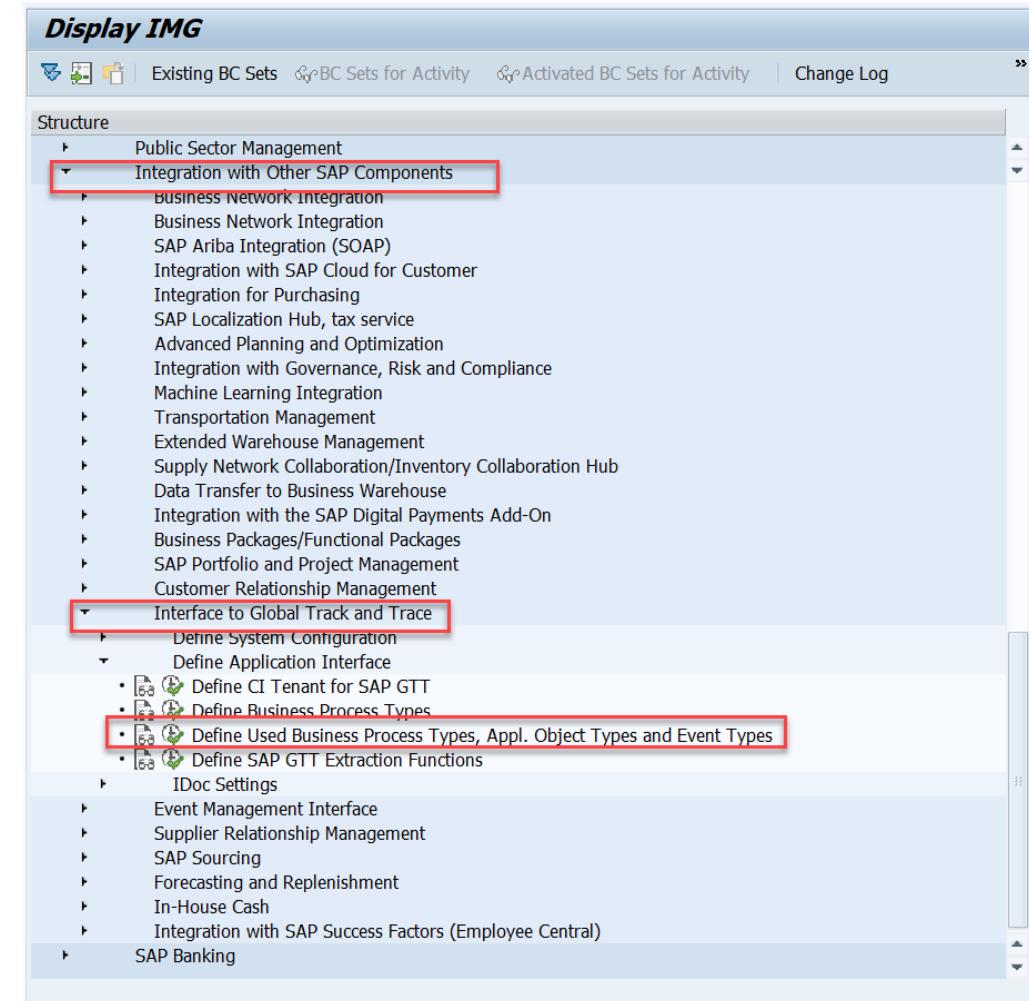
The screenshot shows the SAP Function Builder interface. In the Repository Browser, the 'Function Group' dropdown is set to 'ZSST\_GTT'. The 'Object Name' tree view shows several function modules under 'ZSST\_GTT', including 'ZSST\_GTT\_OTE\_FO\_HDR\_REL', which is highlighted with a red box. The main workspace displays the ABAP source code for the selected function module:

```
FUNCTION ZSST_GTT_OTE_FO_HDR_REL.  
  *--> Local Interface:  
  *-->  IMPORTING  
  *-->    REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM  
  *-->    REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/ACTYPES  
  *-->    REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER  
  *-->    REFERENCE(I_APPTYPE_TAB) TYPE TRXAS_APPTYPE_TABS_WA  
  *-->    REFERENCE(I_APP_OBJECT) TYPE TRXAS_APPOBJ_CTAB_WA  
  *--> EXPORTING  
  *-->    VALUE(E_RESULT) LIKE SY-BINPT  
  *--> TABLES  
  *-->    C LOGTABLE STRUCTURE BAPIRET2 OPTIONAL  
  *--> EXCEPTIONS  
  *-->    PARAMETER_ERROR  
  *-->    RELEVANCE_DETERM_ERROR  
  *-->    STOP_PROCESSING  
  *-->  
  DATA: lt_app_objects TYPE trxas_appobj_ctabs,  
        lo_udm_message TYPE REF TO cx_udm_message,  
        ls_bapiret TYPE bapiret2.  
  lt_app_objects = VALUE #( ( i_app_object ) ).  
  TRY.  
    e_result = lcl_ef_performer->check_relevance(  
      is_definition = VALUE #( maintab = lif_sst_constants->cs_tabledef-fo_header_new )  
      io_bo_factory = NEW lcl_tor_factory( )  
      iv_absvs = i_absvs  
    ).  
  Scope: \FUNCTION ZSST_GTT_OTE_FO_HDR_REL
```

# 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

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

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

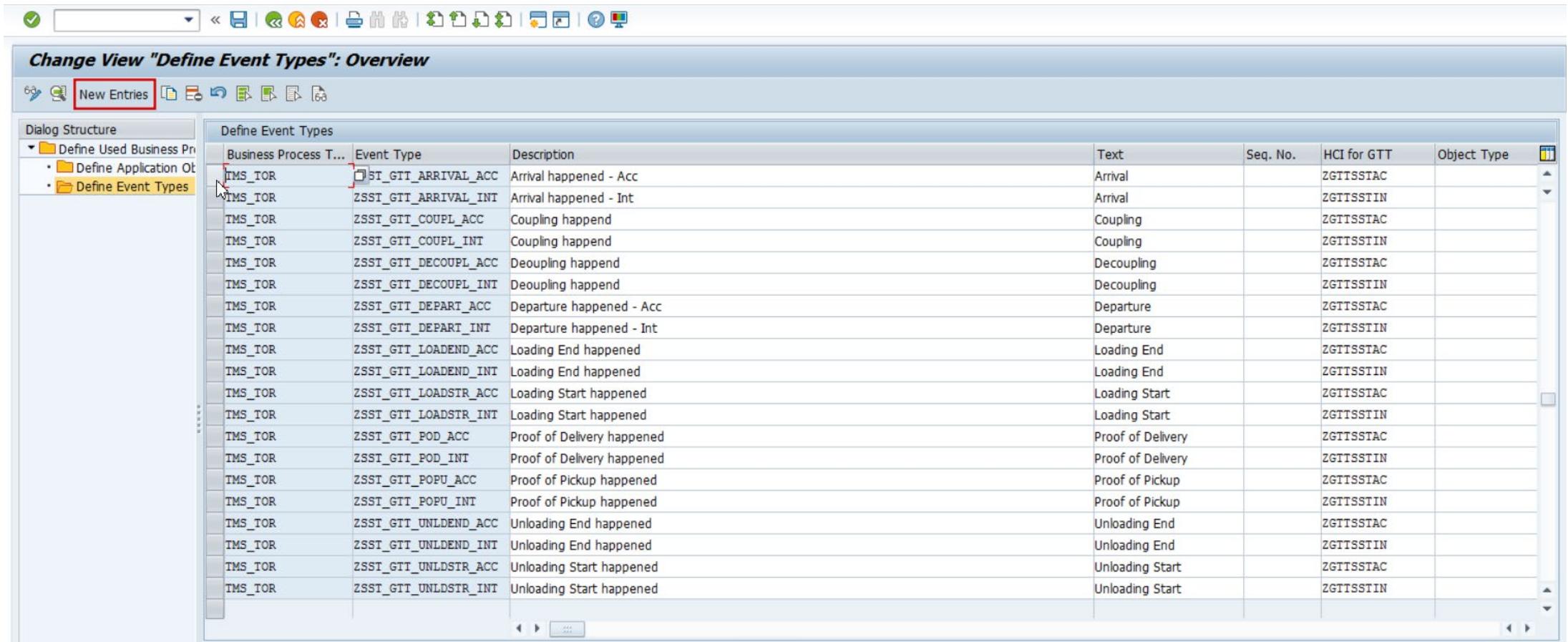
8-3: Choose the business process type from the **Define Used Business Process Types** on the right side

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 left sidebar, titled "Dialog Structure", shows a tree view with "Define Used Business Pro..." expanded, and "Define Application Obj..." and "Define Event Types" selected. A red box highlights the "New Entries" button in the toolbar at the top. The main area is a table titled "Define Event Types" with the following columns: Business Process T..., Event Type, Description, Text, Seq. No., HCI for GTT, and Object Type. The table lists numerous entries, mostly starting with "TMS\_TOR" and "ZSST\_GTT\_". The "Event Type" column contains names like "ST\_GTT\_ARRIVAL\_ACC", "ZSST\_GTT\_ARRIVAL\_INT", etc. The "Description" column provides a brief explanation for each entry. The "Text" column contains simplified versions of the descriptions. The "Seq. No." column is empty. The "HCI for GTT" and "Object Type" columns also contain empty fields.

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

# 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.

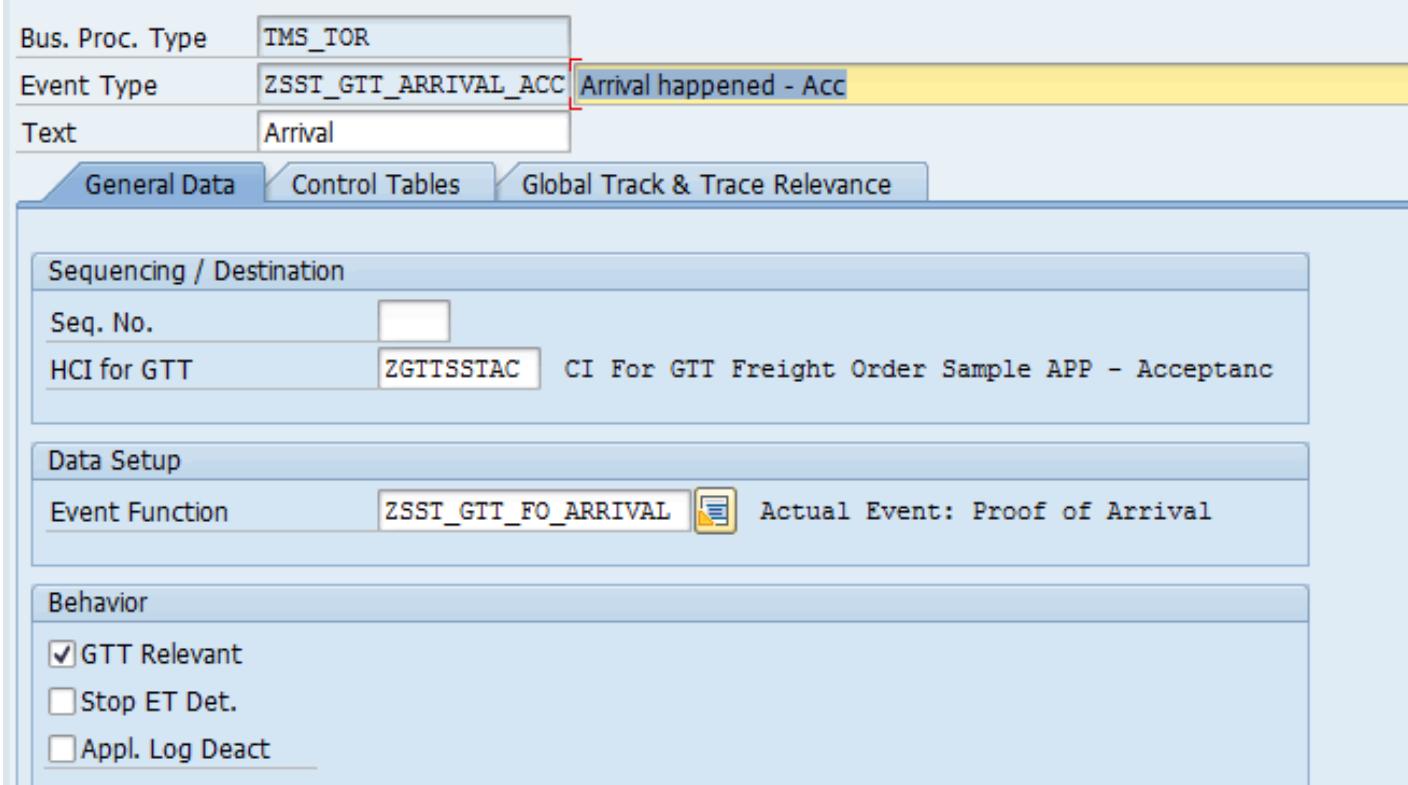
HCI for GTT ZGTTSSSTAC CI For GTT Freight Order Sample APP - Acceptanc

Data Setup

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

Behavior

GTT Relevant  
 Stop ET Det.  
 Appl. Log Deact



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

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

### Caution:

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

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

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

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

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

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

Click **Save**.

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

General Data    Control Tables    Global Track & Trace Relevance

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

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

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

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

The screenshot shows the SAP GUI interface for defining used business process types. The title bar reads "Change View "Define Used Business Process Types": Overview". The menu bar includes "Table View", "Edit", "Goto", "Selection", "Utilities", "System", and "Help". The toolbar contains various icons for navigation and data manipulation.

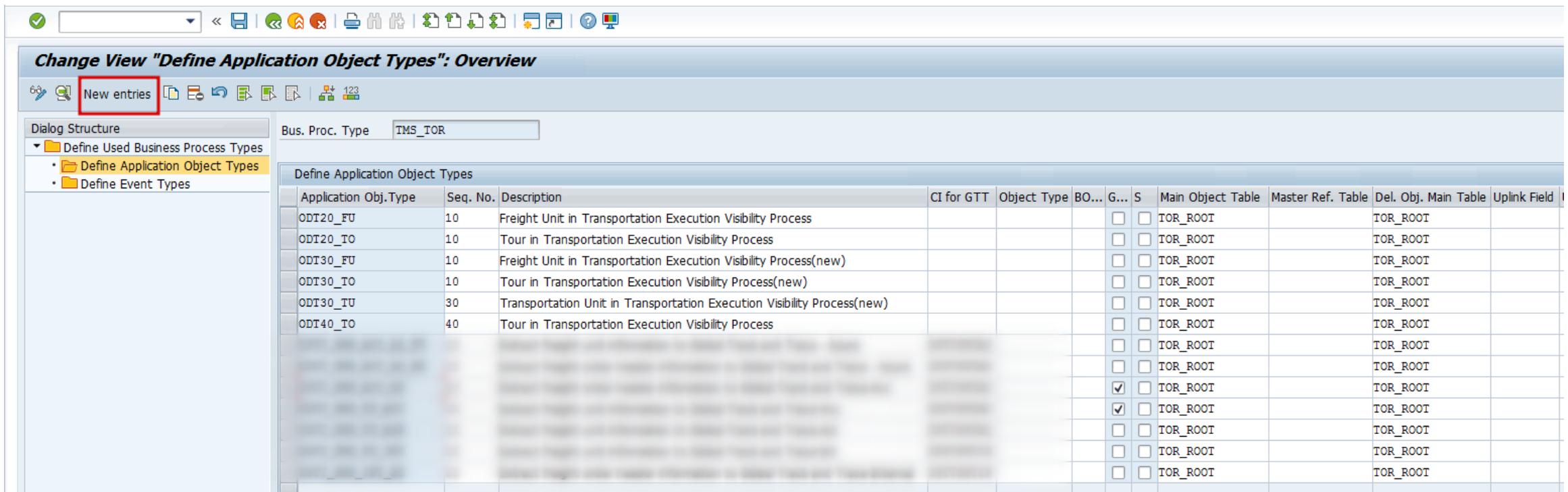
The left pane displays the "Dialog Structure" tree, which shows a folder named "Define Used Business Process Types" expanded. Inside this folder, there are three items: "Define Application Object Types" (highlighted with a red box), "Define Event Types", and another unnamed item. Below the tree is a large empty space.

The right pane contains a table titled "Define Used Business Process Types". The columns are "Bus. Proc. Type", "Update Mode", "BPT Process Mode", and "Description". The table lists several entries:

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

8-13: 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. 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 'Appl. Obj. Type' states: 'Extract freight order header information to Global Track and Trace-Acc'. Below these are five tabs: General Data, Control Tables, Object Identification (selected), Global Track & Trace Relevance, and Parameter Setup. The 'Object Identification' tab contains sections for Sequencing / Destination (Seq. No. 10, CI for GTT: ZGTTISSTAC, CI For GTT Freight Order Sample APP - Acceptance) and Business Object Reference (Object Type and BO Setup Fnct.). The 'Behavior' section at the bottom contains checkboxes: 'GTT Relevant' (checked), 'Stop AO Determ.', and 'Appl. Log Deact.'.

