



# SAP Business Network Global Track and Trace

## Track Sales Orders - Deep Dive with 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



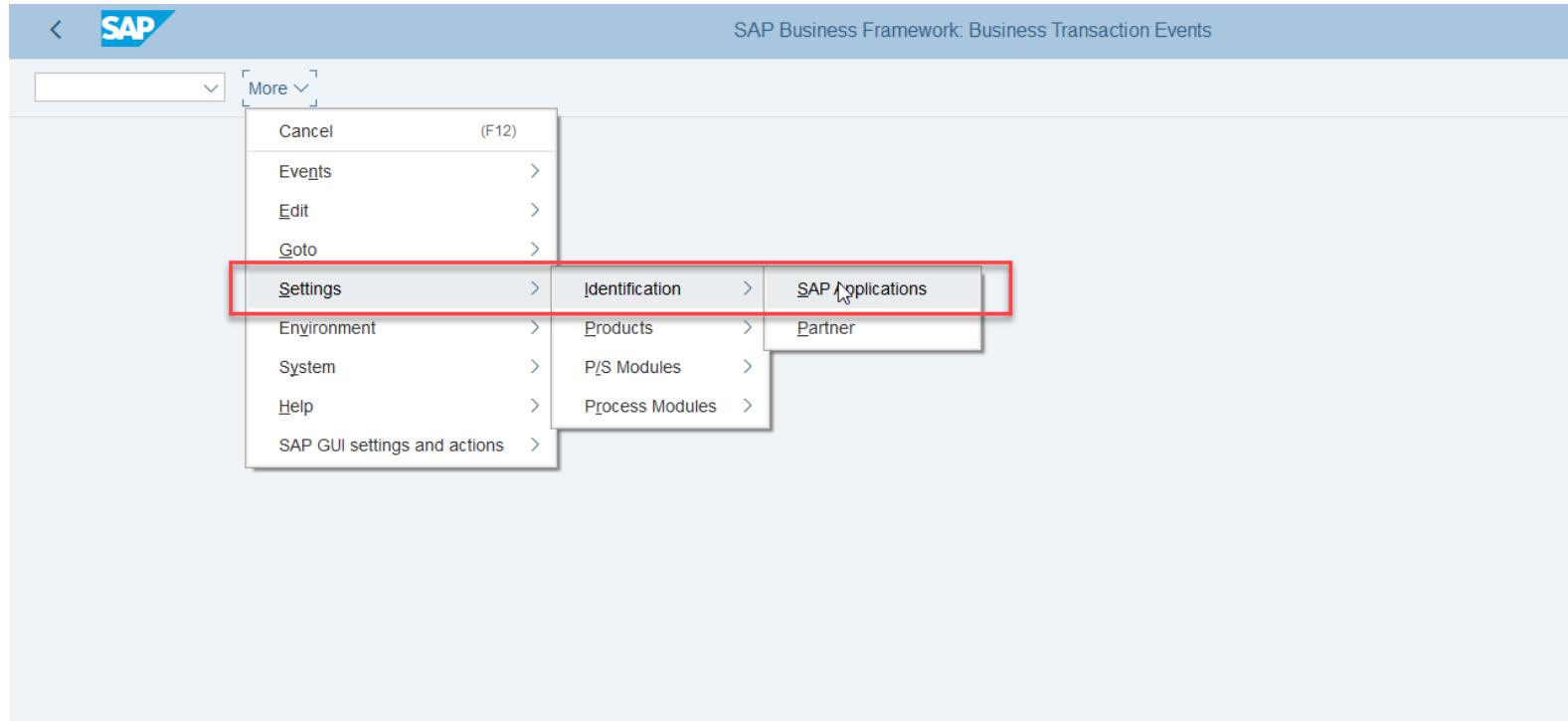
# STEP 1: Check the SAP Product Version

1-1: Make sure that you have met the requirements for the product version mentioned in the “[the Prerequisites](#)” section of *Appendix one: Connect to SAP ERP in Administration Guide for Version 2*. You can find this guide at <http://help.sap.com/gtt>.

1-2 : The ABAP codes on Github to support sample apps for SAP Business Network Global Track and Trace shall be implemented in SAP S/4HANA 1909 SP03 on premise or higher. Please note that the codes are not validated in its lower version or other ECC series of products, so you might need to do further adaptation work or build your own extractor.

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

- 2-1: Ensure you have development access to the client for cross-client customizing and local development
- 2-2: Log on to the client and enter transaction code (T-code): **FIBF**
- 2-3: Click **More -> Settings -> Identification -> SAP Applications**



# STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**

The screenshot shows a SAP application interface titled "Change View 'BTE Application Indicator'. Overview". The main area is a table with three columns: "Appl.", "A", and "Text". The "Appl." column lists various application codes, and the "Text" column provides a brief description of each. The row for "PI-EM" is highlighted with a red box around the "Appl." cell. In the "A" column for "PI-EM", there is a checkbox that is checked, indicating the application is active. Other applications listed include PM, PM-BW, PM-EQM, PM-PAM, PMA-PC, PMAT, PMIPUR, PMPUSH, PP-BD, PP-DD, PP-MRP, PRICAT, PS-REP, PSRV, QBEXT, QBEXTP, QILPO, RDSVFI, and RDSVMD. The bottom right of the screen shows a dark footer bar with "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
PMPUSH	<input type="checkbox"/>	MAM Push
PP-BD	<input checked="" type="checkbox"/>	Production Planning MasterData
PP-DD	<input checked="" type="checkbox"/>	Demand Driven Replenishment
PP-MRP	<input checked="" type="checkbox"/>	Material Requirements Planning
PRICAT	<input type="checkbox"/>	Preiskatalog
PS-REP	<input checked="" type="checkbox"/>	Projektsystem
PSRV	<input checked="" type="checkbox"/>	Produkt - Service
QBEXT	<input checked="" type="checkbox"/>	External Inspection Procurement
QBEXTP	<input checked="" type="checkbox"/>	External Inspection Production
QILPO	<input checked="" type="checkbox"/>	Inspection Lot Order Integr.
RDSVFI	<input type="checkbox"/>	Dgtl.Signature Validation FI
RDSVMD	<input checked="" type="checkbox"/>	Dgtl.Signature BP Check

# B) Configuration and Implementation

- Basic

## B1. IDOC Configuration



# STEP 1: Define RFC Connection for 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

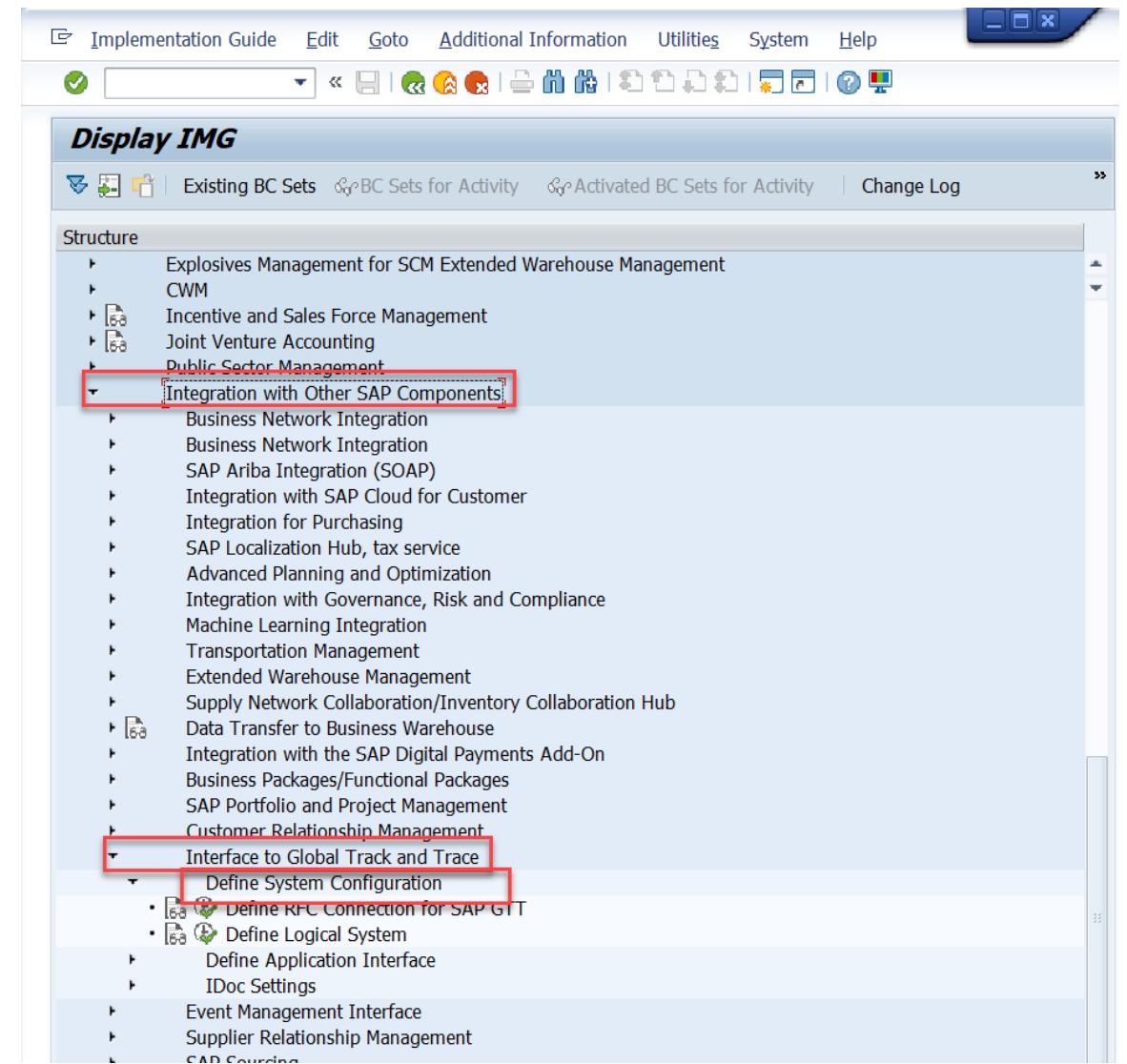
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