# 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 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**.

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

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

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

Method for determination of AOID

AOID Method Determine from Field

Application Object ID Source

First Field to Build Appl. Obj. ID Cntrl Tab. Type Main Object Table  
AO ID Field TOR\_ID

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

Determine AOID By Function

AOID Function

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

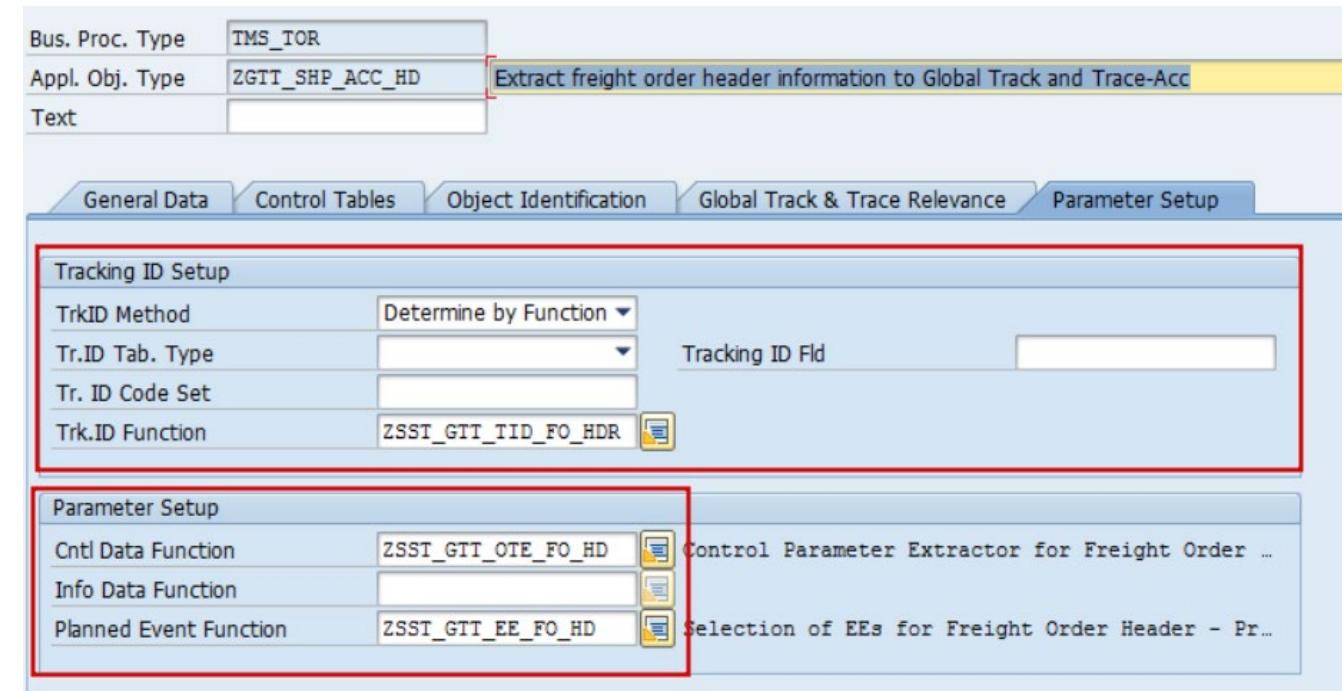
8-20: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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

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

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

Click **Save**.



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

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

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
<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 screenshot displays two side-by-side configurations for a business process.

**Left Configuration (Business Process Type):**

- Bus. Proc. Type: TMS\_TOR
- Appl. Obj. Type: ZGTT\_SHP\_FU\_ACC
- Text: Extract freight unit information to Global Track and Trace-Acc

**Right Configuration (Business Process Type):**

- 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 (Top Tab):**

- General Data: AOID Method set to "Determine from Field".
- Control Tables: Cntrl Tab. Type set to "Main Object Table" and AO ID Field set to "TOR\_ID".
- Object Identification: Cntrl Tab. Type set to "Main Object Table" and AO ID Field set to "TOR\_ID".
- Global Track & Trace Relevance: Active tab.
- Parameter Setup: TrkID Method set to "Determine by Function".
- Tracking ID Setup: Trk.ID Tab. Type, Tr. ID Code Set, and Trk.ID Function fields are present.
- Parameter Setup: Cntl Data Function, Info Data Function, and Planned Event Function fields are present.
- 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 (Bottom Tab):**

- GTT Rel. Method: Check Function (Function Module).
- GTT Rel. Function: ZSST\_GTT\_FO\_HDR
- Text: Extractor for relevance determination for Freight Order

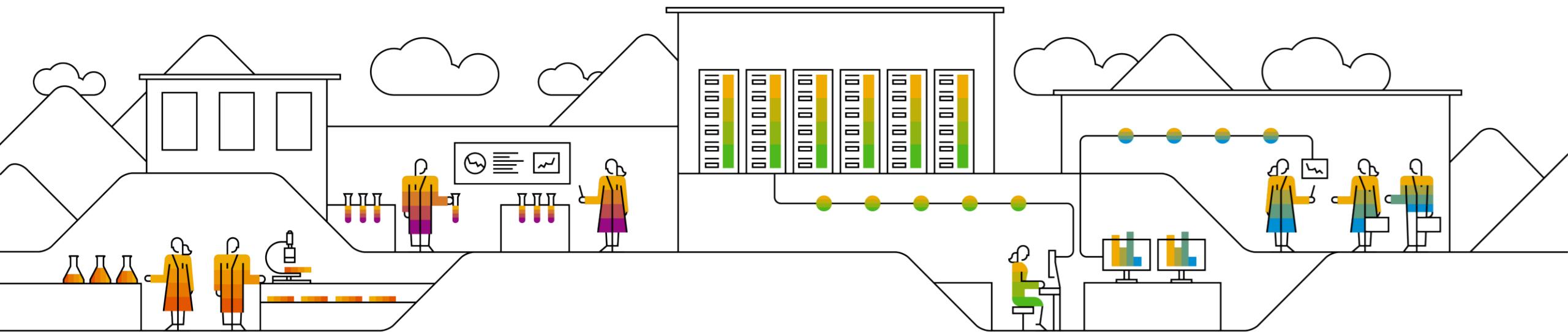
# C) Download ABAP Code from GitHub

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**.

 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

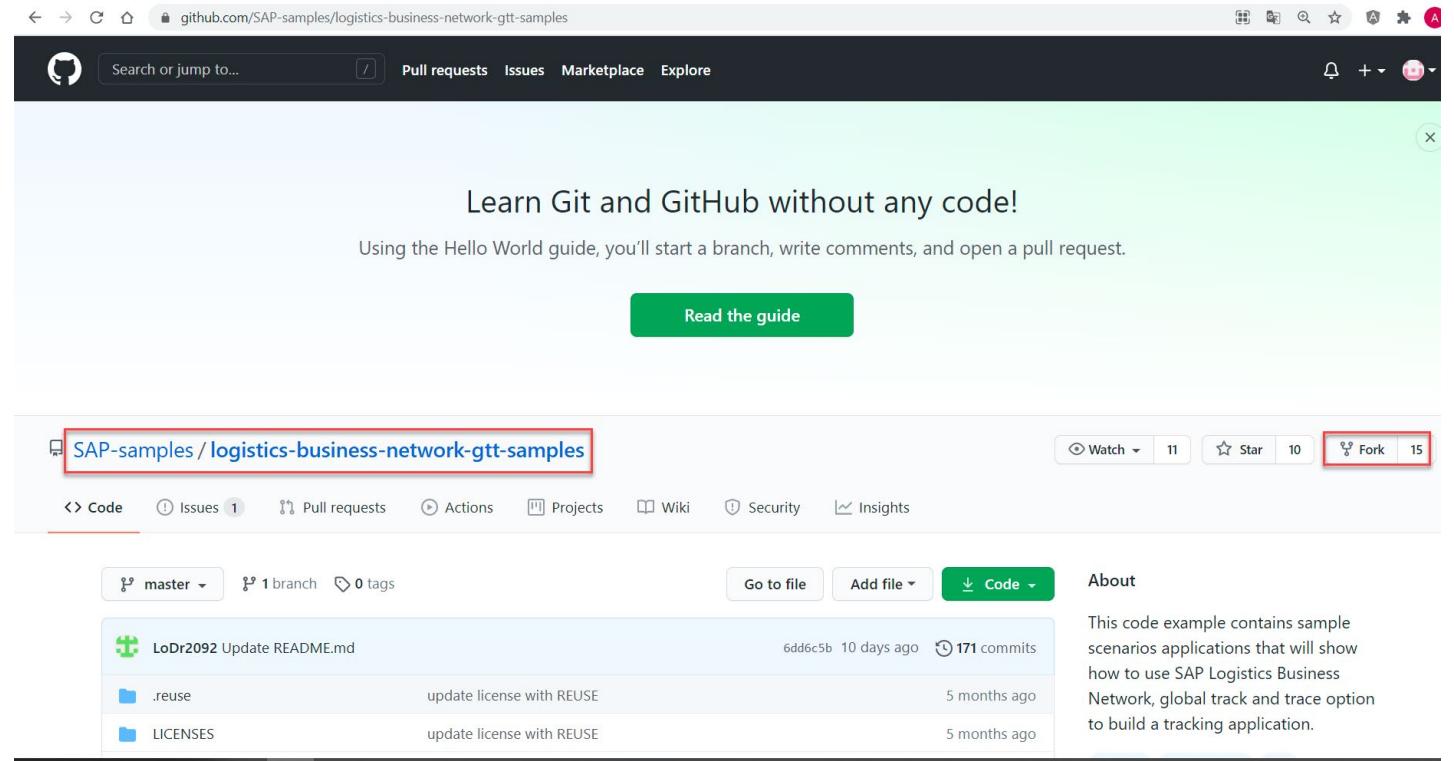
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `.$ZABAPGIT`).

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

# STEP 2: 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’

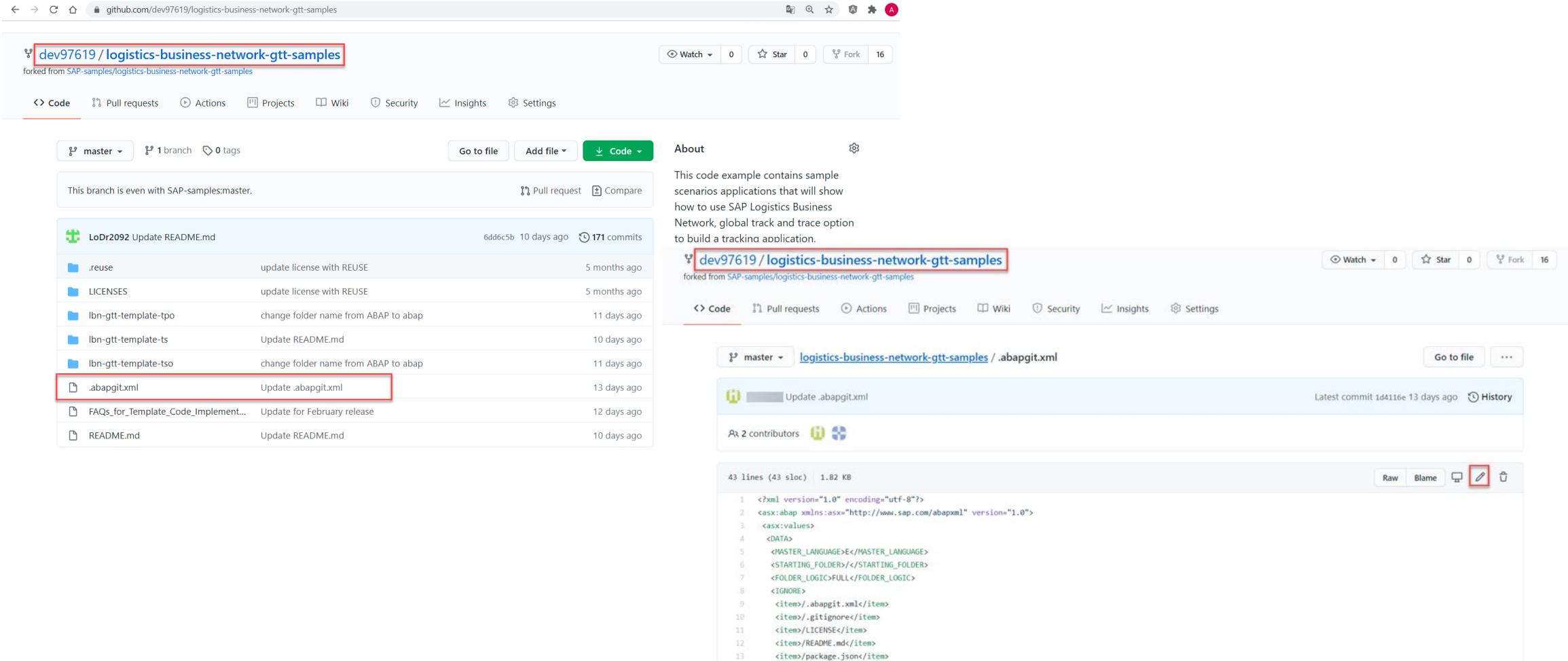
3-1: In the user’s account repository, click the file ‘.abapgit.xml’

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 for '.abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgitxml'. Other visible commits include 'Update README.md', 'update license with REUSE', and 'change folder name from ABAP to abap'. To the right of the commits, there is an 'About' section with a description of the repository, a 'Readme' link, a 'Releases' section (no releases published), and a 'Packages' section (no packages published).

Commit	Message	Date
LoDr2092 Update README.md	6dd6c5b 10 days ago	171 commits
.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 .abapgitxml</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 landing page for `dev97619 / logistics-business-network-gtt-samples`. The bottom view is a detailed look at the `.abapgit.xml` file within the repository.

**Repository Landing Page:**

- Branch: master
- Pull requests: 1
- Tags: 0
- Actions: 0
- Projects: 0
- Wiki: 0
- Security: 0
- Insights: 0
- Settings: 0

**.abapgit.xml File View:**

- Branch: master
- Last commit: Update .abapgit.xml (1d4116e, 13 days ago)
- Contributors: 2
- Raw: 43 lines (43 sloc) 1.82 KB
- Blame: 0
- Copy: 0
- Edit: 
- Diff: 0

```
<?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>/LICENSE</item>
        <item>/README.md</item>
        <item>/package.json</item>
      </IGNORE>
    </DATA>
  </asx:values>
</asx:abap>
```

## 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.

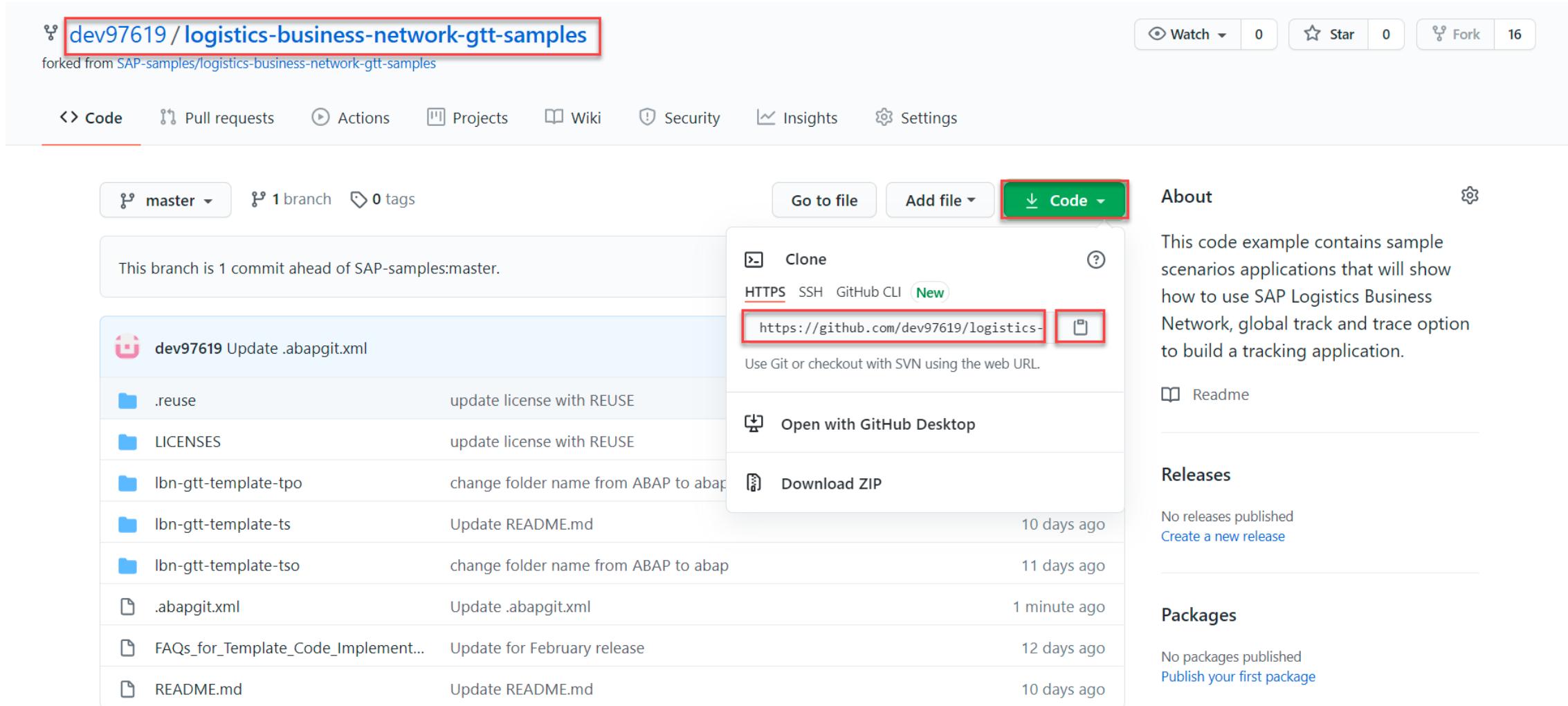
3-4: Commit change

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The '.abapgit.xml' file is open, showing its XML content. A specific line of code, '<STARTING\_FOLDER>/</STARTING\_FOLDER>', is highlighted with a red box. To the right, a 'Commit changes' dialog is displayed, also containing a red box around the same line of code. The dialog includes fields for a commit message ('Update .abapgit.xml'), an optional extended 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.' Below 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 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 0 stars and 16 forks. The 'Code' tab is selected. A commit from 'dev97619' titled 'Update .abapgit.xml' is highlighted. The 'Code' dropdown menu is open, showing the 'Clone' section with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples>. The URL is highlighted with a red box, and a copy icon is also highlighted with a red box.

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.