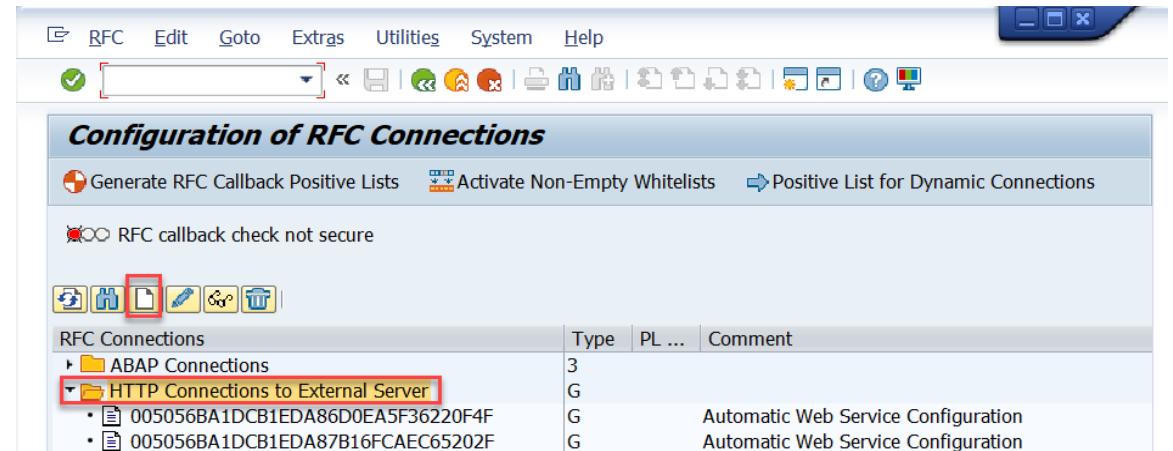
**Define RFC Connection for SAP GTT**



# 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 follows :

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

**Host:** `xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com`

**Port:** `443`

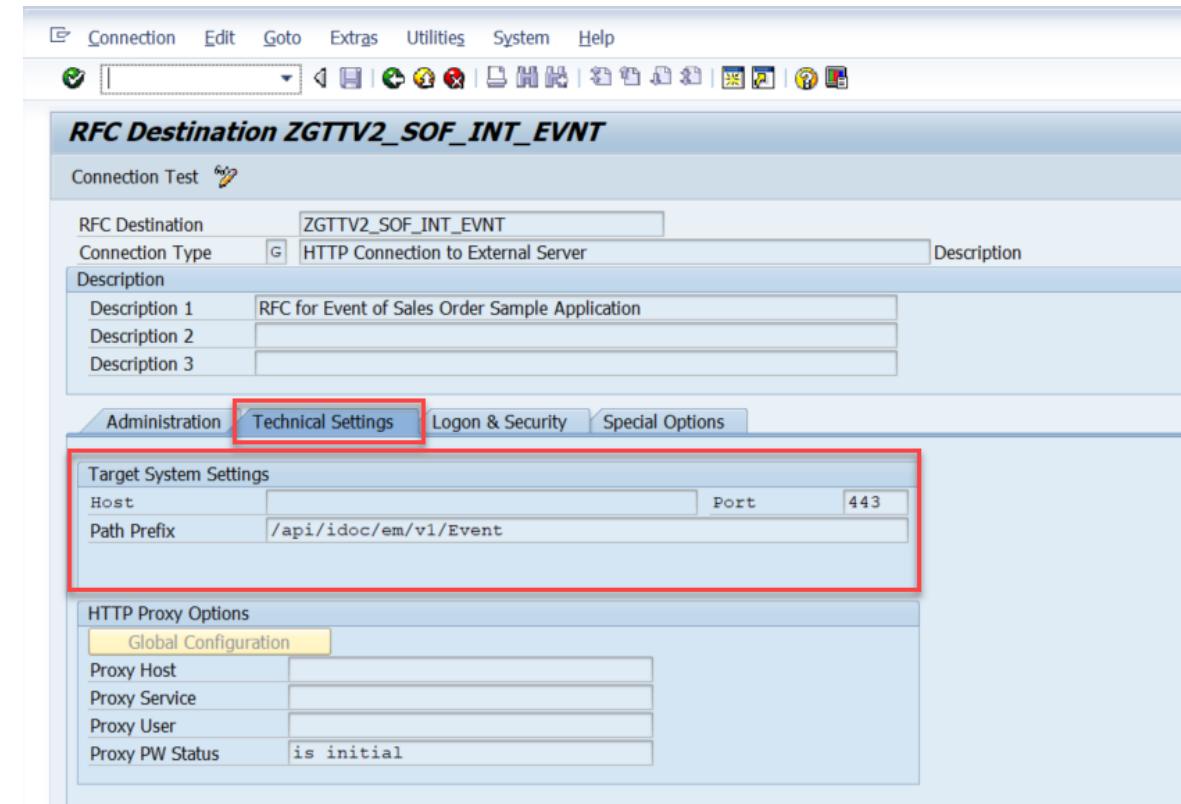
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

For the event:

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

For the tracked Process:

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



RFC Destination	RFC Destination Description	Host	Path Prefix	Port
ZGTTV2_SOF_INT_EVNT	RFC for Event of Sales Order Sample Application	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/Event	443
ZGTTV2_SOF_INT_TP	RFC for Tracked Process of Sales Order 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.

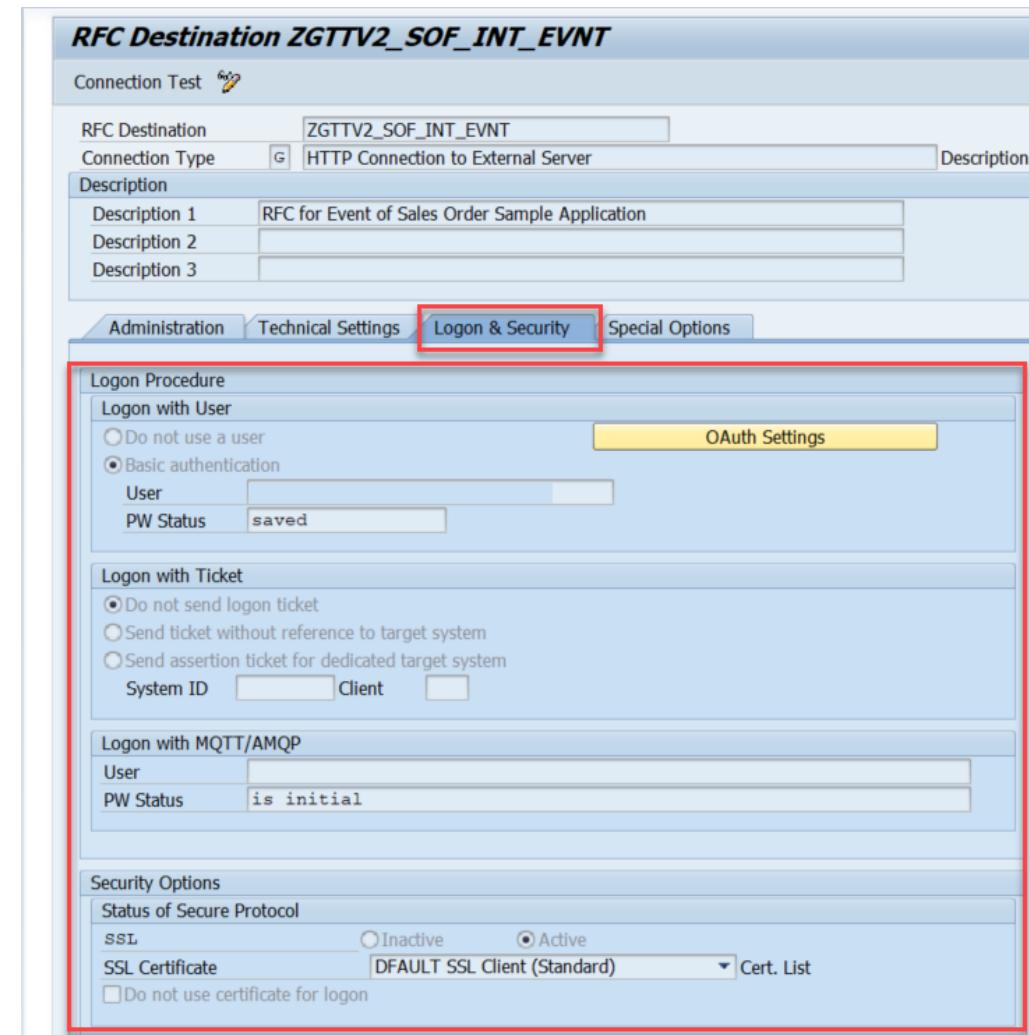
Also, SSL must be *Active*.