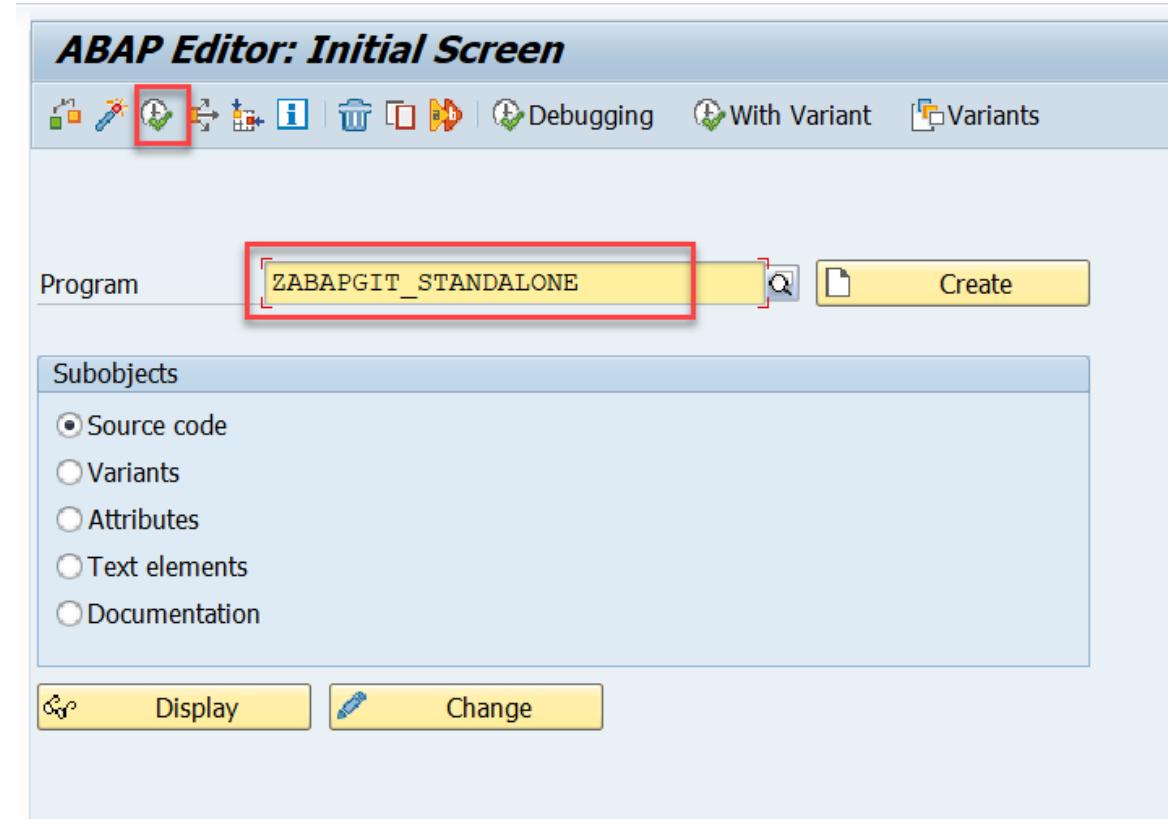
No releases published  
Create a new release

No packages published  
Publish your first package

## STEP 4: Download ABAP code from GitHub

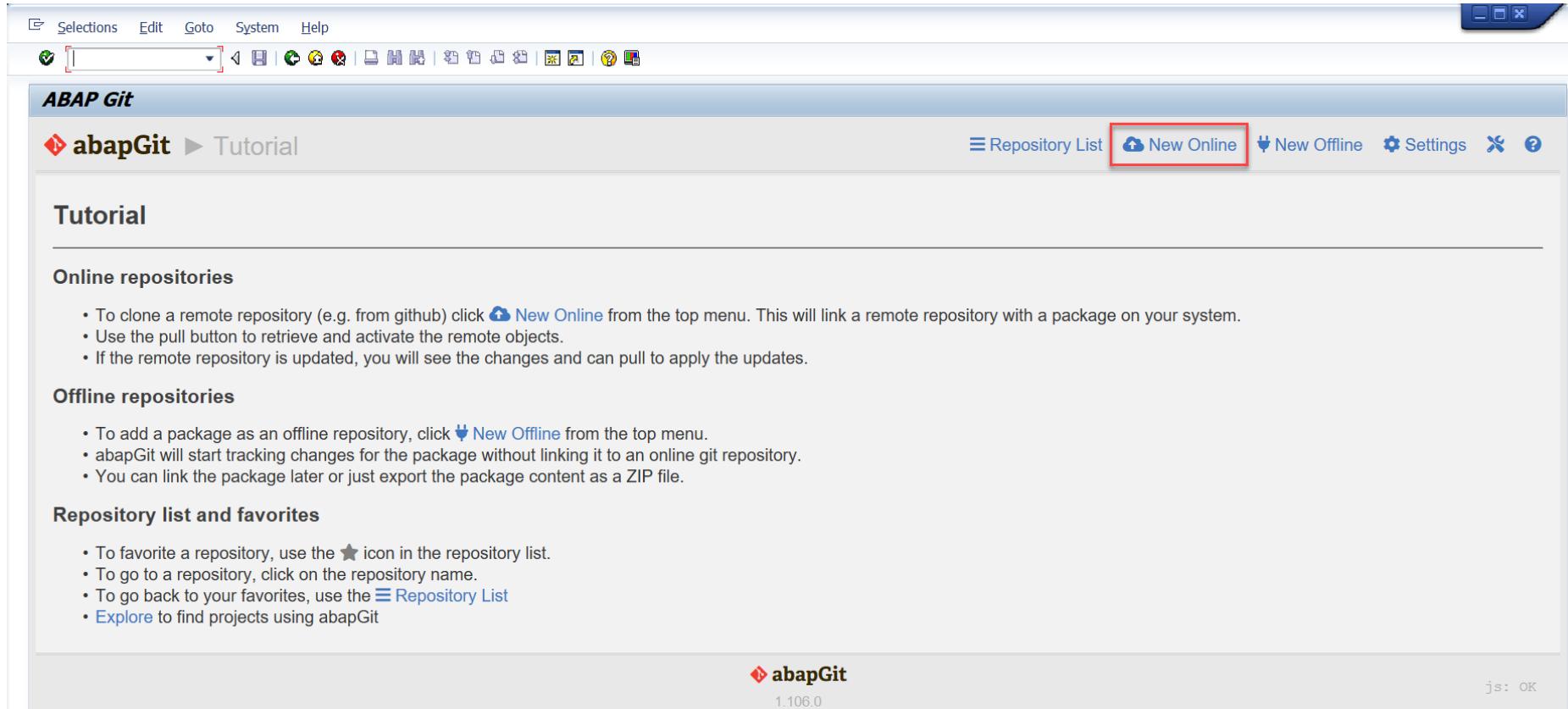
4-1: Enter T-code **SE38** and fill in the report name from STEP 1,  
**ZABAPGIT\_STANDALONE**

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>

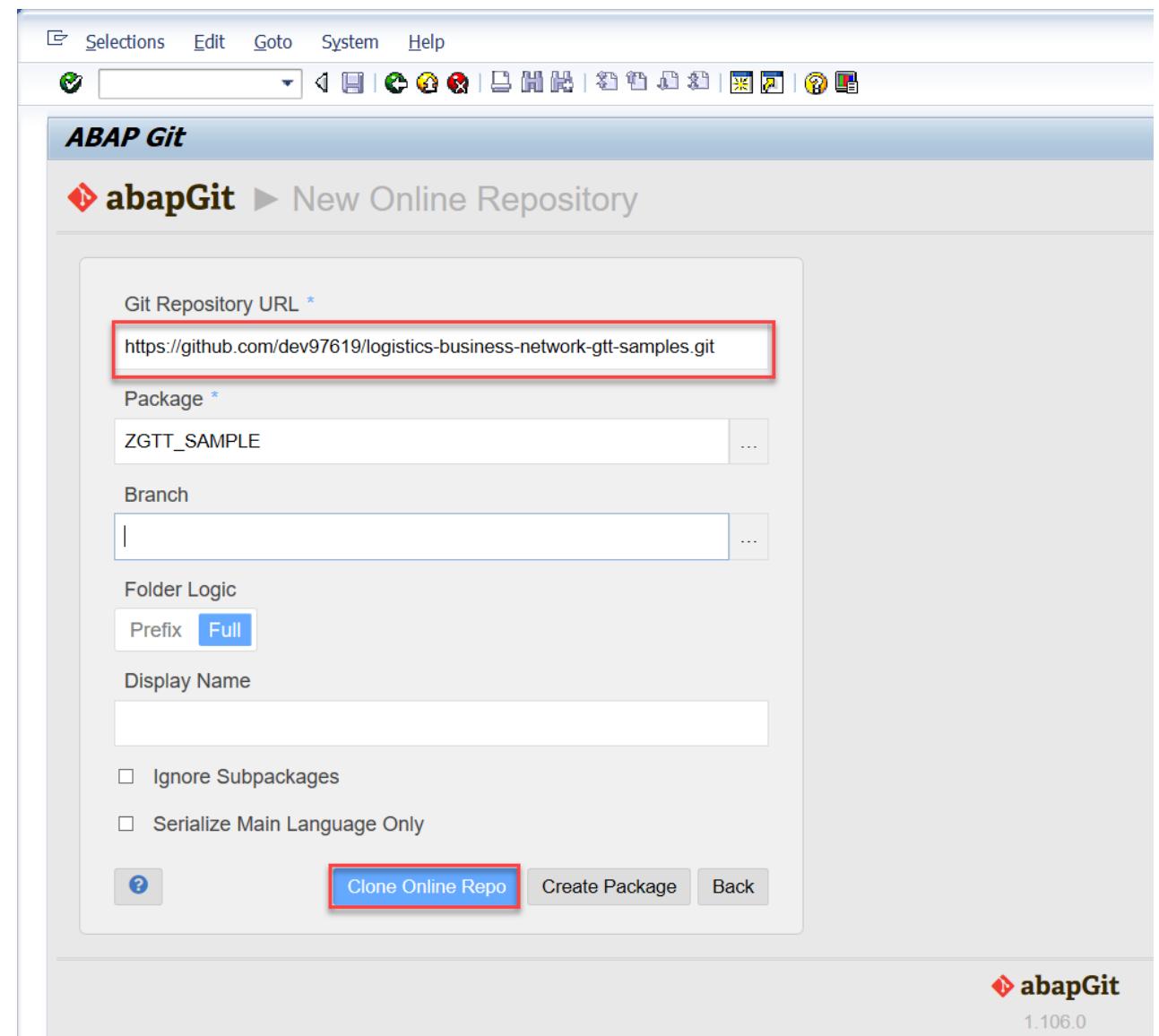
## Caution:

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.

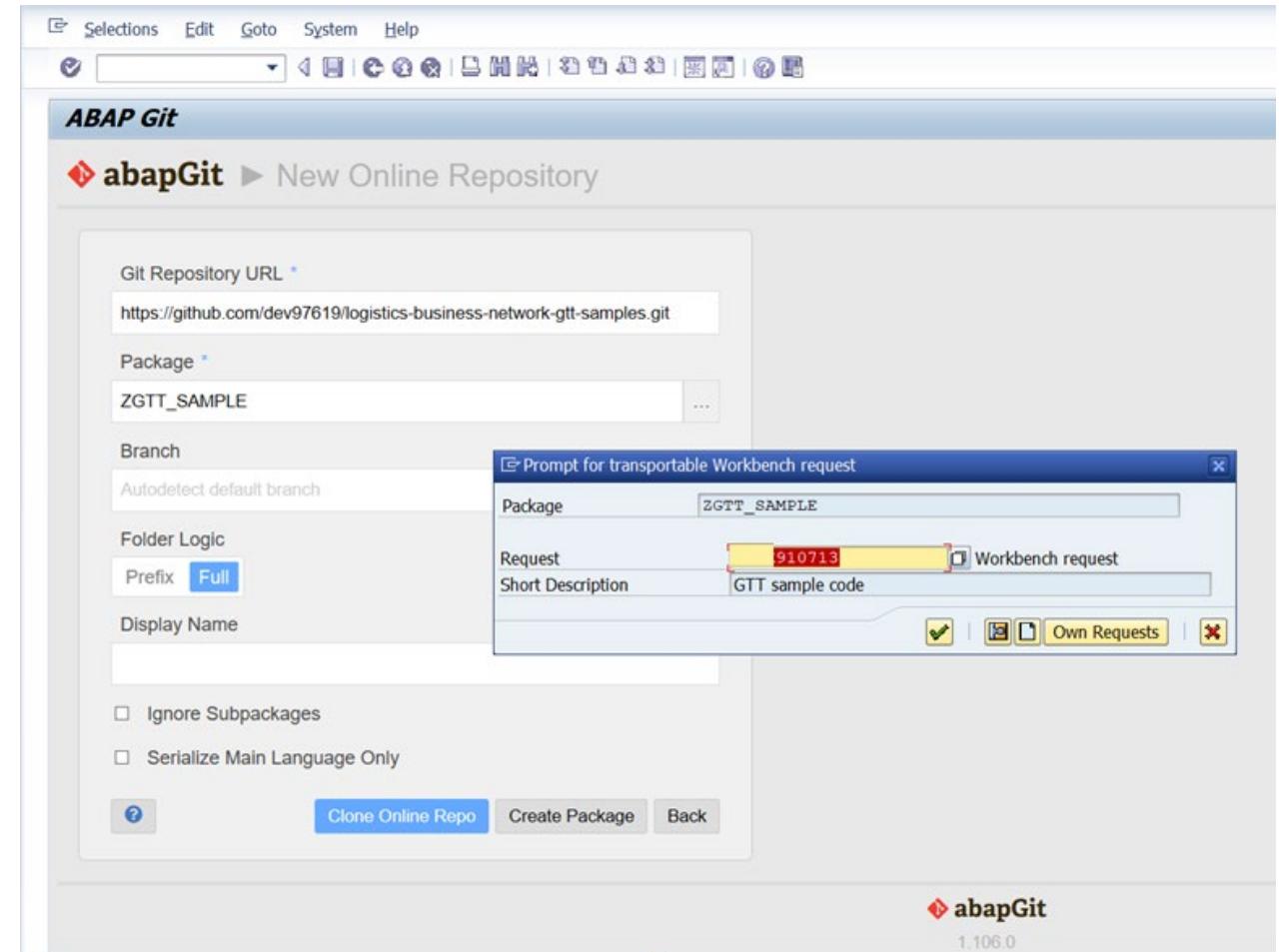
4-6: Set *Full* for **Folder Logic**

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

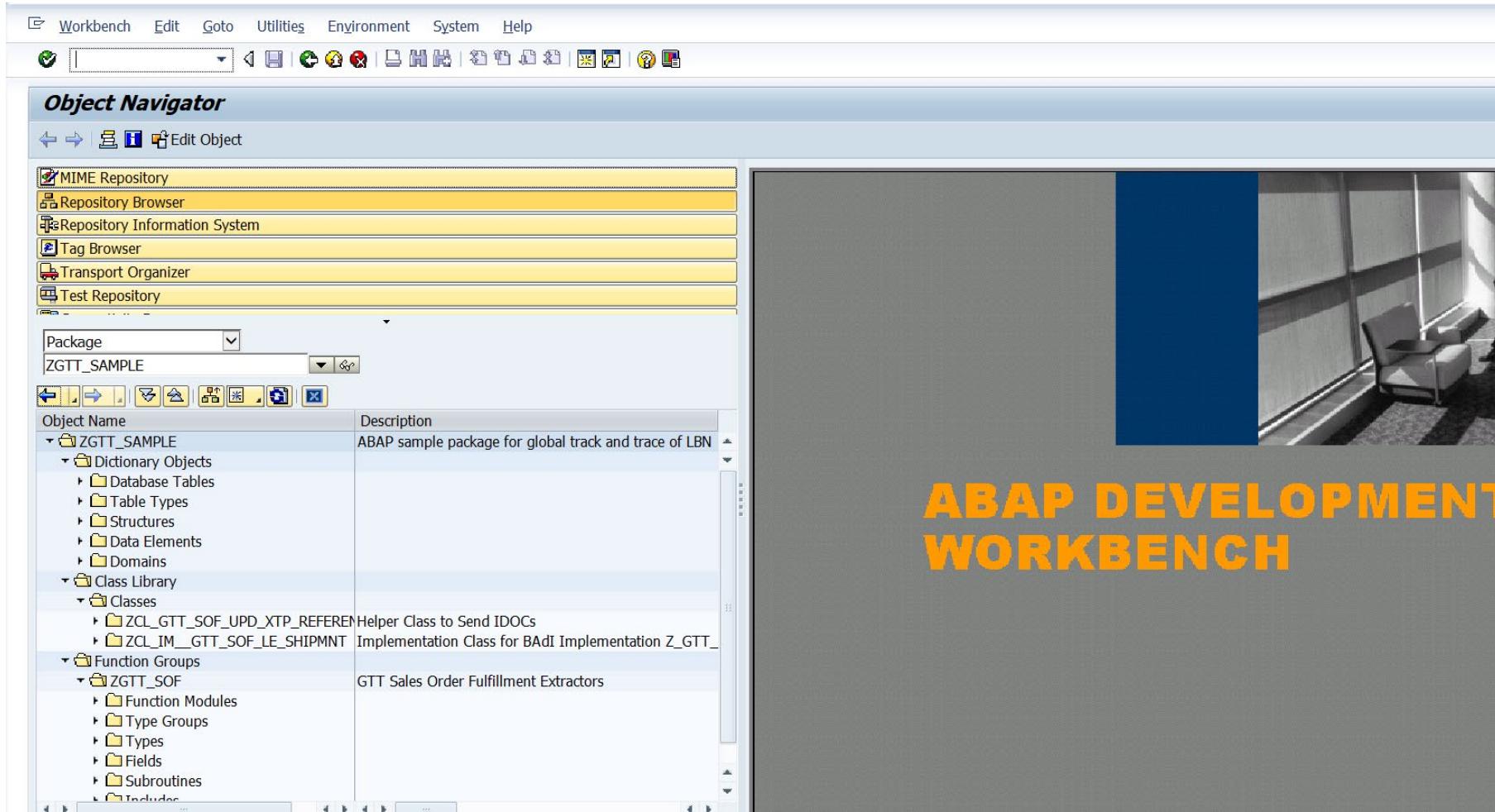
4-9: Click Pull to pull down the latest version code

The screenshot shows the ABAP Git interface within the SAP IDE. The title bar reads "ABAP Git". The main area displays a list of files under the repository "logistics-business-network-gtt-samples". The "Pull" button in the toolbar is highlighted with a red box. The table below lists the file type, name, path, and various download and diff options.

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

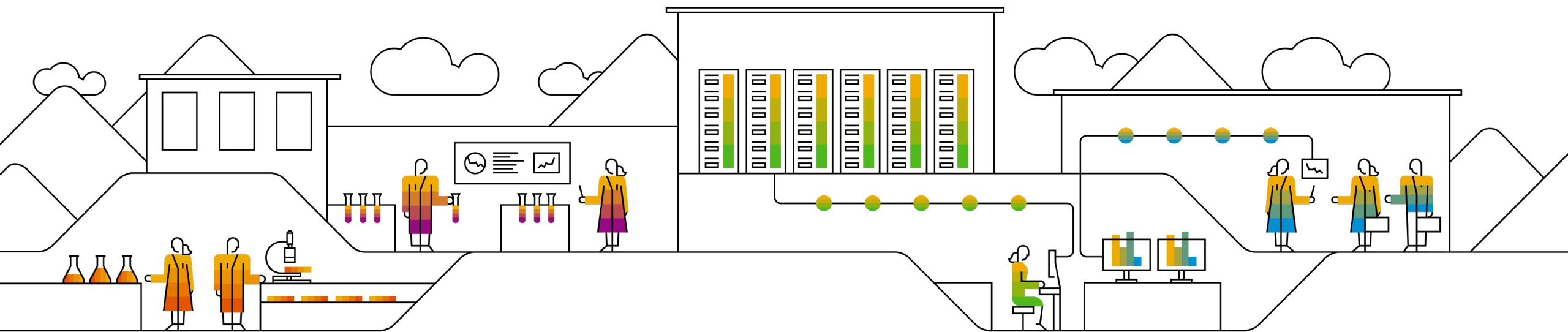
# STEP 4: Download ABAP code from GitHub

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"

dev97619 / logistics-business-network-gtt-samples  
forked from SAP-samples/logistics-business-network-gtt-samples

Code Pull requests Actions Projects Wiki Security Insights Settings

master 2 branches 0 tags Go to file Add file Code

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

dev97619 Update .abapgit.xml	8b46800 29 minutes ago	173 commits
.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
.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

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 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 '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 it is a 'Danger Zone' section with four options: 'Change repository visibility', 'Transfer ownership', 'Archive this repository', and 'Delete this repository'. The 'Delete this repository' button is highlighted with a red border.

None Save

Theme Chooser  
Select a theme to publish your site with a Jekyll theme using the gh-pages branch. [Learn more.](#)

Choose a theme

Danger Zone

Change repository visibility  
You cannot change the visibility of a fork. Please [duplicate the repository](#). [Change visibility](#)

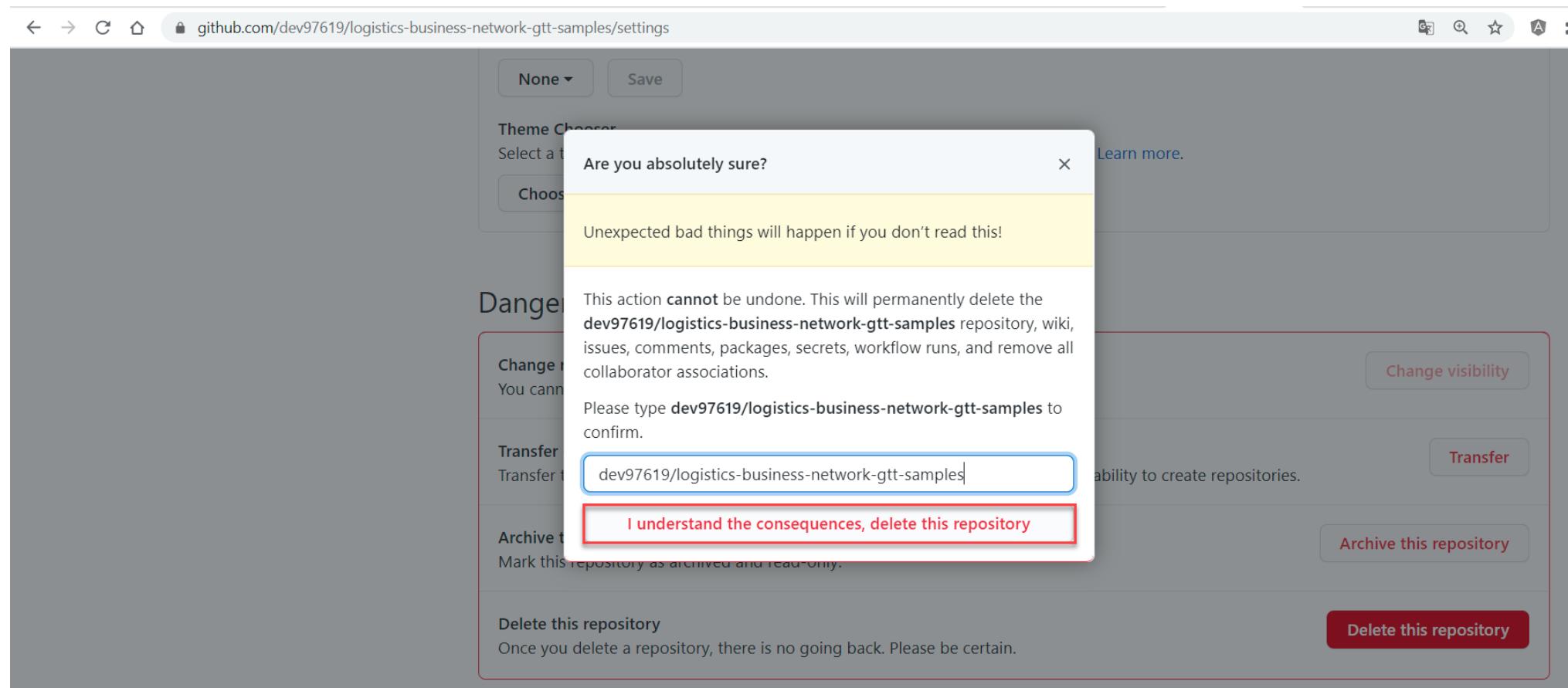
Transfer ownership  
Transfer this repository to another user or to an organization where you have the ability to create repositories. [Transfer](#)

Archive this repository  
Mark this repository as archived and read-only. [Archive this repository](#)

Delete this repository  
Once you delete a repository, there is no going back. Please be certain. [Delete this repository](#)

# STEP 1: Delete the user's Account Repository

1-4: The popup shows some waring 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

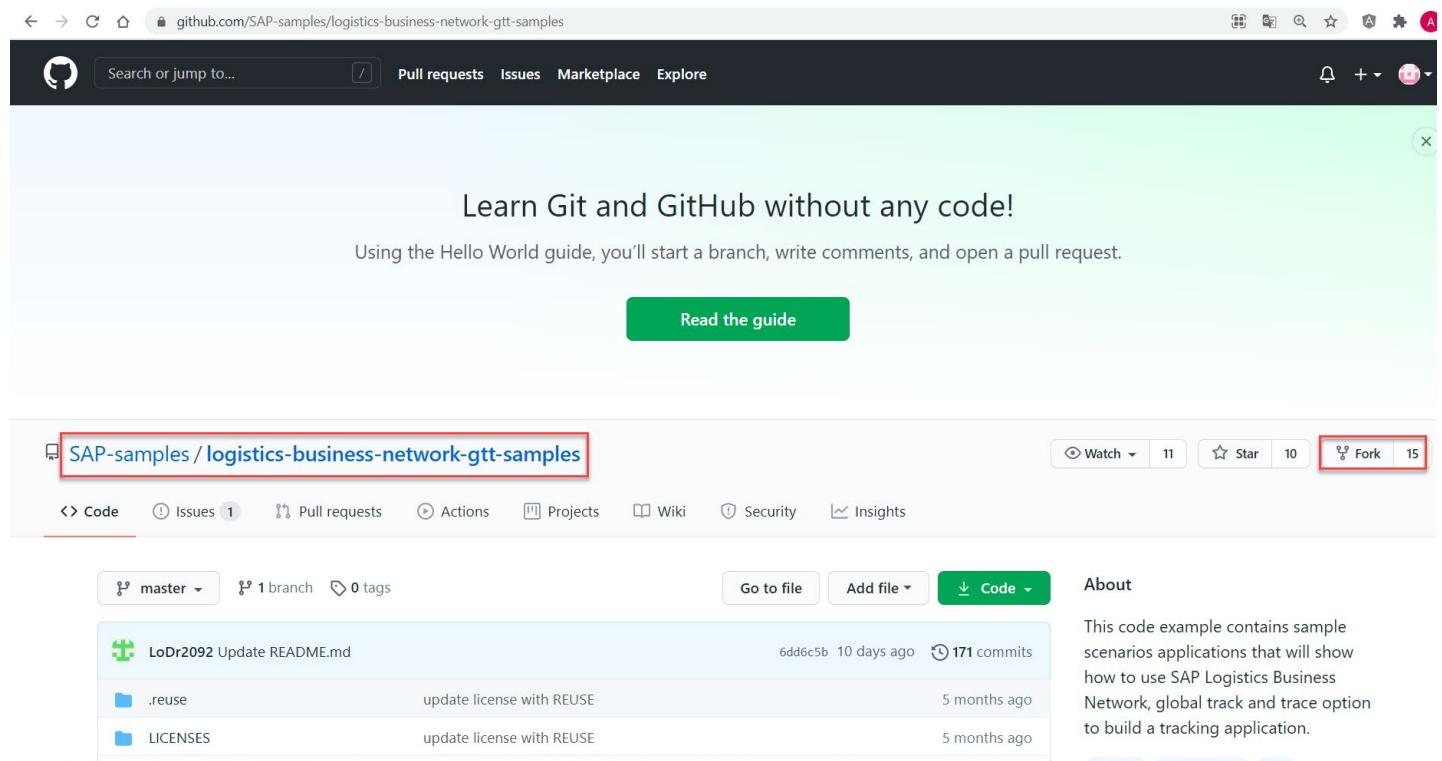
1-5: The user account's repository is deleted.

The screenshot shows a dark-themed GitHub interface. At the top, there is a navigation bar with a search bar, a pull requests button, an issues button, a marketplace button, and an explore button. On the right side of the top bar are icons for notifications, a plus sign, and a profile picture. Below the navigation bar, a message box contains the text "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." This message is highlighted with a red rectangular border. To the left of the message box, there is a sidebar with sections for "Create your first project" (with "Create repository" and "Import repository" buttons), "Working with a team?" (with "Create an organization" button), and a large central area titled "Learn Git and GitHub without any code!" with a "Read the guide" button. The entire interface has a light blue background.

# 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. A red box highlights the repository name 'dev97619 / logistics-business-network-gtt-samples' in the header and the '.abapgit.xml' file in the commit list.

This branch is even with SAP-samples:master.

File	Commit Message	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 .abapgitxml</b>	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.

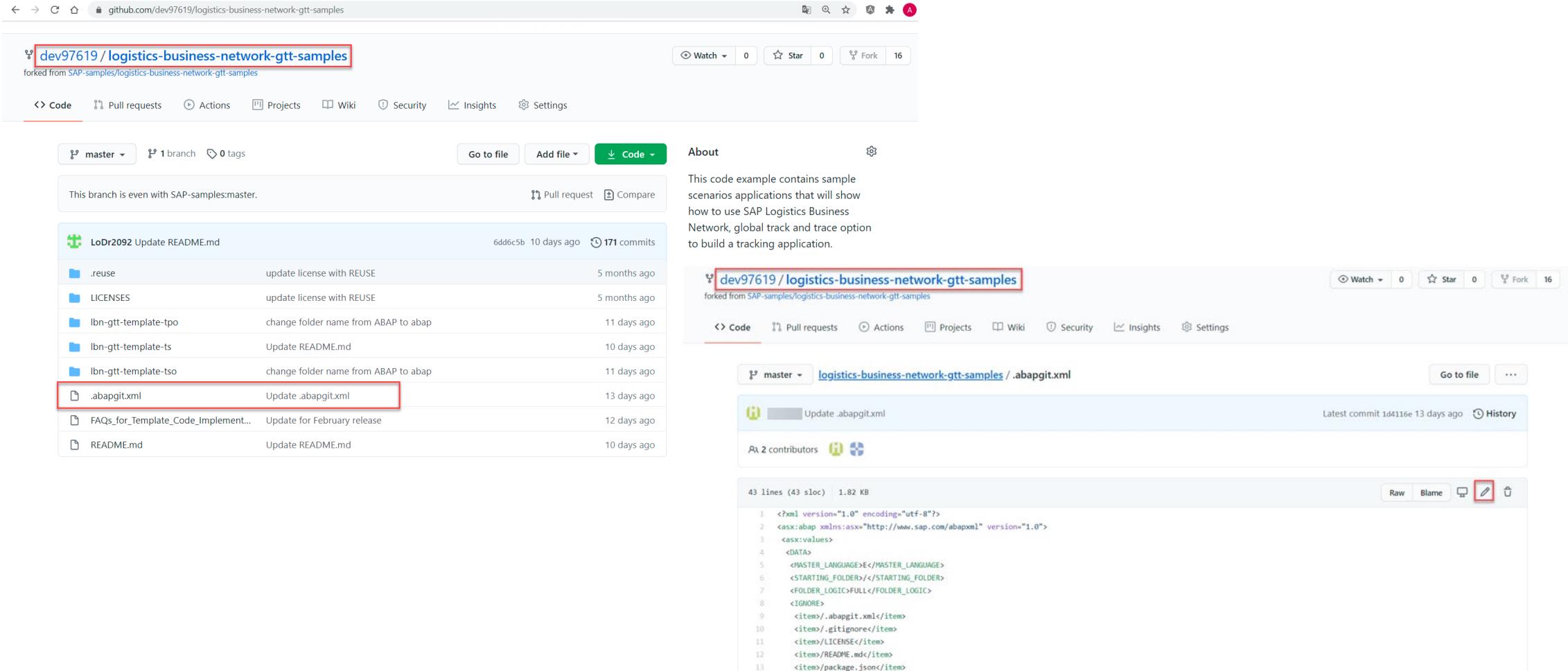
**Readme**

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

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

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

3-2: 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`, which is a fork of `SAP-samples/logistics-business-network-gtt-samples`. It displays a list of commits, including one from LoDr2092 that updated the `README.md` file. A commit to update the `.abapgit.xml` file is highlighted with a red box. The bottom page shows the contents of the `.abapgit.xml` file, which is an XML configuration for ABAP Git. The file includes sections for `MASTER_LANGUAGE`, `STARTING_FOLDER`, `FOLDER_LOGIC`, and `IGNORE`.

```
<?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>/LICENSE</item>
        <item>/README.md</item>
        <item>/package.json</item>
      </IGNORE>
    </DATA>
  </asx:values>
</asx:abap>
```

## 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 for editing. The configuration line '6 <STARTING\_FOLDER>/</STARTING\_FOLDER>' has been replaced by '6 <STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>'. The commit message field contains 'Update .abapgit.xml'. The 'Commit changes' button is highlighted with a red border.