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

1-10: Save the configuration

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

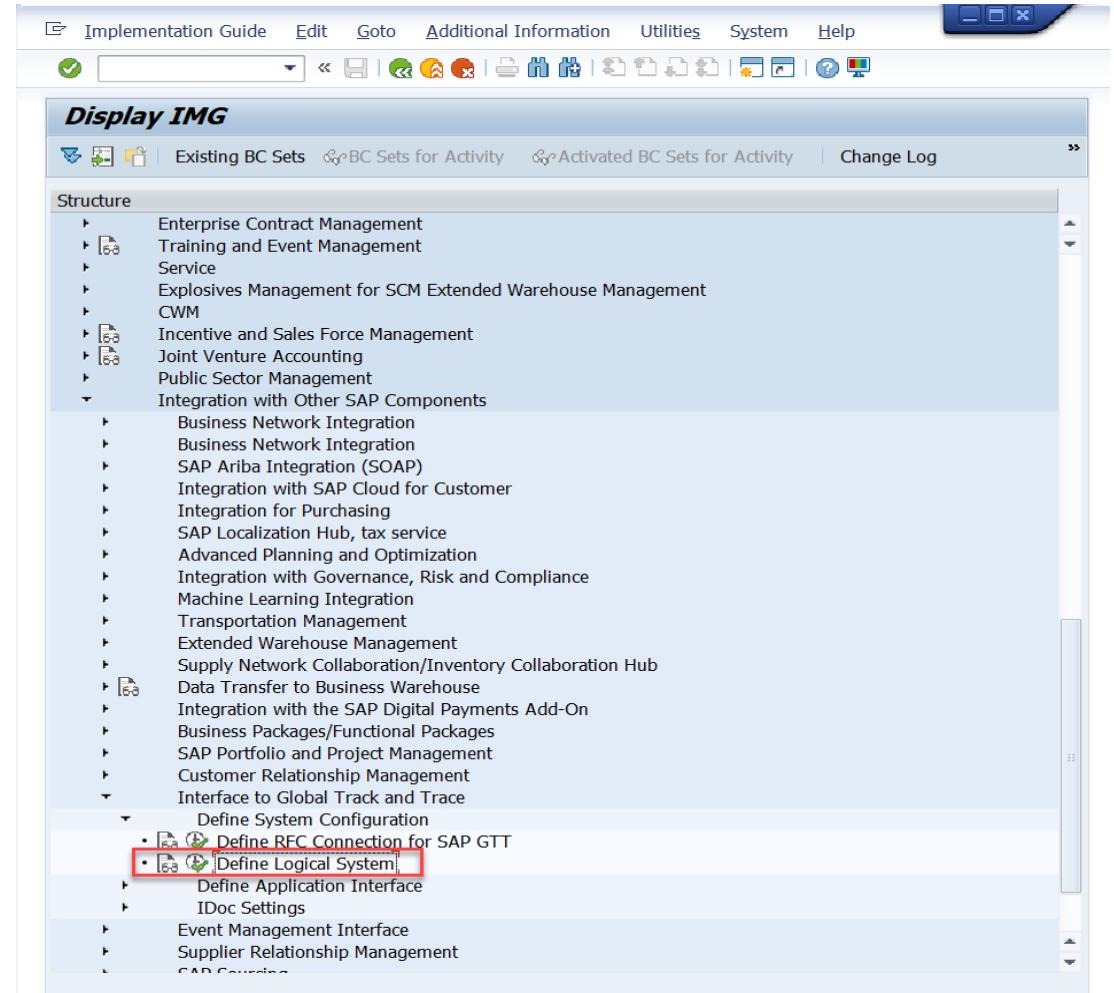
- one for event and
- the other for tracked process.