Code Pull requests Actions Projects Wiki Security Insights Settings

Watch 0 Star 0 Fork 16

logistics-business-network-gtt-samples / .abapgit.xml in master

Edit file Preview changes

```
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>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

-o- Commit directly to the master branch.

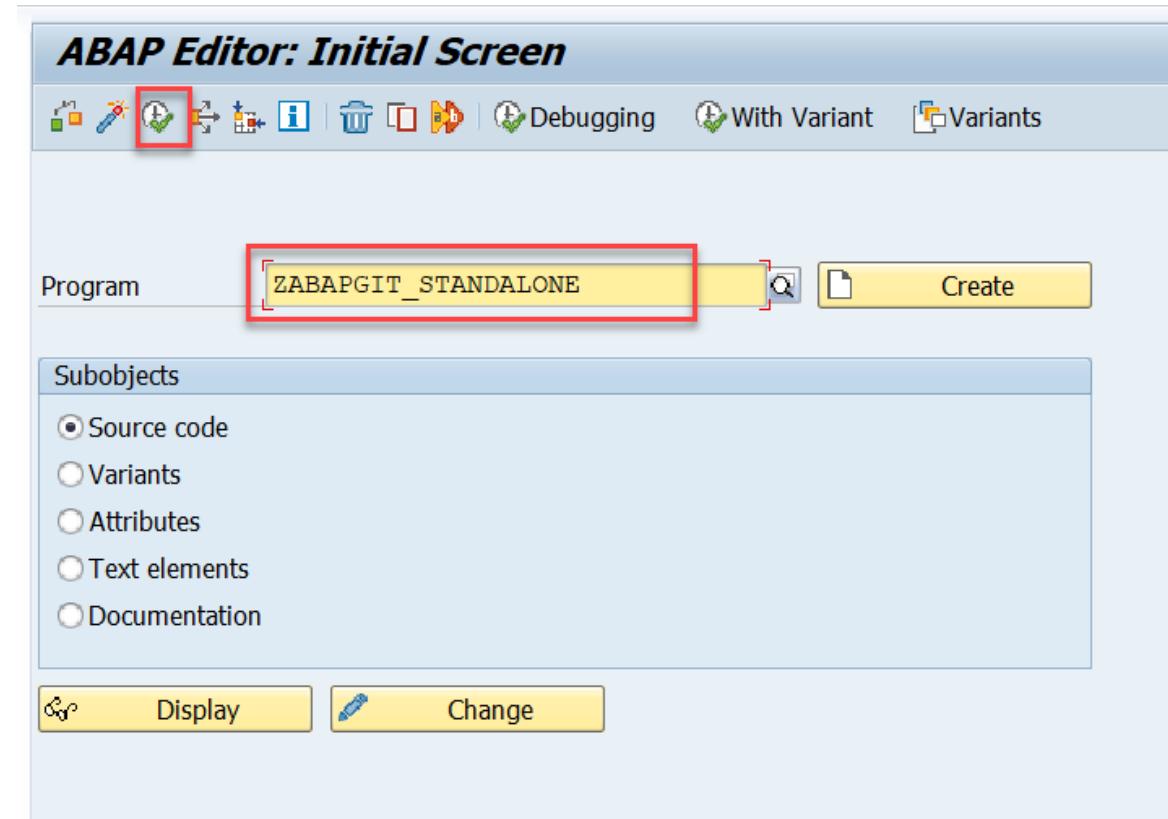
! Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes Cancel

## STEP 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

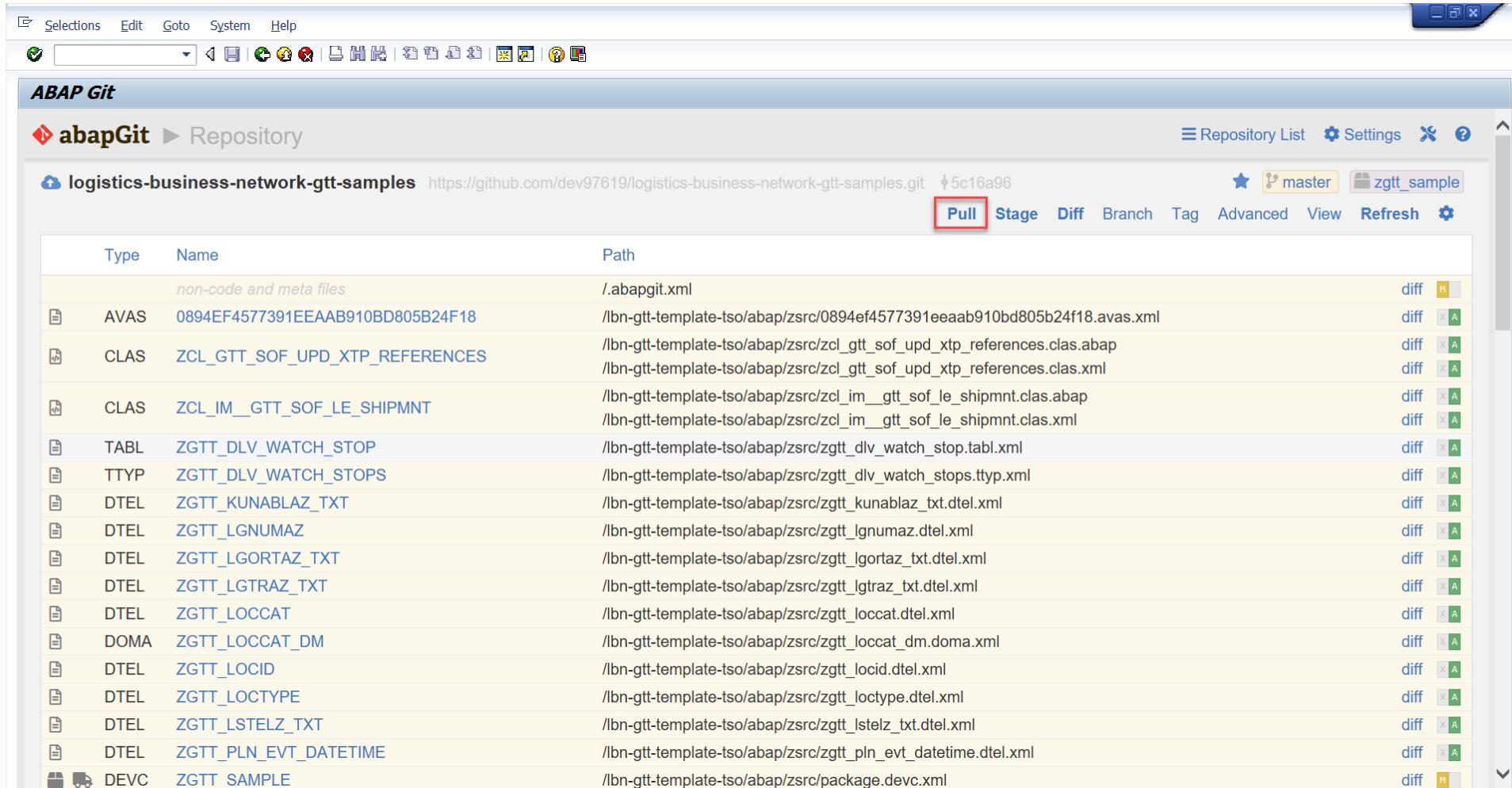
4-3: Check the URL is not changed after your recreation of repository copy. Access the TSO Repository by clicking button.

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 a toolbar of icons. The main area is titled "ABAP Git" and displays a "Repository List". A navigation bar at the top right includes "New Online", "New Offline", "Settings", and a help icon. Below the navigation bar, there is a "Filter:" input field and checkboxes for "Only Favorites" and "Detail". The repository list table has columns: Name, Url, Package, Branch, and Action. One row is shown for "logistics-business-network-gtt-samples" with the URL "github.com/dev97619/logistics-business-network-gtt-samples.git", package "zgtt\_sample", branch "master", and actions "Check", "Stage", "Patch", and "Settings". The "Settings" button for this row is highlighted with a red box. At the bottom of the interface, the "abapGit" logo and version "1.106.0" are displayed, along with a status message "js: OK".

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

# STEP 4: Update ABAP code from GitHub

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

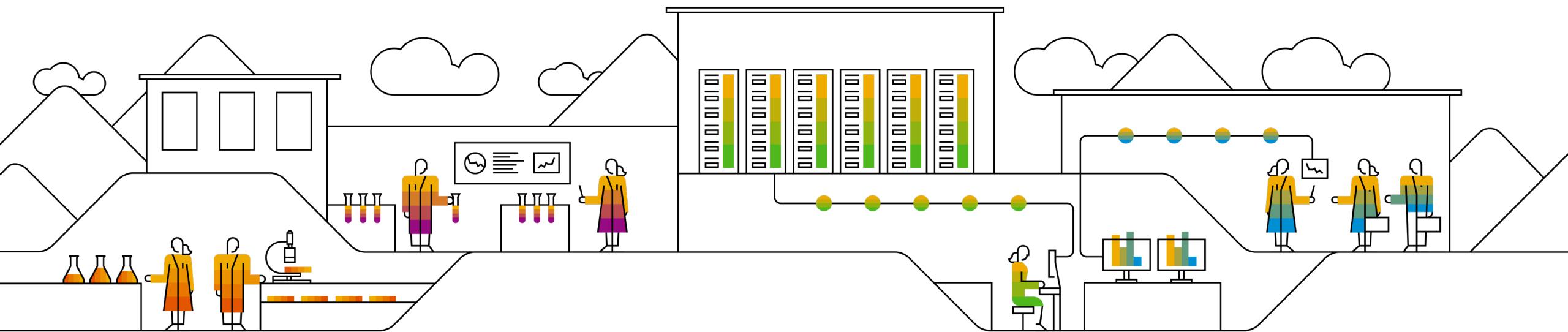


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git". Underneath, the repository path "abapGit > Repository" is shown, along with the URL "logistics-business-network-gtt-samples" and the commit hash "5c16a96". A "master" branch is selected. The main area displays a table of files with columns for Type, Name, Path, and Diff status. The "Pull" button in the header is highlighted with a red box.

Type	Name	Path	Diff
	non-code and meta files	/abapgit.xml	M
AVAS	0894EF4577391EEAAB910BD805B24F18	//lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	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	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	A
TABL	ZGTT_DLV_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	A
TTYP	ZGTT_DLV_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_kunablaTxt.dtel.xml	A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zggt_lgnuzaz.dtel.xml	A
DTEL	ZGTT_LGORAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgoraz_txt.dtel.xml	A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	A
DTEL	ZGTT_LOCCAT	//lbn-gtt-template-tso/abap/zsrc/zggt_locat.dtel.xml	A
DOMA	ZGTT_LOCCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zggt_locat_dm.doma.xml	A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.datetime.dtel.xml	A
DEV	ZGTT_SAMPLE	//lbn-gtt-template-tso/abap/zsrc/package.devcl.xml	M

# C) Download ABAP Code from GitHub

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

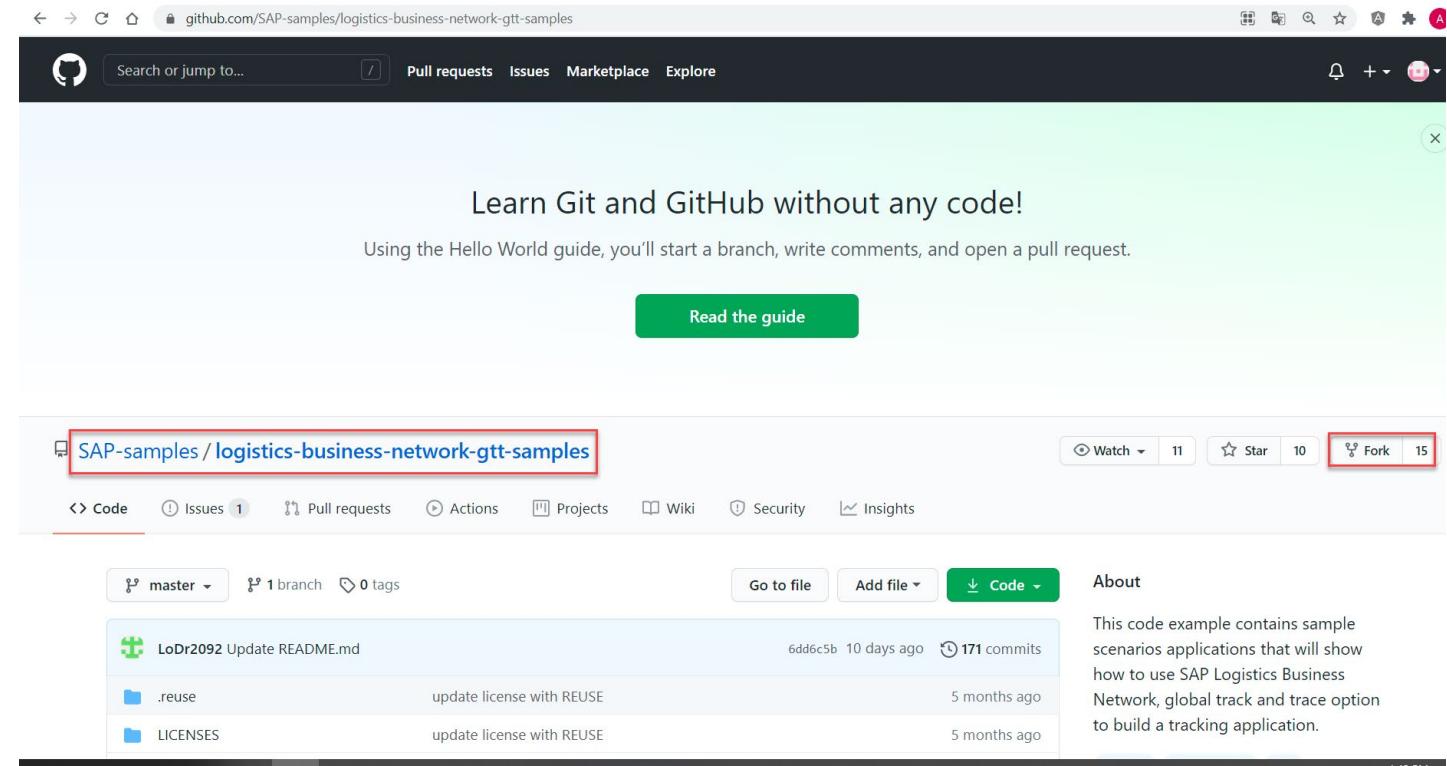


# STEP 1: Fork Sample code Repository

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’

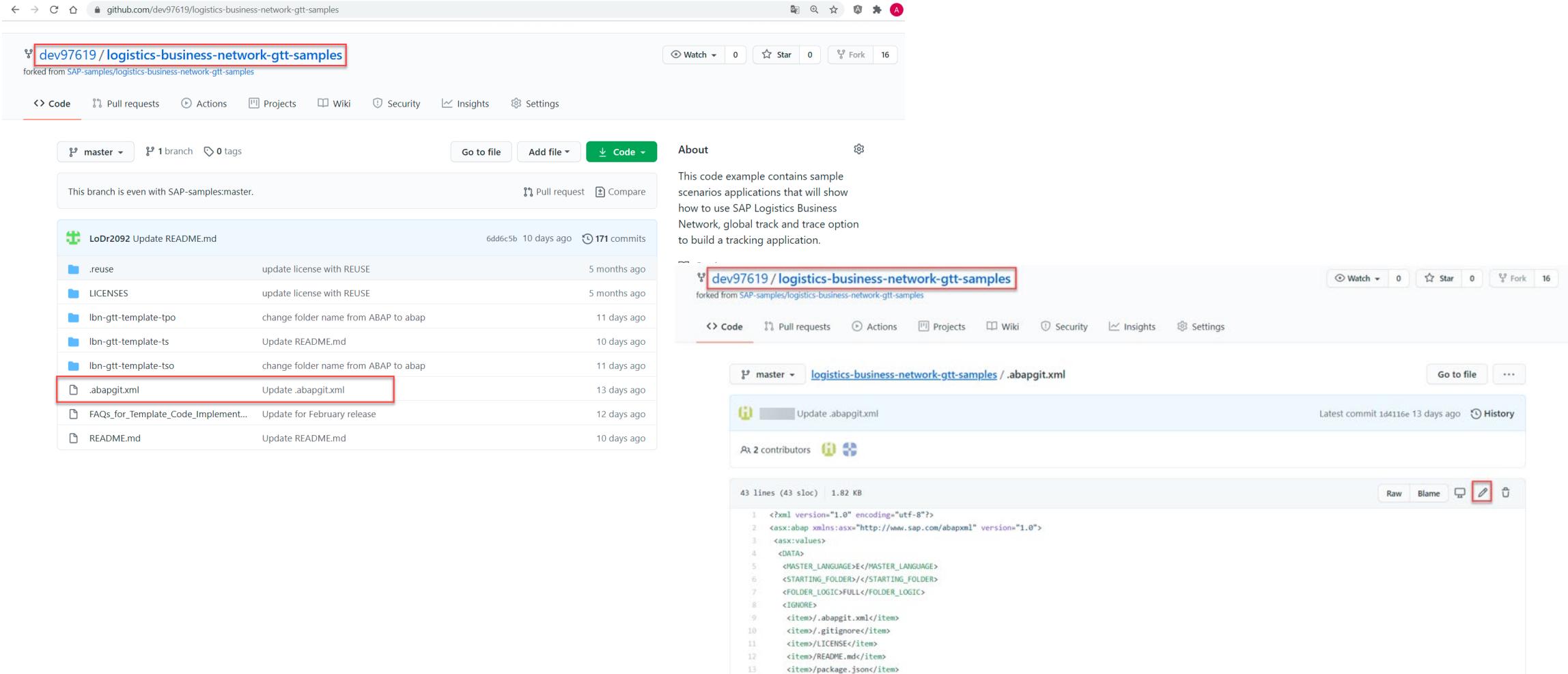
2-1: In the user’s account repository, click the file ‘.abapgit.xml’

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 main content area displays a list of commits. A commit for '.abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgitxml'. Other visible commits include 'Update README.md', 'update license with REUSE', and 'change folder name from ABAP to abap'. To the right of the commits, there is an 'About' section with a description of the repository, a 'Readme' link, a 'Releases' section (which is empty), and a 'Packages' section (which is also empty).

Commit	Message	Date
LoDr2092 Update README.md	6dd6c5b 10 days ago	171 commits
.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 .abapgitxml</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 side-by-side. The left page is for the repository `dev97619 / logistics-business-network-gtt-samples`, which is forked from `SAP-samples/logistics-business-network-gtt-samples`. The right page is for the same repository. Both pages show the `Code` tab selected. On the left, there is a list of commits, one of which is highlighted with a red box around its commit message: `[LoDr2092] Update README.md`. On the right, the `.abapgit.xml` file is open in a code editor, also with a red box around the edit icon in the top right corner of the editor interface.

## 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 the following XML content:

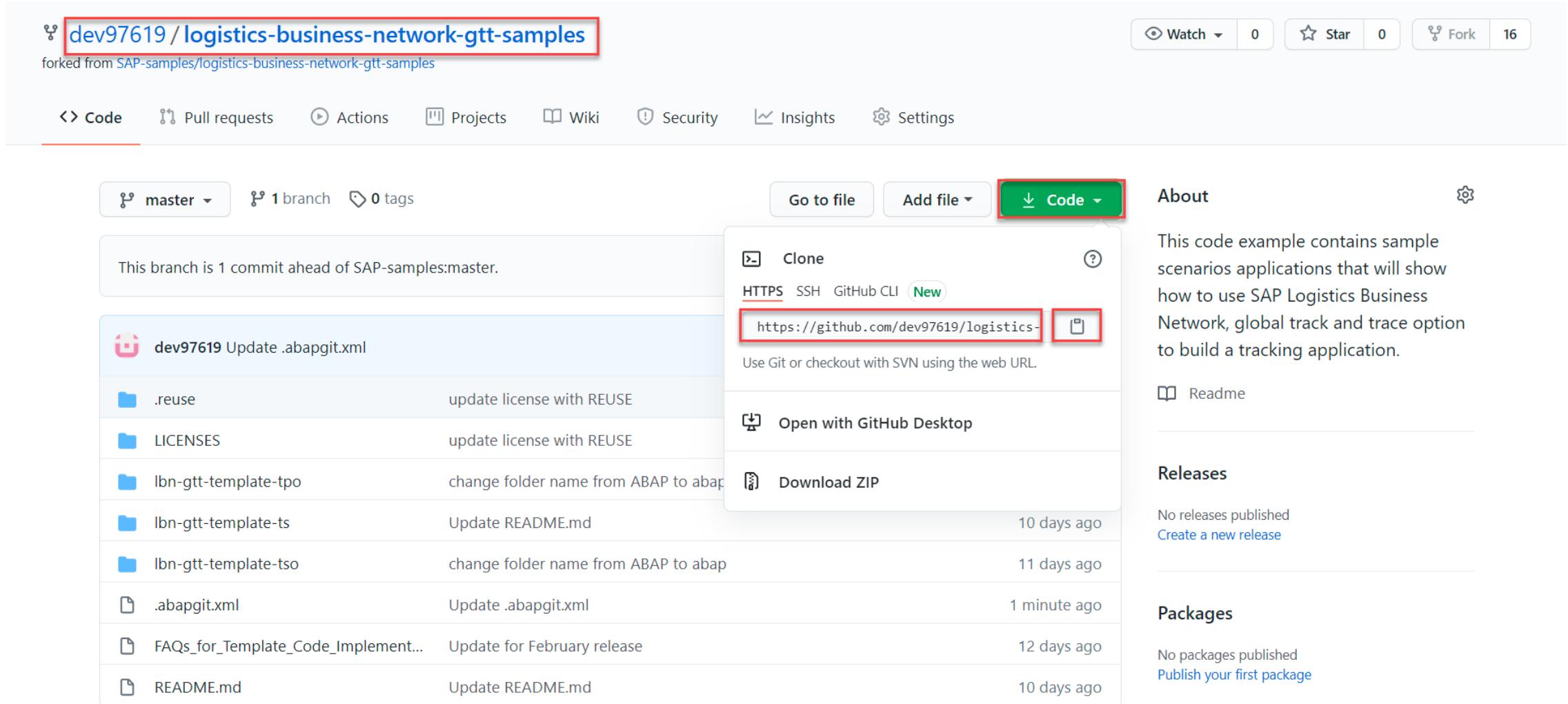
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>
7       <FOLDER_LOGIC>FULL</FOLDER_LOGIC>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

A red box highlights the line '<STARTING\_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING\_FOLDER>'. To the right, a 'Commit changes' dialog is open, containing the following fields:

- Commit message: Update .abapgit.xml
- Description placeholder: Add an optional extended description...
- Branch selection:
  - o- Commit directly to the master branch.
  - ! Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)
- Commit changes button (highlighted with a red border)
- Cancel button

## 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. A message indicates it is 1 commit ahead of SAP-samples:master. A list of commits shows updates to '.abapgit.xml' and other files. On the right, a 'Code' dropdown menu is open, with the 'Clone' section showing the repository URL: <https://github.com/dev97619/logistics-business-network-gtt-samples>. The 'Copy' button next to the URL is highlighted with a red box.

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-business-network-gtt-samples> 

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](#)