# 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**.

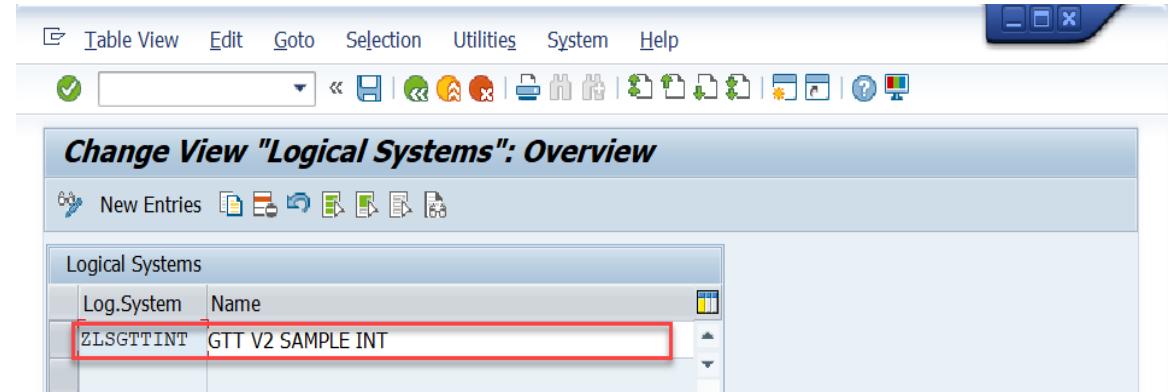


## STEP 2: Define Logical System

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

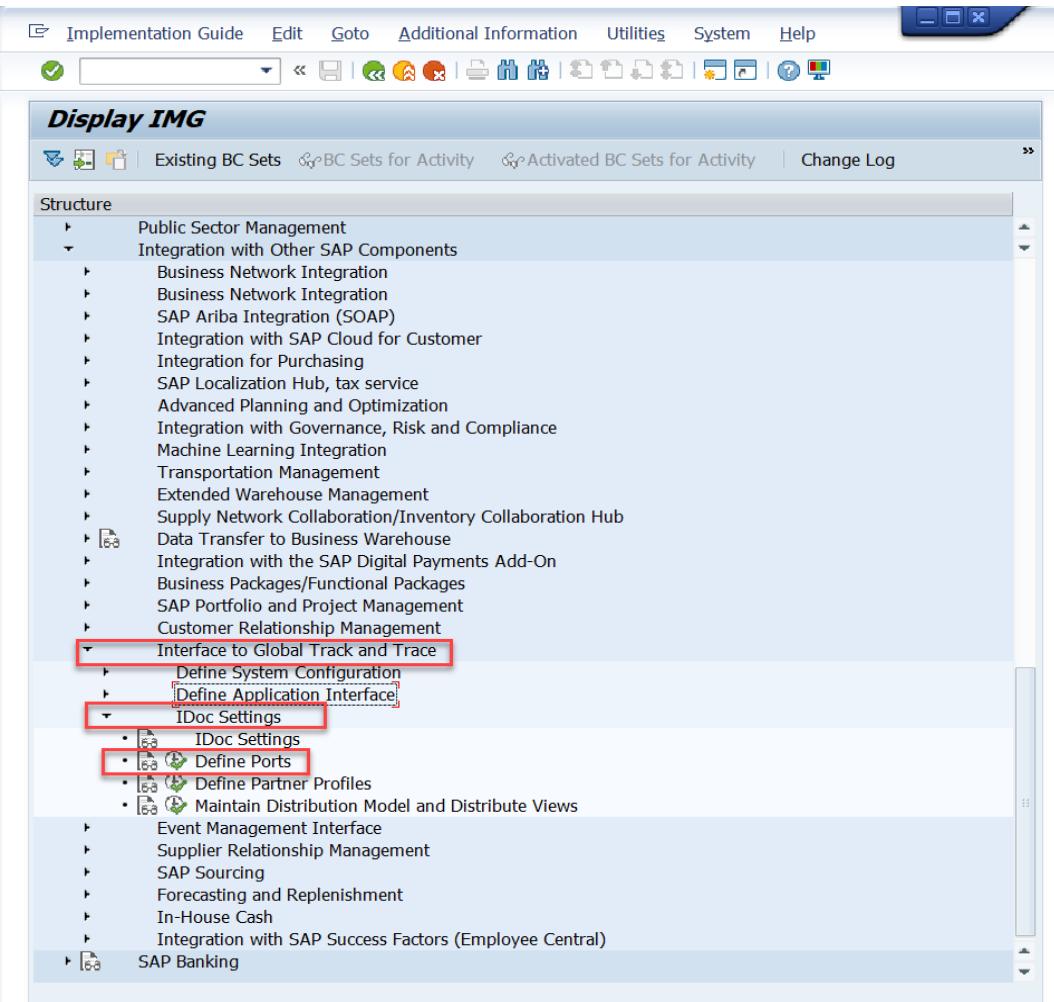


Log.System	Name
ZLSGTTINT	GTT V2 SAMPLE INT

# STEP 3: Define Ports

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

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



## STEP 3: Define Ports

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

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

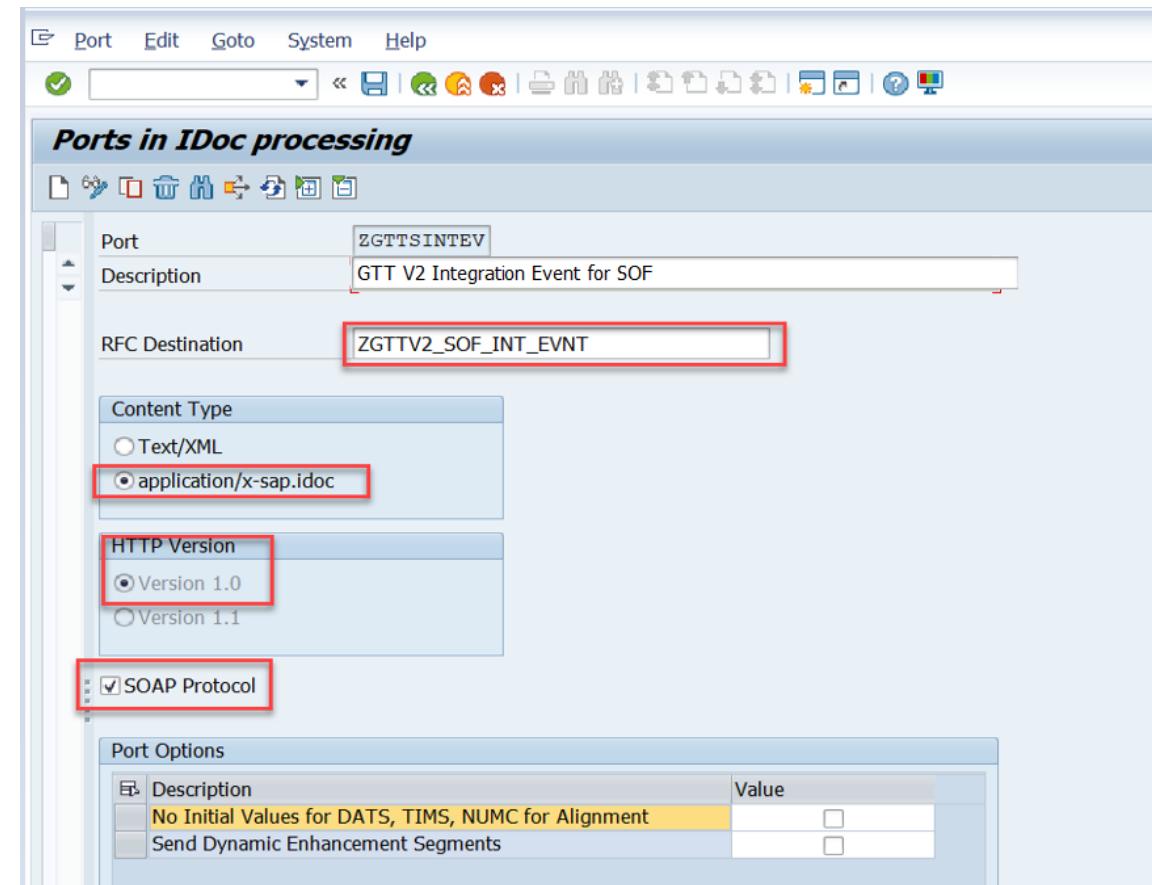
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

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

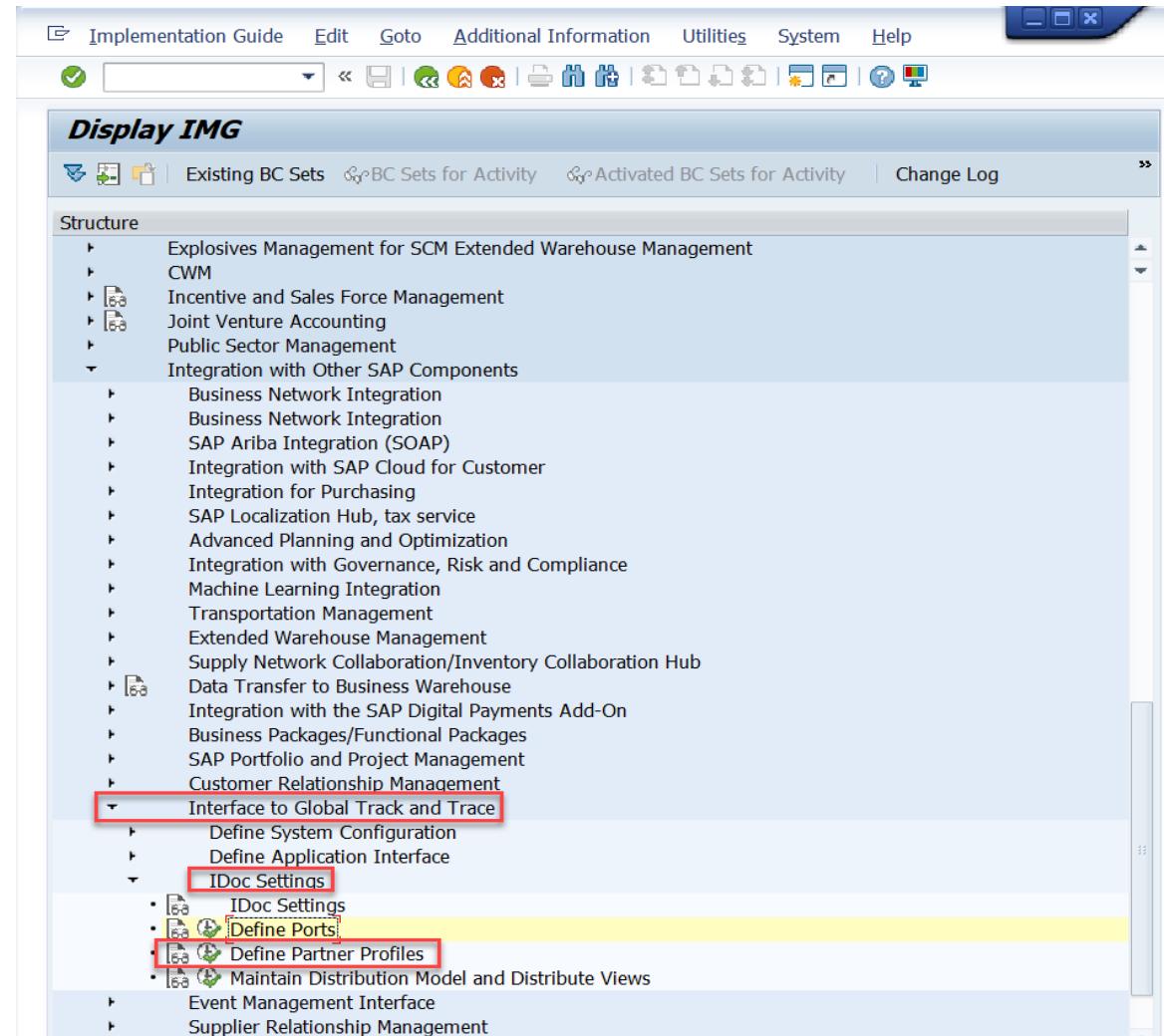


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSINTEV	GTT V2 Integration Event for SOF	ZGTTV2_SOF_INT_EVNT	application/x-sap.idoc	Version 1.0	Checked
ZGTTSINTTP	GTT V2 Integration Tracked Process for SOF	ZGTTV2_SOF_INT_TP	application/x-sap.idoc	Version 1.0	Checked