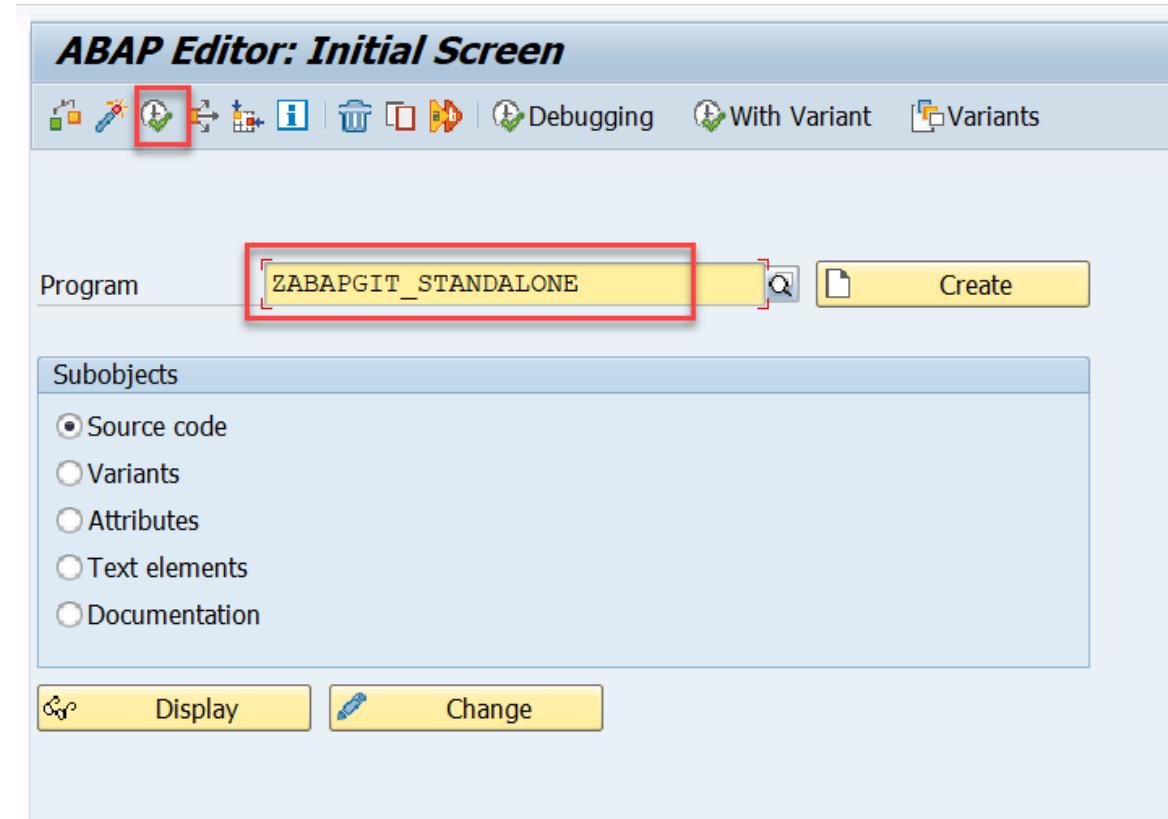
Packages

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

## STEP 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.
  - 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. A repository card for 'logistics-business-network-gtt-samples' is displayed, showing its URL and a commit hash. The 'Advanced' menu is open, with the 'Remove' option highlighted by a red box. The 'Advanced' menu also lists other options like 'Reset Local (Force Pull)', 'Checkout commit', etc. On the left, a table lists repository contents by type (e.g., AVAS, TABL, CLAS) and name, along with their paths. On the right, there's a sidebar with 'diff' status indicators for each item.

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_KUNABLAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_kunablaz_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.

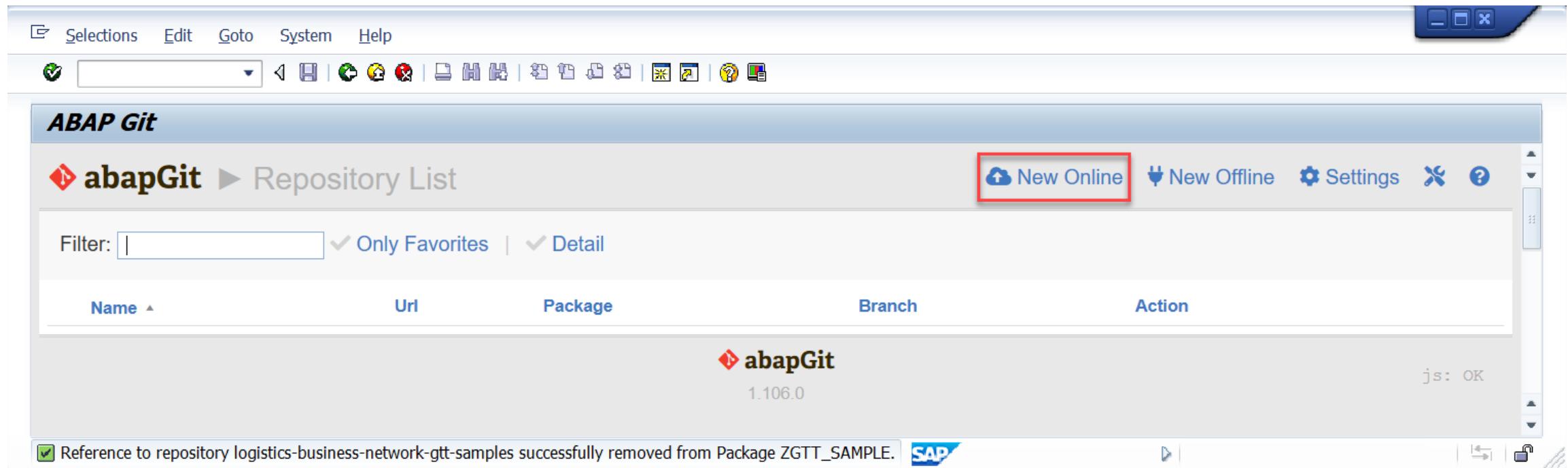
The screenshot shows the ABAP Git interface for managing repositories. The main window displays a list of files under the repository 'logistics-business-network-gtt-samples'. A modal dialog box titled 'Remove' is open over the list, containing a warning message: 'This will remove the repository reference to the package ZGTT\_SAMPLE. All objects will safely remain in the system.' At the bottom of this dialog are two buttons: 'Remove' (highlighted with a red box) and 'Cancel'.

Type	Name	Path	diff
AVAS	0894	abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	M
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	M
TABL	ZGTT_SOF_EE_REL	/lbn-gtt-template-tso/abap/zsrc/zggt_sof_ee_rel.tabl.xml	M
TABL	ZGTT_STOP_INFO	/lbn-gtt-template-tso/abap/zsrc/zggt_stop_info.tabl.xml	M
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	M
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	M

At the bottom of the screen, a status message reads: 'Reference to repository logistics-business-network-gtt-samples successfully removed from Package ZGTT\_SAMPLE.' There is also a SAP logo.

# 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>

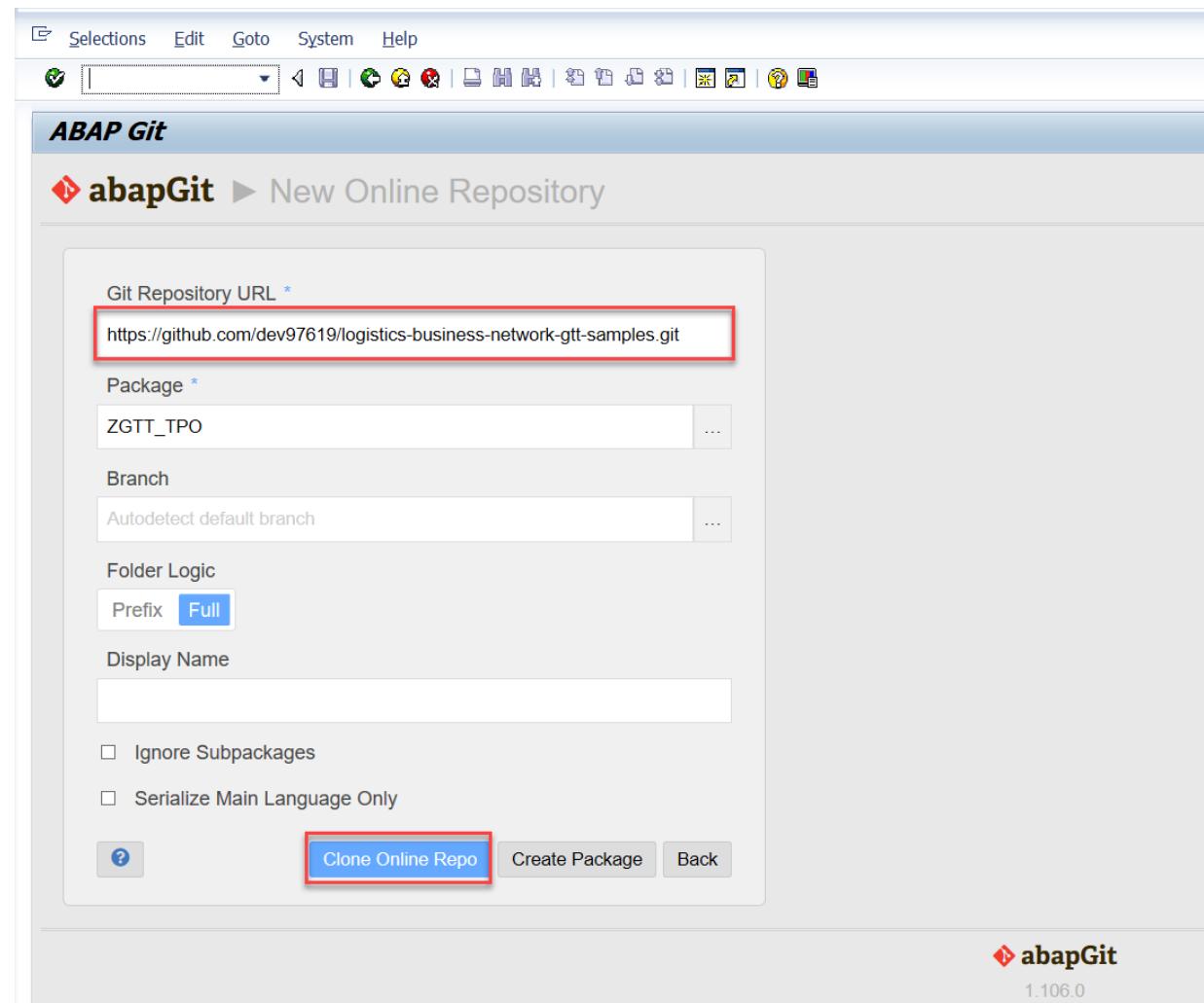
## Caution:

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.

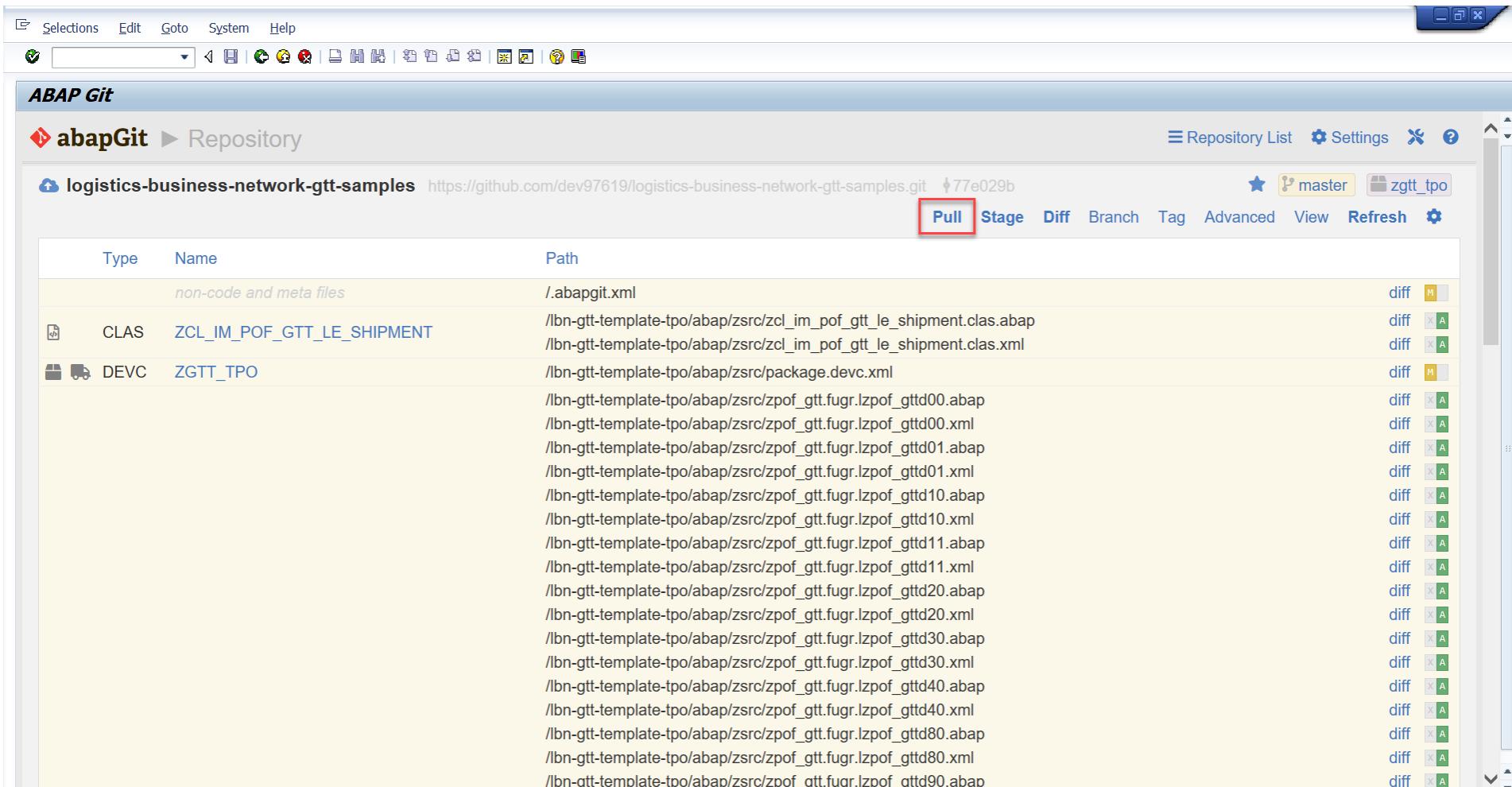
4-4: Set **Full** for **Folder Logic**

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

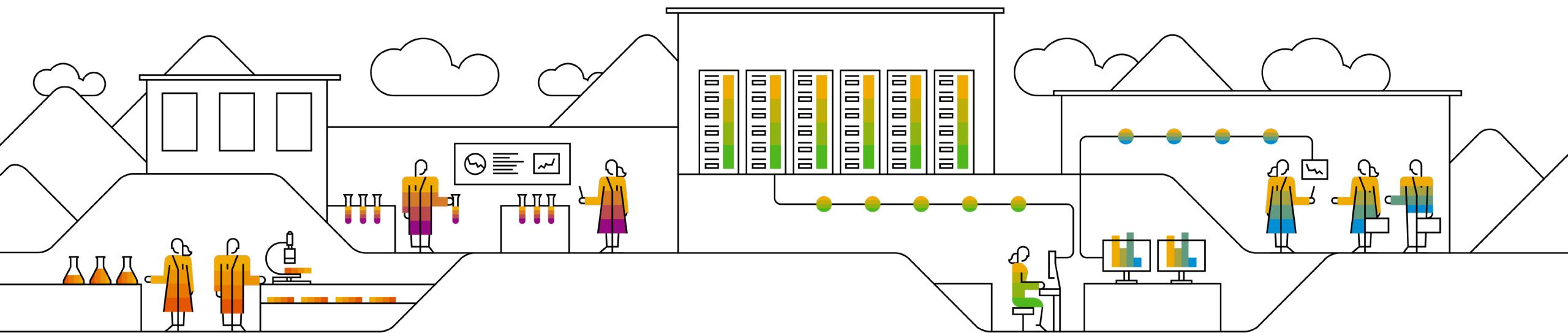


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation "abapGit > Repository". The main area displays a list of files under the repository "logistics-business-network-gtt-samples" with the URL "https://github.com/dev97619/logistics-business-network-gtt-samples.git". The commit hash "77e029b" is shown above the file list. A toolbar at the top right includes "Repository List", "Settings", "Pull" (which is highlighted with a red box), "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The file list table has columns for Type, Name, Path, and Diff. It lists several ABAP classes and XML files under the ZGTT\_TPO folder, each with a "diff" button next to it.

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

# 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

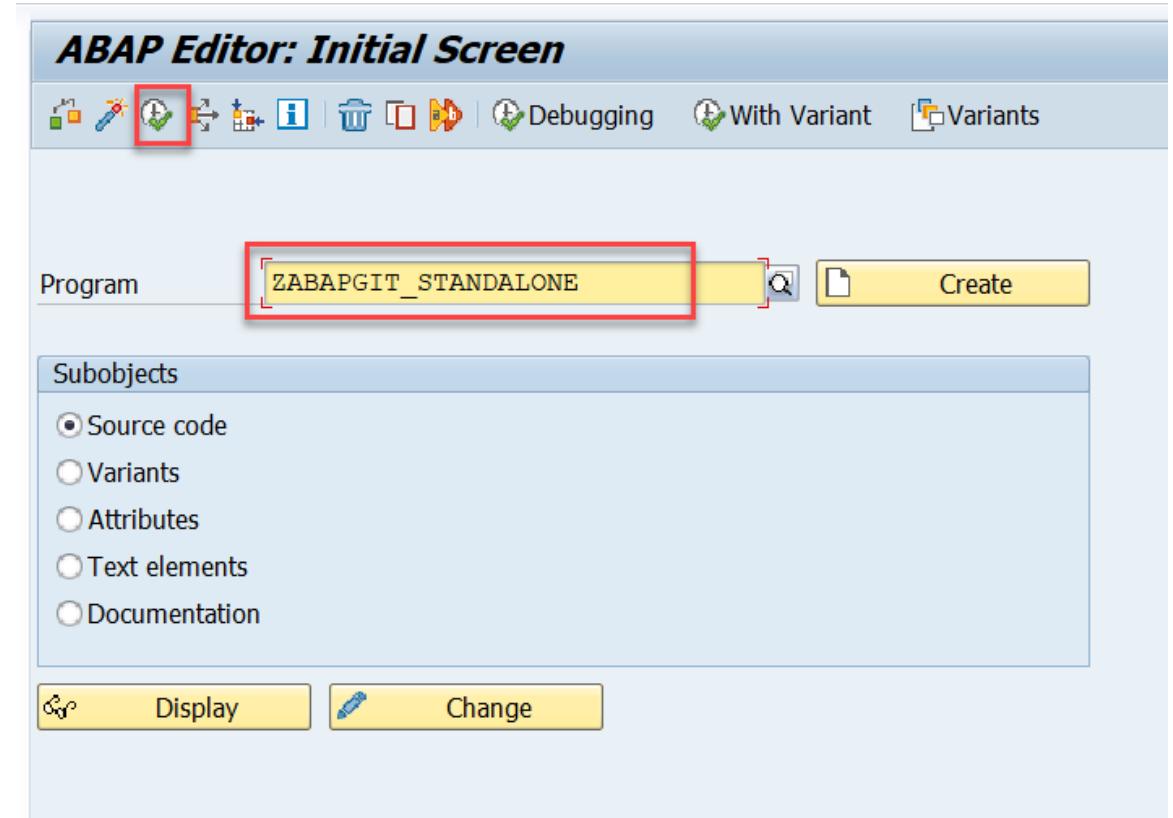
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `:$ZABAPGIT`).

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

## STEP 2: Download ABAP Code

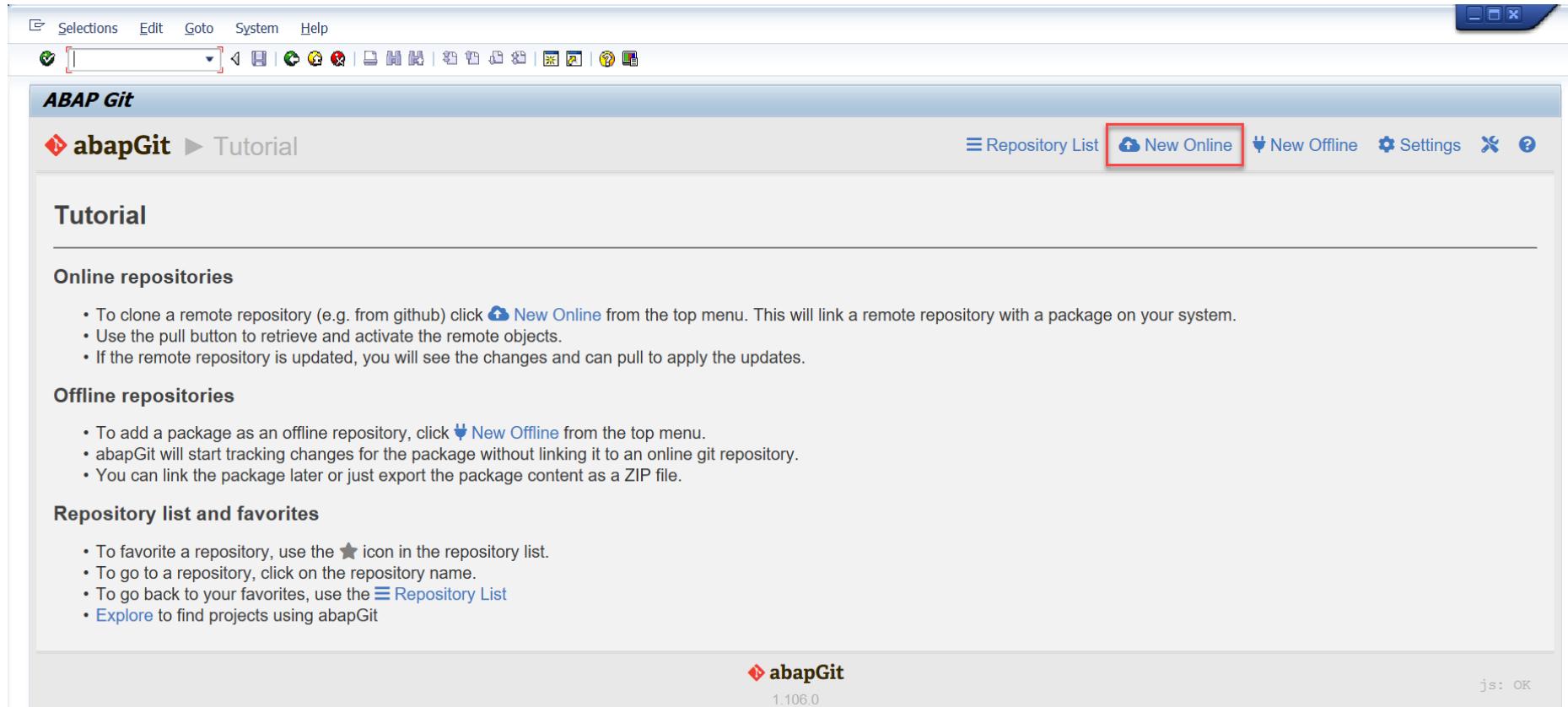
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,  
**ZABAPGIT\_STANDALONE**

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

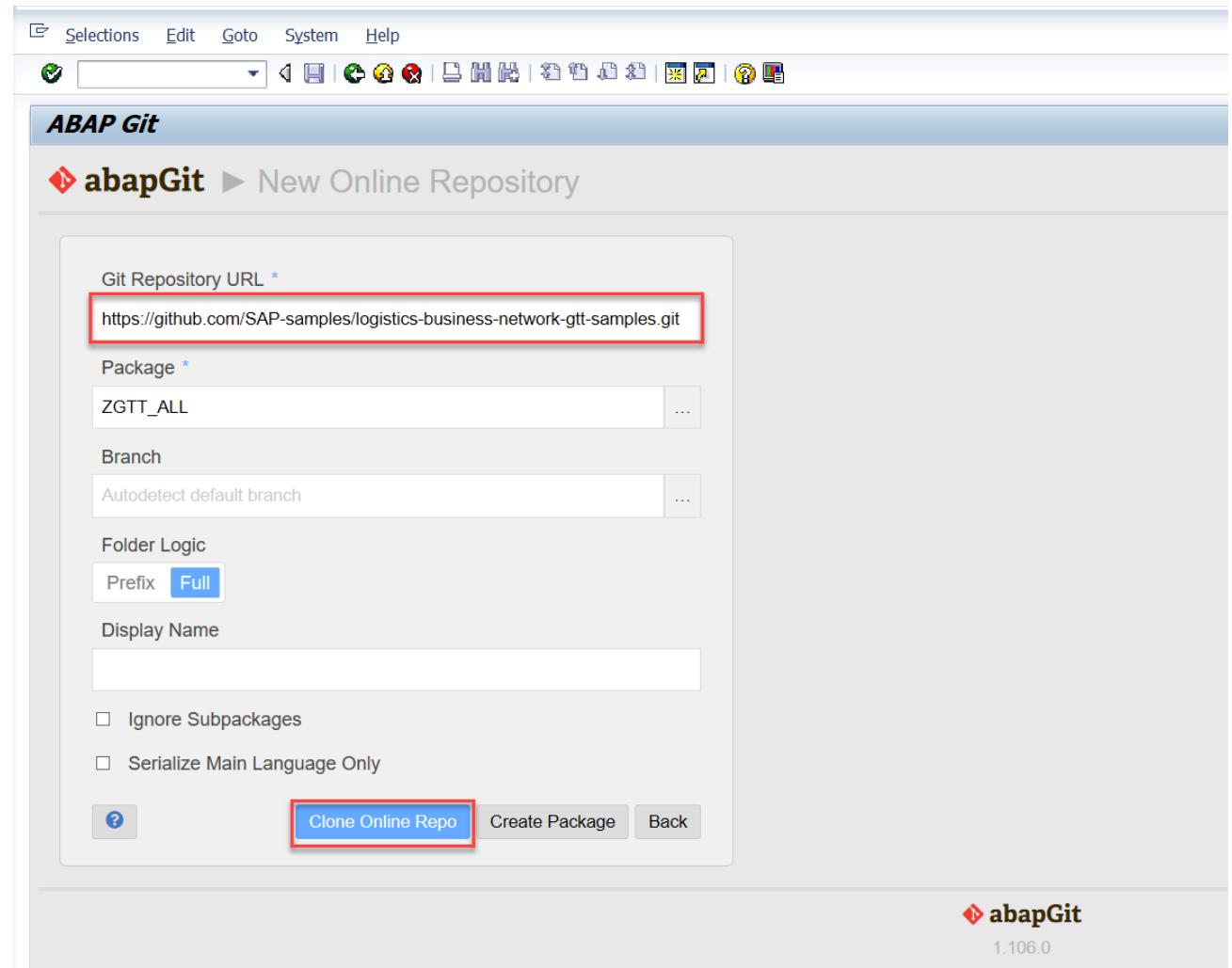
2-4: Fill in the **Git repository URL**:

<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.

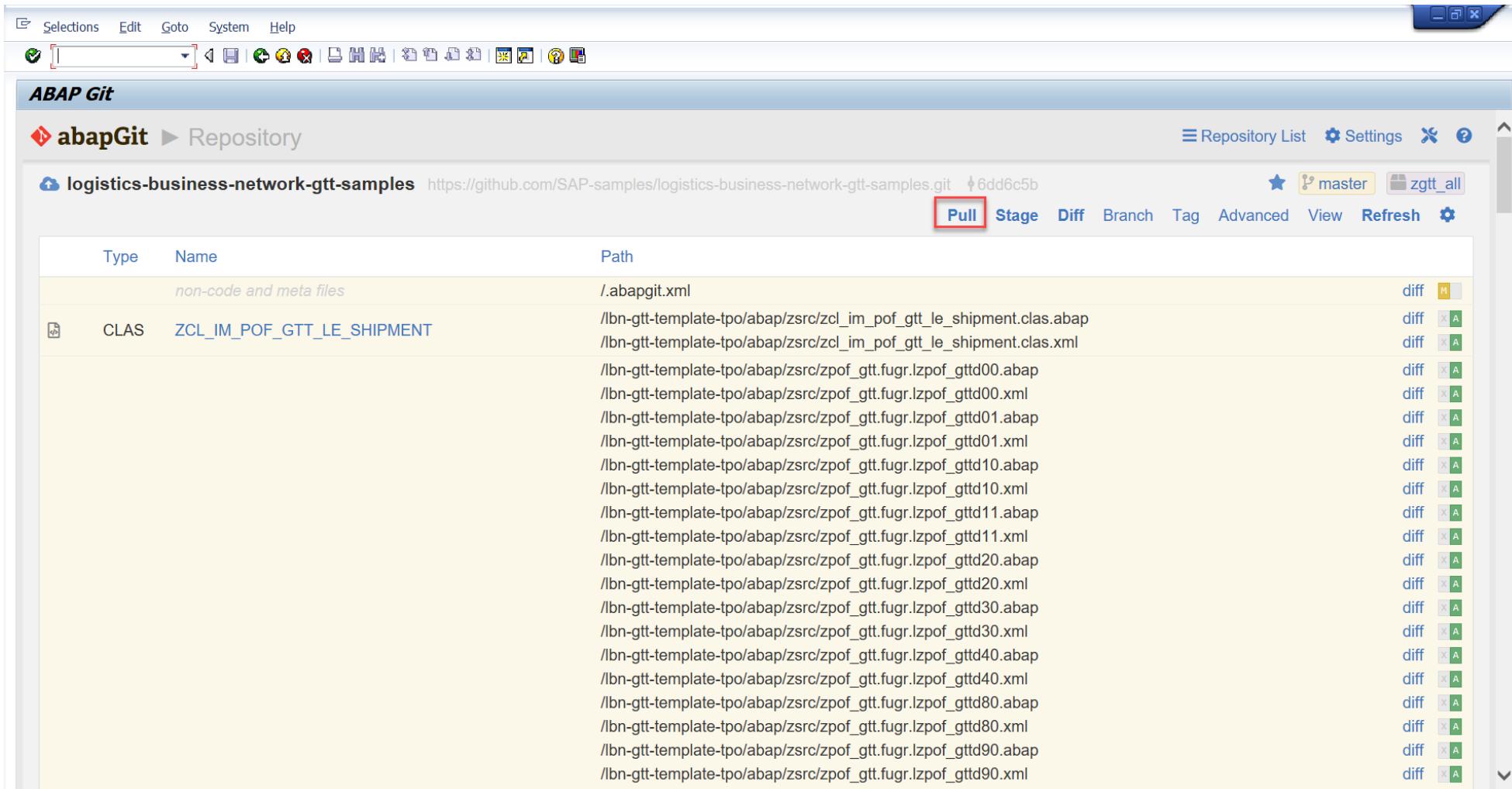
2-6: Set *Full* for **Folder Logic**

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 a breadcrumb navigation: "abapGit > Repository". Underneath is a sub-header for the repository "logistics-business-network-gtt-samples" with its URL and a commit hash. A toolbar below the sub-header includes buttons for "Pull", "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and settings. 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 absolute paths for each file. To the right of the table, there are "diff" buttons and status indicators (M, A, C) for each file.

Type	Name	Path	
		.abapgit.xml	diff M
		/bn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	diff A
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.xml	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.abap	diff A
		/bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.xml	diff 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 table of files and their paths. A red box highlights two rows: one for a class named "ZCL\_GTT\_SOFTWARE\_SHIPPING" and another for an enhancement named "Z\_GTT\_SOFT\_SOFTWARE\_SHIPPING\_DELIVERY\_PROC". Both rows have a "diff" button next to them, which is highlighted with a red box. The "diff" button for the class row shows a green "A" and a red "X", indicating a deletion. The "diff" button for the enhancement row also shows a green "A" and a red "X".

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_software_im_le_shipping.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_software_im_le_shipping.clas.xml	[diff] [A X]
ENHO	Z_GTT_SOFT_SOFTWARE_SHIPPING_DELIVERY_PROC	/lbn-gtt-template-tso/abap/zsrc/z_gtt_software_shp_delivery_proc.enho.xml	[diff] [A X]

# **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.

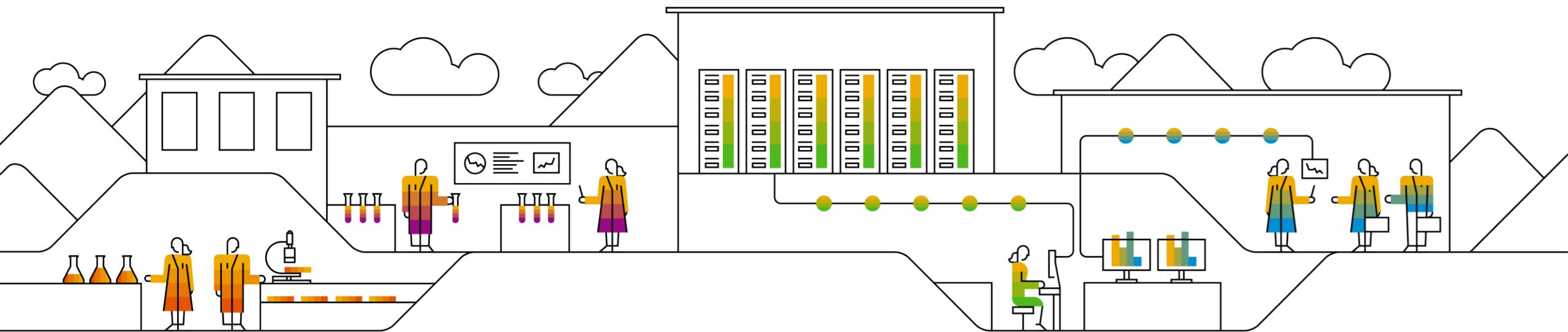