# STEP 4: Define Partner Profiles

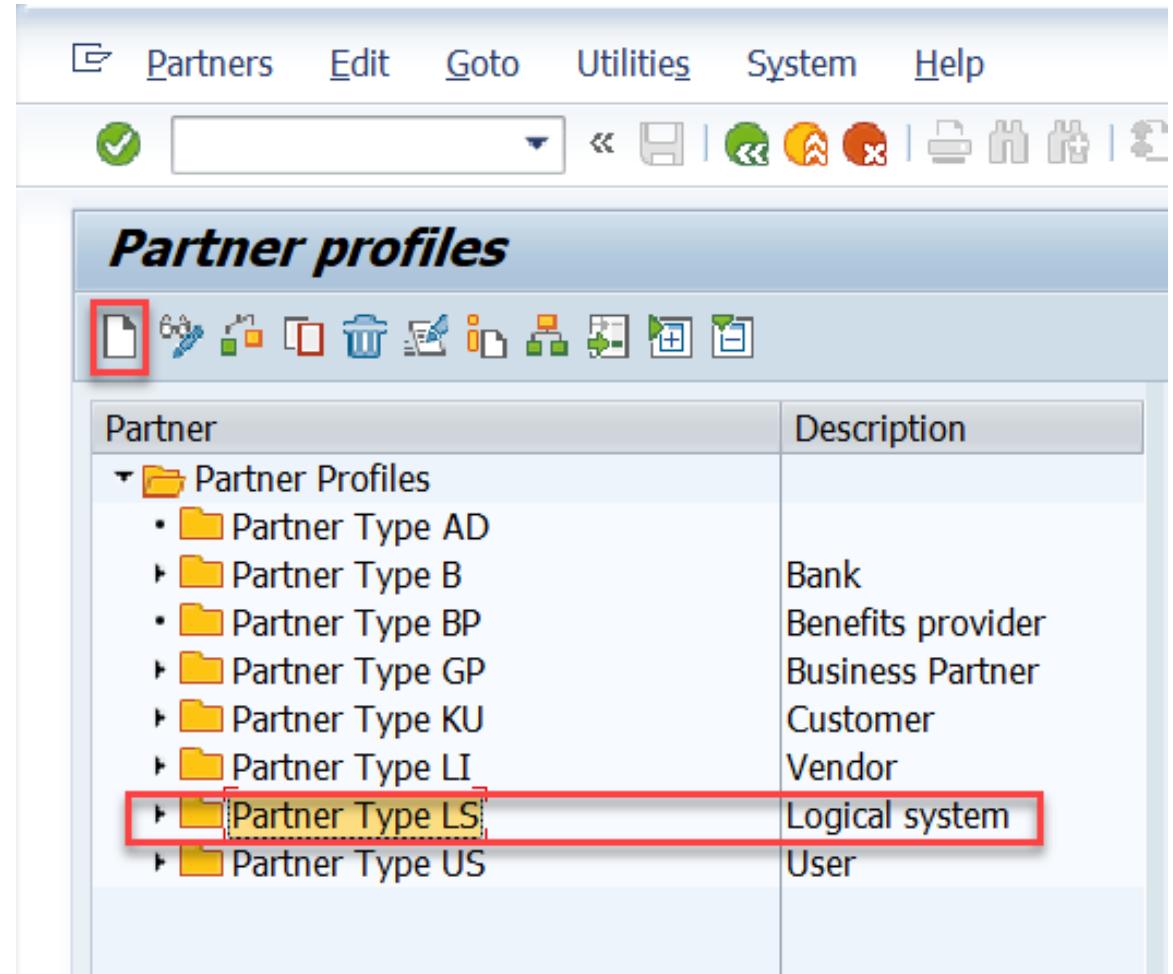
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components -> Interface to Global Track and Trace -> IDoc Settings**

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



## STEP 4: Define Partner Profiles

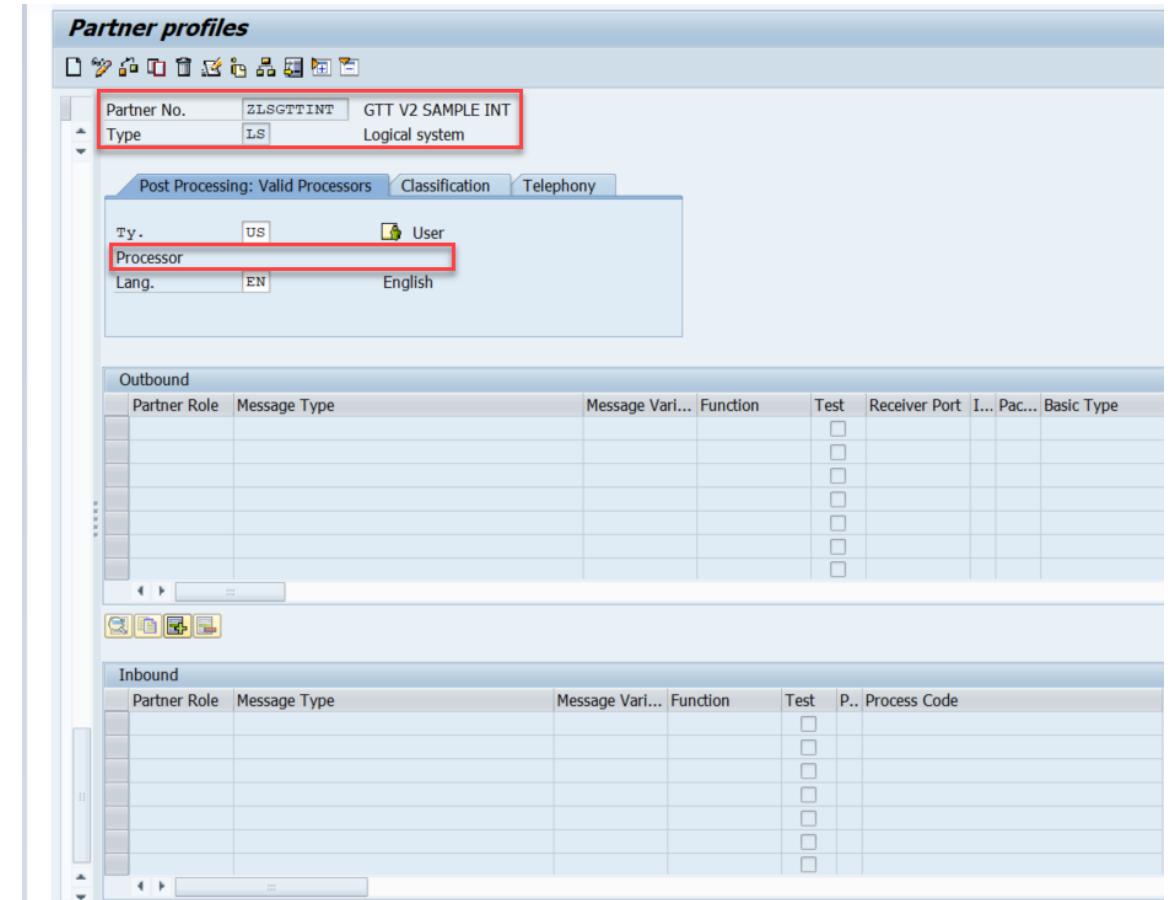
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



# 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



# STEP 4: Define Partner Profiles

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

The screenshot shows the SAP Fiori interface for defining partner profiles. At the top, there's a toolbar with various icons. Below it, the main area is titled "Partner profiles". A header bar displays "Partner No. ZLSGTTINT", "GTT V2 SAMPLE INT", "Type LS", and "Logical system". There are three tabs: "Post Processing: Valid Processors" (selected), "Classification", and "Telephony". Under the "Post Processing" tab, there's a section for "Ty." (Type) set to "US" and "User", "Processor" (Processor) set to "EN" and "English", and a "Lang." (Language) field. The main content area is divided into two sections: "Outbound" and "Inbound". The "Outbound" section has a table with columns: Partner Role, Message Type, Message Vari..., Function, Test, Receiver Port, I... Pac..., and Basic Type. The "Inbound" section has a similar table. At the bottom of each section, there are buttons for "Add", "Edit", "Delete", and "Search". The "Add" button is highlighted with a red box.

# STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

**Message Type:** EVMSTA

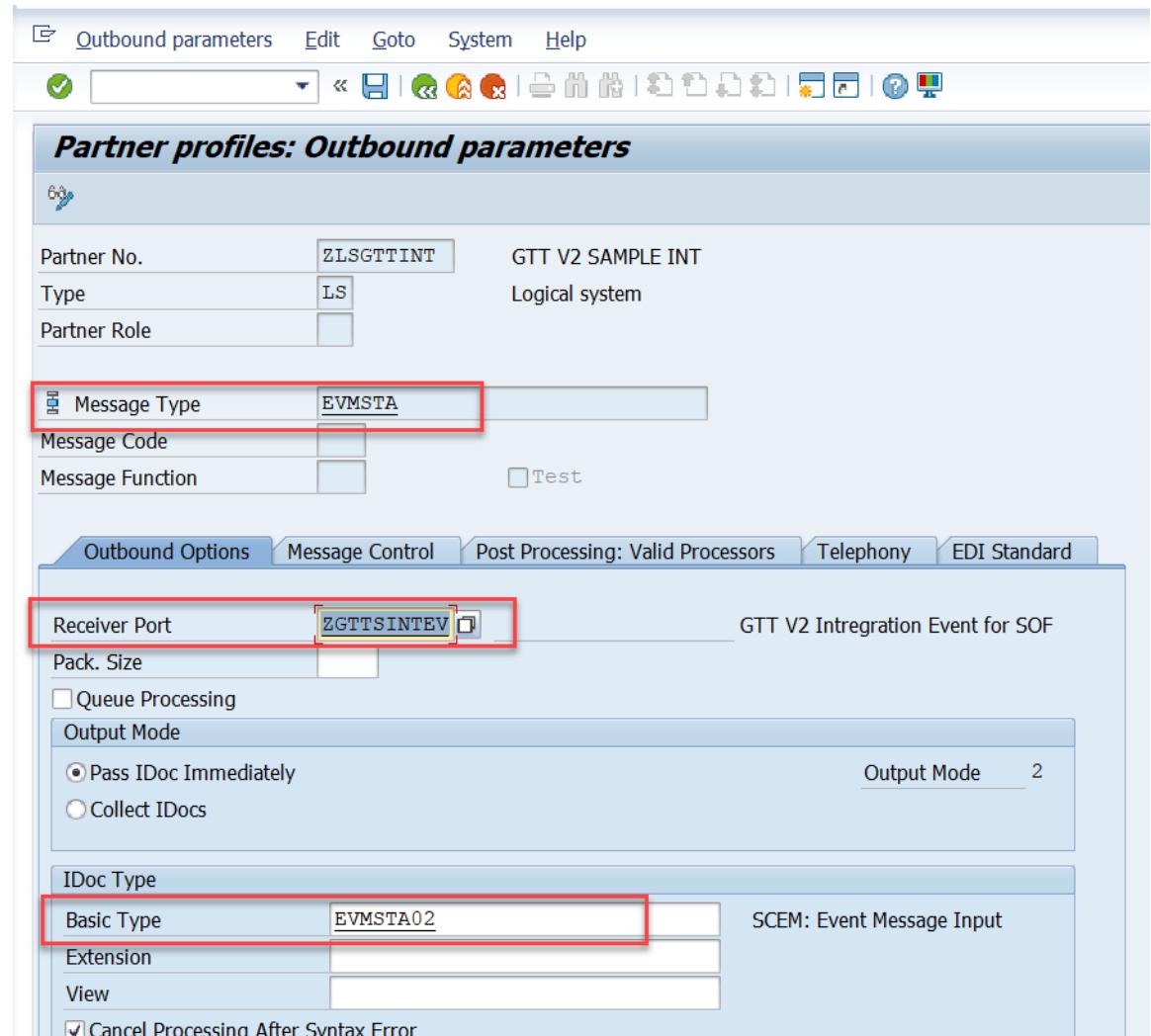
For the tracked Process:

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



# 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
ZLSGTTINT	LS	Yes	AOPOST	ZGTTSINTTP	EHPOST01
ZLSGTTINT	LS	Yes	EVMSTA	ZGTTSINTEV	EVMSTA02

*Partner profiles: Outbound parameters*

The screenshot shows the SAP Fiori interface for defining partner profiles. The title bar says "Partner profiles: Outbound parameters". The main area has sections for Partner No., Type, and Partner Role. Below these, there are fields for Message Type (set to AOPOST), Message Code, and Message Function. A "Test" button is also present. At the bottom, there are tabs for Outbound Options, Message Control, Post Processing: Valid Processors, Telephony, and EDI Standard. Under Outbound Options, the Receiver Port is set to ZGTTSINTTP. There are options for Queue Processing, Output Mode (Pass IDoc Immediately is selected), and Output Mode value 2. In the EDI Standard section, the Basic Type is set to EHPOST01. There are checkboxes for Cancel Processing After Syntax Error and Seg. release in IDoc type. Application Release and View fields are also present. A status bar at the bottom right says "SCEM: Event Handler Posting".

# B) Configuration and Implementation

## - Basic

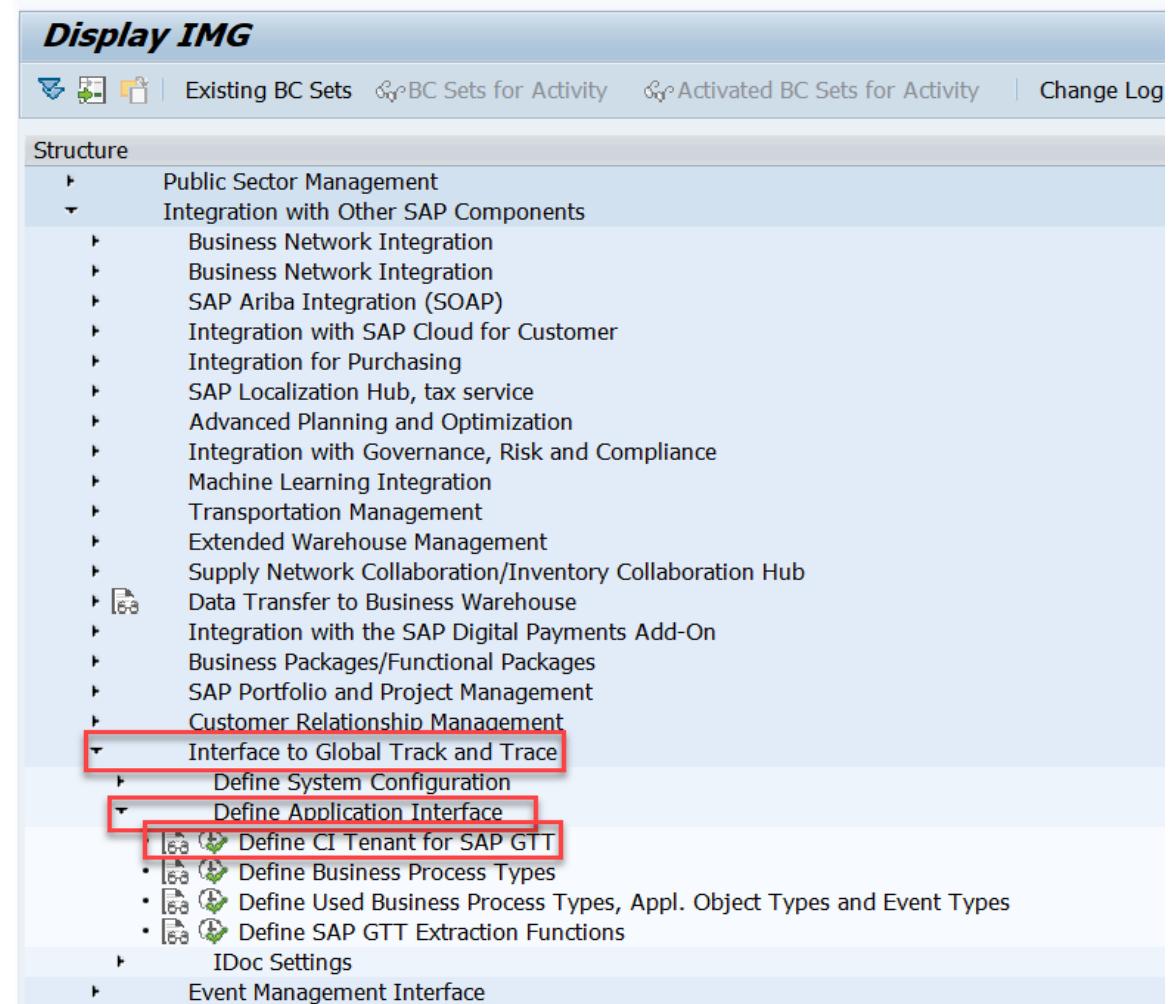
### B2. Extractor Configuration



# STEP 5: Define CI Tenant for SAP Business Network Global Track and Trace

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

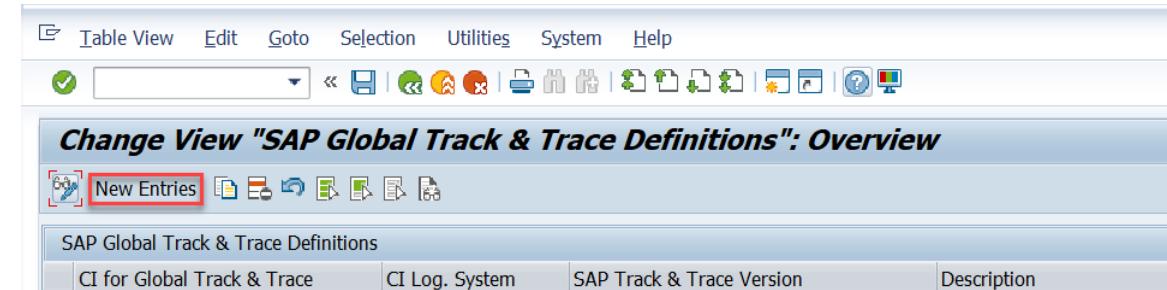
5-2: Choose activity  
**Define CI Tenant for SAP GTT**