# Known Issue: Remote Deleted Object Cannot be Synchronized to the Local Object

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 a radio button for "New BAdI" selected. An input field for "Enhancement implementation" contains the value "Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC", which is highlighted with a red box. Below this, there's another radio button for "Classic BAdI" and an input field for "Implementation". At the bottom of this section are "Display" and "Change" buttons. To the right, a detailed view of the "Enhancement Implementation Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC Display" is shown. It includes tabs for "Properties", "History", "Technical Details", and "Implementation Elements". Under "Implementation Elements", there's a table with a single row for "BAdI Implementations". The row shows "Z\_GTT\_SOF\_IM\_LE\_SHIPPING" as the implementation and "Implementing Class" as the description. In the "Runtime Behavior" section, there's a checkbox for "Implementation is active" which is unchecked, and a note below it says "The implementation will not be called".

# 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 components:

- Define Application Object Types Dialog:** On the left, under "Bus. Proc. Type" TMS\_TOR, the "Appl. Obj. Type" field is set to ZGTT\_SHP\_ACC\_HD. This field is highlighted with a red box.
- Model Details View:** On the right, under the "SOF" tab, the "Tracked Process" is set to "Shipment". The "Application Object Type" field is also set to ZGTT\_SHP\_ACC\_HD and is highlighted with a red box.

Both views show the same configuration, ensuring consistency between the AOT definition and the corresponding model in the SAP Business Network.

## 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:

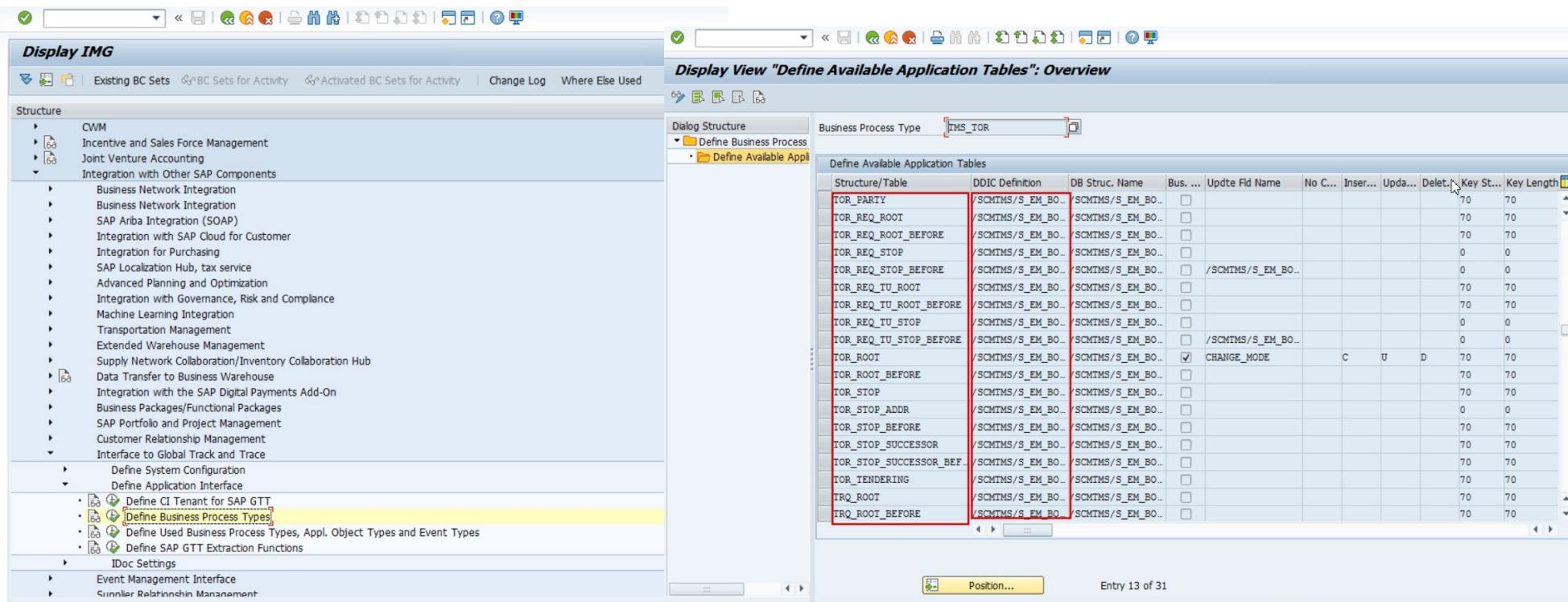
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HDR	Function for control parameters of Freight Order and Freight Booking
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL	Actual Event of Arrival
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_COUPLING	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 displays two SAP application screens side-by-side.

**Left Screen (Display IMG):** This screen shows the SAP IMG navigation structure. The path 'Define Application Interface > Define Business Process Types' is highlighted with a yellow background and a red border around the 'Define Business Process Types' node. Other nodes like 'Define System Configuration' and 'Define Used Business Process Types' are also visible.

**Right Screen (Display View "Define Available Application Tables": Overview):** This screen shows a table of application tables defined for the business process type 'TMS\_TOR'. The table has columns for Structure/Table, DDIC Definition, DB Struc. Name, Bus..., Update Fld Name, No C..., Insert..., Upda..., Delete..., Key St..., and Key Length. The 'Structure/Table' column lists various TOR\_\* 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, TOR\_ROOT, and TOR\_ROOT\_BEFORE. The 'DDIC Definition' column shows the corresponding SCNTMS/S\_EM\_BO... table for each.

# 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 the SAP ABAP Development Workbench interface with two main windows:

- Function Builder: Display ZSST\_GTT\_OTE\_FO\_HDR\_REL**: This window shows the source code for the function module ZSST\_GTT\_OTE\_FO\_HDR\_REL. The code includes declarations for data types like `lt_app_objects`, `lo_udm_message`, and `ls_bapiret`. It features a `TRY` block handling exceptions and performing relevance checks. The code is annotated with SAP-specific comments and annotations.
- ABAP Editor: Display Include LZSST\_GTT\_D20**: This window shows the source code for the include module LZSST\_GTT\_D20. It contains methods such as `lif_bo_reader~check_relevance` and `lif_ef_constants`. The code uses various SAP constructs like `ASSIGN`, `MESSAGE`, and `lcl_tools>throw_exception`. The code is also annotated with SAP-specific comments and annotations.

# 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 *ACT/ON* 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 the ABAP code for handling tracking IDs, including imports, exports, and local variables like *lo\_udm\_message* and *ls\_bapiret*.

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

19 DATA: lo_udm_message TYPE REF TO cx_udm_message,
20 ls_bapiret TYPE bapiret2.
21
22 TRY.
23   lcl_ef_performer->get_track_id_data(
24     EXPORTING
25       is_definition = VALUE #(          maintab = lif_sst_constants->cs_tabledesc-fo_header_new )
26       io_bo_factory = NEW lcl_bo_factory( )          i_appsyst
27       iv_appsyst
28       is_app_obj_types = i_app_obj_types
29       it_all_appl_tables = i_all_appl_tables
30       it_app_type_cntl_tabs = i_app_type_cntl_tabs
31       it_app_objects = i_app_objects
32     IMPORTING
33       et_track_id_data = e_trackidata[]
34   ).          i_appsyst
35
36   CATCH cx_udm_message INTO lo_udm_message.
37   lcl_tools->get_errors_log( )
38   EXPORTING
39     io_udm_message = lo_udm_message
40     iv_appsyst = i_appsyst
41   IMPORTING
42     es_bapiret = ls_bapiret .
43
44   " add error message
45
Scope: |FUNCTION ZSST_GTT_OTE_FO_HEADER_TID|TRY ABAP Ln 38 Col 33

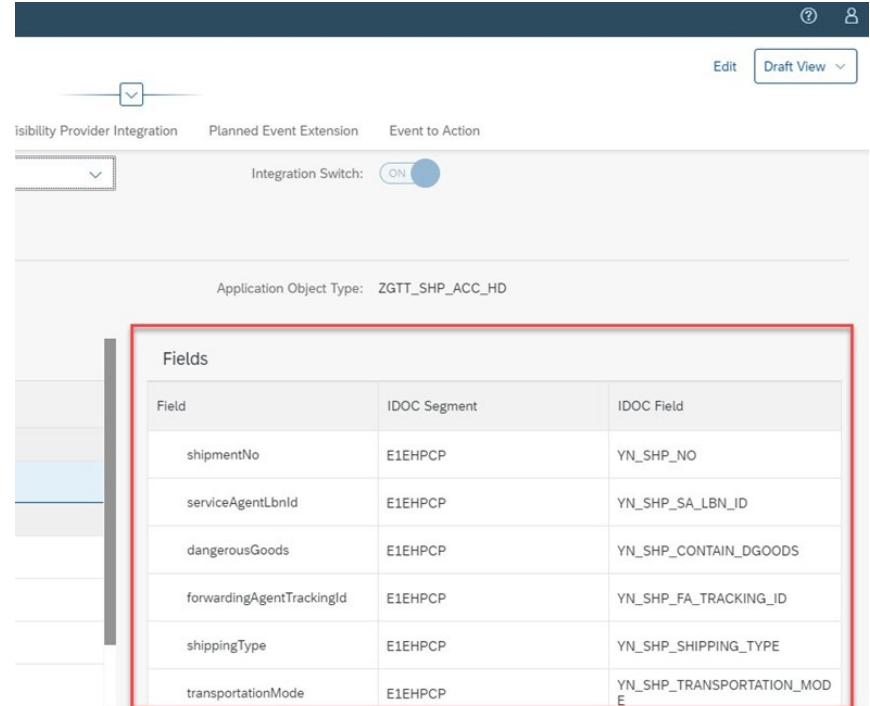
ABAP Editor: Display Include LCL_BO_FREIGHT_D20
Include LCL_BO_FREIGHT_D20 Active
METHOD lif_bo_reader~get_track_id_data.
1614 DATA: lr_item TYPE REF TO data,
1615 lr_item_old TYPE REF TO data,
1616 lt_track_id_data TYPE lif_ef_types->tt_enh_track_id_data,
1617 lt_track_id_data_old TYPE lif_ef_types->tt_enh_track_id_data,
1618 lr_root_new TYPE REF TO data,
1619 lr_root_old TYPE REF TO data.
1620
1621 FIELD-SYMBOLS: <lt_item> TYPE ANY TABLE,
1622 <lt_item_old> TYPE ANY TABLE,
1623 <ls_root> TYPE /smmtms/s_em_bo_tor_root,
1624 <lt_root_new> TYPE /smmtms/t_em_bo_tor_root,
1625 <lt_root_old> TYPE /smmtms/t_em_bo_tor_root.
1626
1627 ASSIGN is_app_object-maintabref->* TO <ls_root>.
1628 IF sy-subrc <> 0.
1629   RETURN.
1630 ENDIF.
1631
1632 lr_root_new = mo_ef_parameters->get_appl_table(
1633   iv_tabledef = lif_sst_constants->cs_tabledesc-fo_header_new).
1634
1635 lr_root_old = mo_ef_parameters->get_appl_table(
1636   iv_tabledef = lif_sst_constants->cs_tabledesc-fo_header_old).
1637
1638 ASSIGN lr_root_new->* TO <lt_root_new>.
1639 IF sy-subrc <> 0.
1640   RETURN.
1641 ENDIF.
1642
1643
1644 Scope: |CLASS lcl_bo_freight_booking_reader|METHOD lif_bo_reader~get_trac... ABAP Ln 1636 Col 76
```

# 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 need 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 section labeled 'Integration Switch' with a 'ON' button. Further down, the 'Application Object Type' is set to '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.

```

Scope: \CLASS lcl\_bo\_freight\_order\_reader\METHOD lif\_bo\_reader~get\_data | ABAP

**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.

```

Scope: \FUNCTION ZSST\_GTT\_OTE\_FO\_HDR\TRY | ABAP

# 8: Coding Tips in the Planned Event Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E\_EXPEVENTDATA*.
3. 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 with the 'Tracked Process' tab selected. The tracked process is set to 'Shipment'. The 'Tracked Process Mapping' section shows 'ERP Object Type: Others'. The 'Tracked Process / Events (26)' table is highlighted with a red border and lists the following data:

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

# 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_sts_constants->cs_tabledef-fo_stop_new).

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

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

```

Scope: \CLASS lcl\_pe\_filler\_fo\_header\METHOD lif\_pe\_filler~get\_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_sts_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 an implementation of the `get_postal_address` method. It uses several IF statements to handle different cases based on the input parameters `lt_stop_target_key`, `lt_loc_log_key_link`, and `lt_address_key_link`. Each case involves retrieving data from service managers and performing associations. The code also includes error handling with TRY, CATCH, and MESSAGE statements.

```
Include LZSST_GTTD10 Active

540 METHOD get_postal_address.
541   DATA(lo_tor_srv_mngr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_tor_c=>sc_bo_key).
542   DATA(lo_loc_srv_mngr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
543
544   lo_tor_srv_mngr->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_mngr->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_mngr->retrieve_by_association(
563       EXPORTING
564         iv_node_key      = /scmtms/if_location_c=>sc_node-root
565         it_key           = CORRESPONDING #( lt_loc_log_key_link MAPPING key = target_key )
566         iv_association  = /scmtms/if_location_c=>sc_association-root-address
567       IMPORTING
568         et_key_link     = DATA(lt_address_key_link).
569
570   IF lt_address_key_link IS NOT INITIAL.
571     TRY.
572       DATA(lr_bo_conf) = /bobf/cl_frw_factory->get_configuration(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
573       CATCH /bobf/cx_frw.
574         MESSAGE e011(zsst_gtt) INTO DATA(lv_dummy).
575         lcl_tools->throw_exception( ).
576     ENDTRY.
577
578     DATA(lv_postal_ass_key) = lr_bo_conf->get_content_key_mapping(
579       iv_content_cat    = /bobf/if_conf_c=>sc_content_ass
580       iv_do_content_key = /bofu/if_addr_constants=>sc_association-root-postal_address
581       iv_do_root_node_key = /scmtms/if_location_c=>sc_node-/bofu/address ).
582
```

# 9: Coding Tips in the Event Data Function Modules

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

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

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

The screenshot shows the SAP Model Details interface for a tracked process named 'Shipment'. The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process / Events' section is highlighted with a red border. It lists 26 events, each mapping a tracked process name to an IDOC and an event code. The events listed are: LoadingStart (E1EVMHDR02, LOAD\_BEGIN), POD (E1EVMHDR02, POD), Departure (E1EVMHDR02, DEPARTURE), Arrival (E1EVMHDR02, ARRIV\_DEST), and LoadingEnd (E1EVMHDR02, LOAD\_END).

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

# 9: Coding Tips in the Event Data Function Modules

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

```

CALL FUNCTION '/SCMTMS/EXTR_EVT_TO_ARRIVAL'
EXPORTING
  i_applsys      = i_applsys
  i_event_type    = i_event_type
  i_all_appl_tables = i_all_appl_tables
  i_event_type_cntl_tabs = i_event_type_cntl_tabs
  i_events        = i_events
TABLES
  ct_trackingheader = ct_trackingheader
  ct_tracklocation  = ct_tracklocation
  ct_trackaddress   = ct_trackaddress
  ct_trackparameters = ct_trackparameters
CHANGING
  c_eventid_map    = c_eventid_map
EXCEPTIONS
  parameter_error  = 1
  event_data_error = 2
  stop_processing   = 3
  OTHERS            = 4.
CASE sy-subrc.
  WHEN 1.
    RAISE parameter_error.
  WHEN 2.
    RAISE event_data_error.
  WHEN 3.
    RAISE stop_processing.
  WHEN 4.
    RAISE stop_processing.

```

Scope: \FUNCTION zsst\_gtt\_ee\_fo\_arrival\ CASE ABAP

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

```

FUNCTION zsst_gtt_ee_fo_arrival_rel.
  *--"Local Interface:
  IMPORTING
    i_all_appl_tables TYPE /SAPTRX/APPLSYSTEM
    iv_event_code      = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
    i_event            = i_event
  IMPORTING
    e_result          = e_result .
  CATCH cx_udm_message INTO DATA(lo_udm_message).

```

Scope: \FUNCTION zsst\_gtt\_ee\_fo\_arrival\_rel\ TRY ABAP | Ln 27 Col 52

# 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.