# STEP 5: Define CI Tenant for SAP Business Network Global Track and Trace

5-3: Click **New Entries** to create a new CI tenant for SAP Business Network Global Track and Trace

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



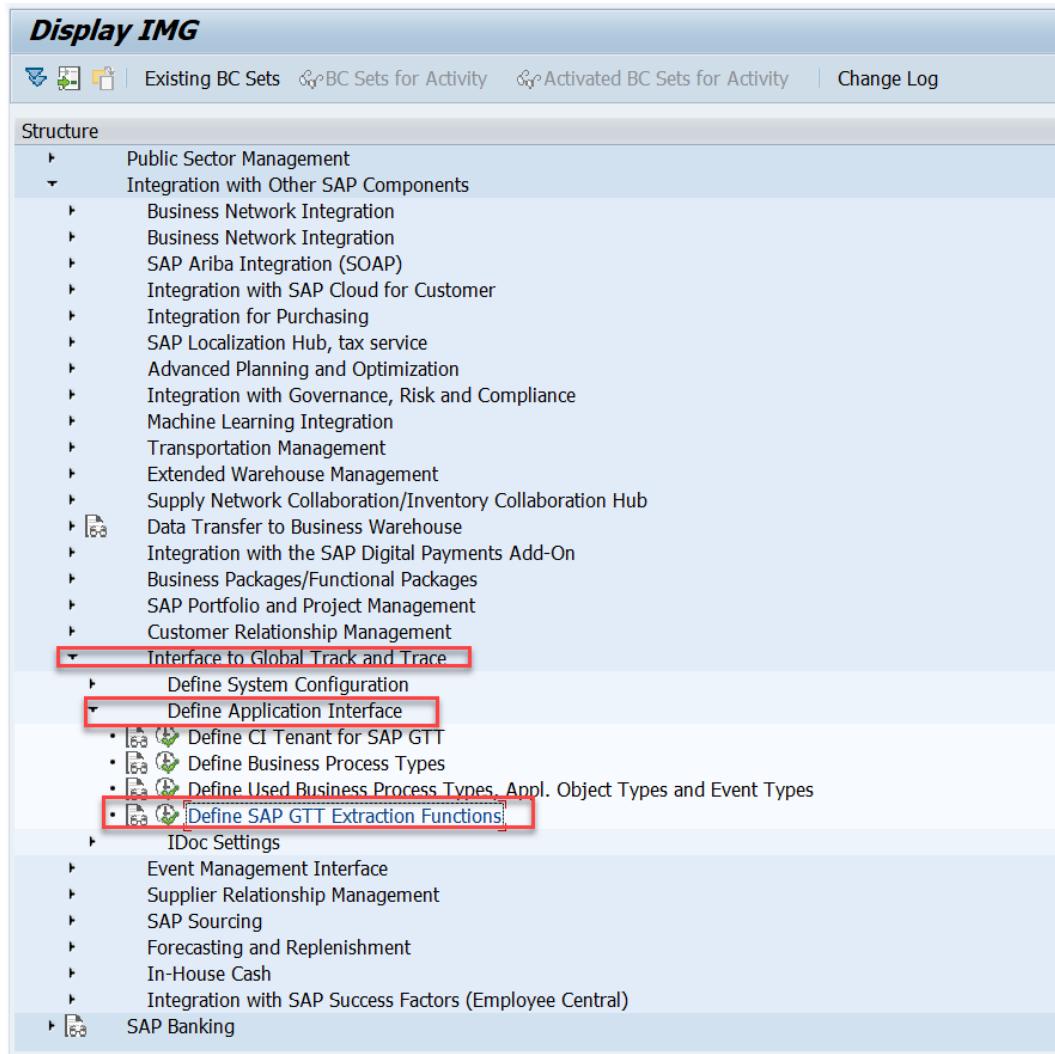
The screenshot shows the SAP Fiori interface with the title "Display View 'SAP Global Track & Trace Definitions': Overview". Below the title is a toolbar with various icons. The main area displays a table titled "SAP Global Track & Trace Definitions" with four columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. A specific row is selected and highlighted with a yellow background. The selected row contains the following data:

CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTSOFINST	ZLSGTTINT	GTT1.0 Global Track & Trace	CI For GTT V2 Integration system Sales Order Sample APP

# STEP 6: Define GTT Extraction Functions

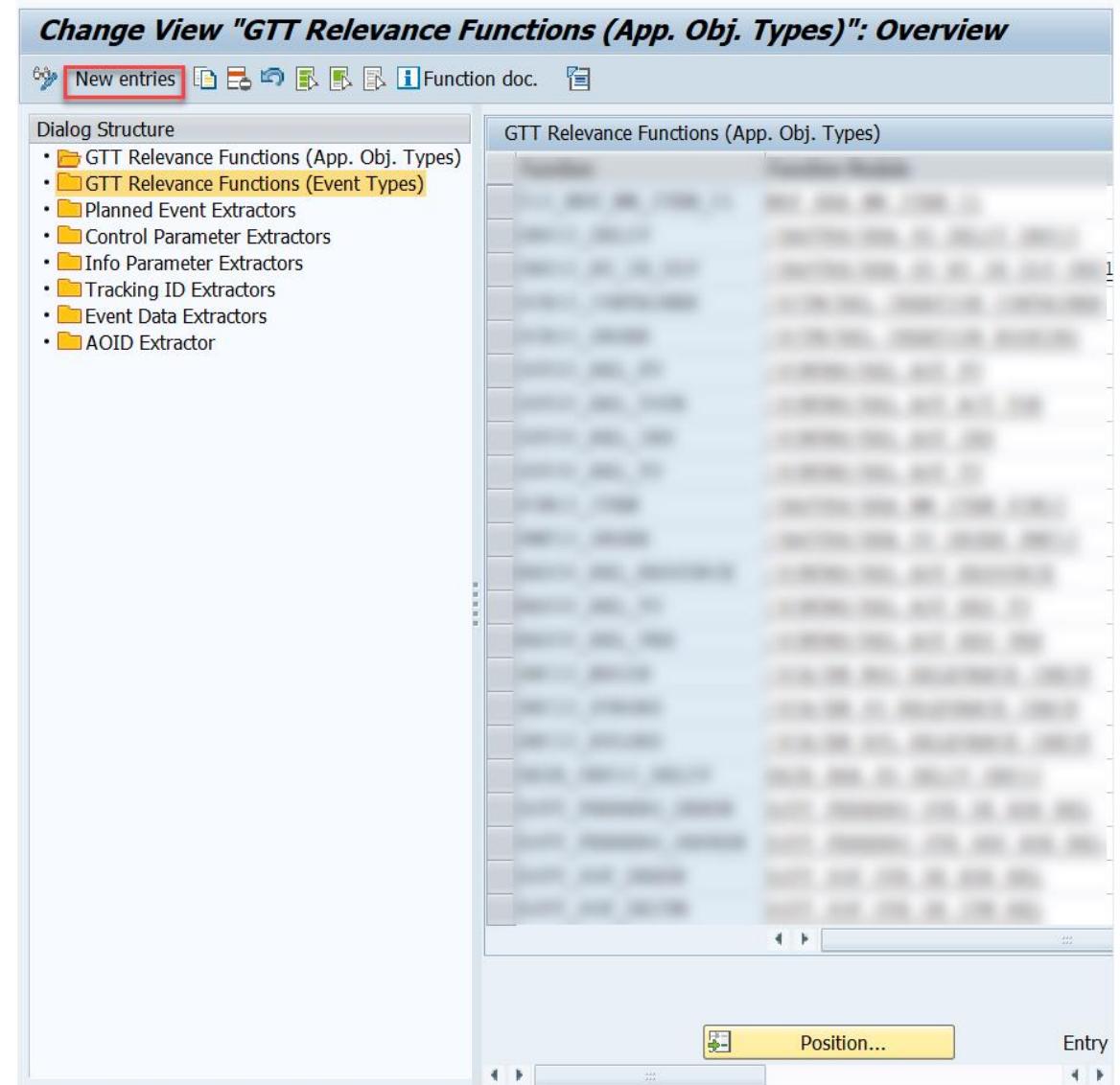
6-1: In **Display IMG** page, click  
**Integration with Other SAP Components ->**  
**Interface to Global Track and Trace ->**  
**Define Application Interface**

6-2: Choose activity  
**Define SAP GTT Extraction Functions**



# STEP 6: Define GTT Extraction Functions

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



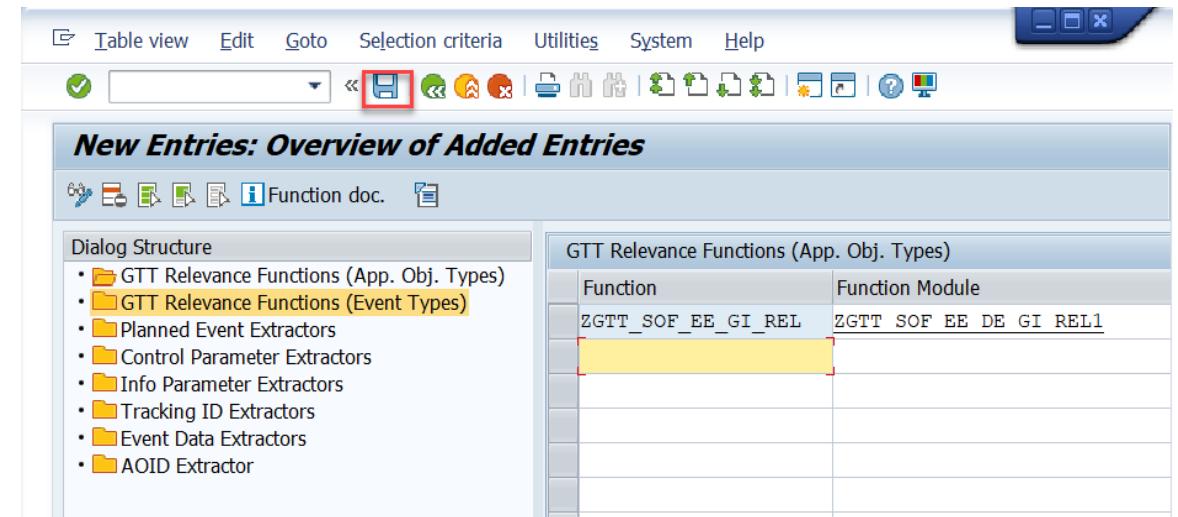
# STEP 6: Define GTT Extraction Functions

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

New Entries: Overview of Added Entries	
Function doc.	
Dialog Structure	
<ul style="list-style-type: none"><li>•  GTT Relevance Functions (App. Obj. Types)</li><li>•  GTT Relevance Functions (Event Types) <b>(highlighted in yellow)</b></li><li>•  Planned Event Extractors</li><li>•  Control Parameter Extractors</li><li>•  Info Parameter Extractors</li><li>•  Tracking ID Extractors</li><li>•  Event Data Extractors</li><li>•  AOID Extractor</li></ul>	
GTT Relevance Functions (App. Obj. Types)	
Function	Function Module
ZGTT_SOF_EE_GI_REL	ZGTT_SOF_EE_DE_GI_REL1

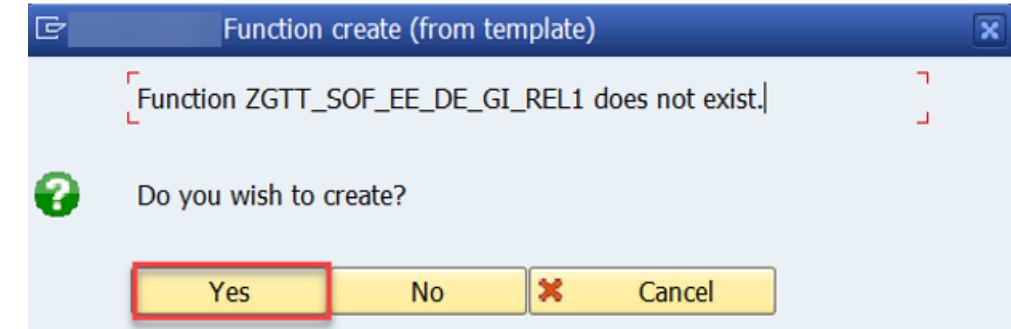
# STEP 6: Define GTT Extraction Functions

6-5: Click Save



## STEP 6: Define GTT Extraction Functions

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



# STEP 6: Define GTT Extraction Functions

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

6-8: Click **Copy**



# STEP 6: Define GTT Extraction Functions

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

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

The screenshot shows the SAP Function Builder interface. In the top navigation bar, the title is "Function Builder: Display ZGTT\_SOEE\_DE\_GI\_REL1". Below the title, there are tabs for "Attributes", "Import", "Export", "Changing", "Tables", "Exceptions", and "Source Code". The "Source Code" tab is selected. On the left, a "Repository Browser" pane shows a tree structure under "Function Group" set to "ZGTT\_SOEE". The node "ZGTT\_SOEE\_DE\_GI\_REL1" is highlighted with a red box. The main pane displays the ABAP source code for the function module:

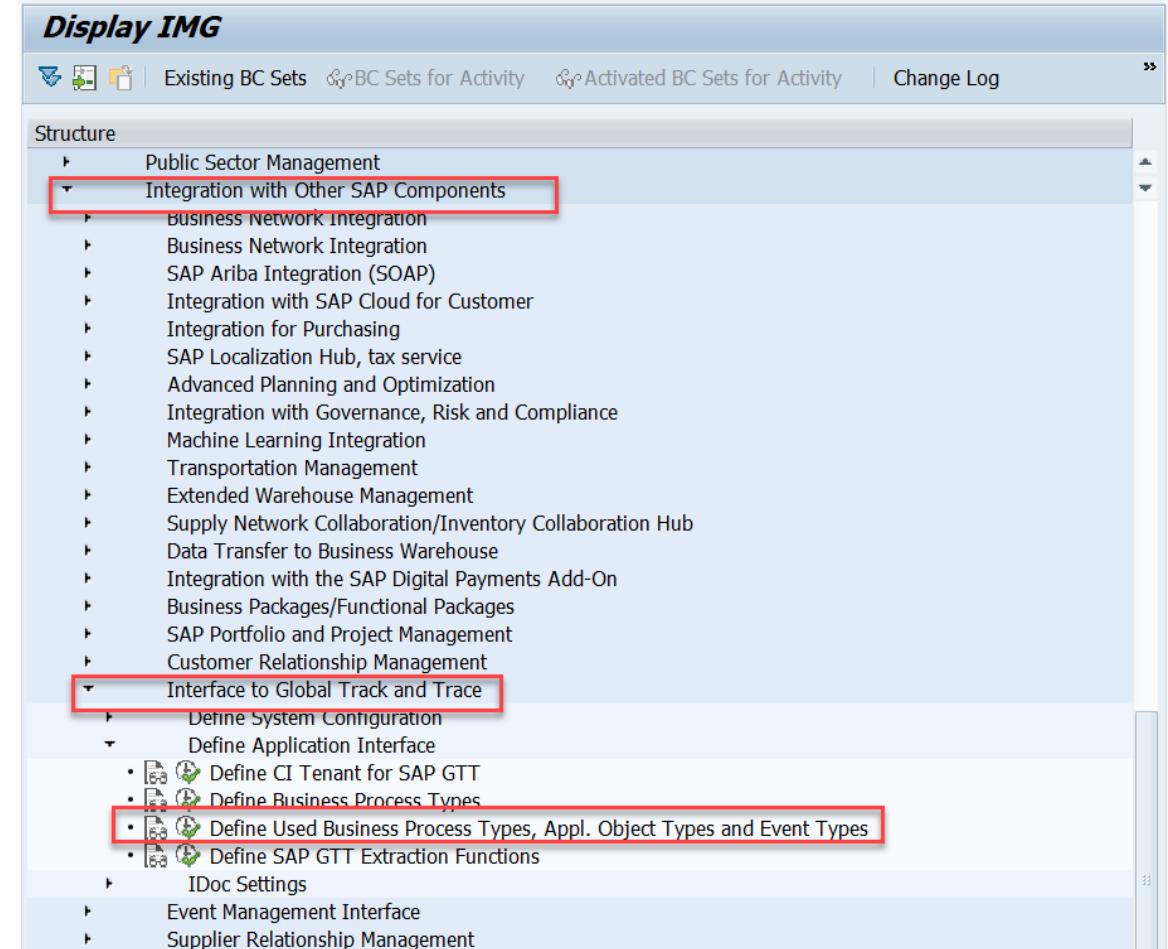
```
1: FUNCTION ZGTT_SOEE_DE_GI_REL1.
2:   *"*
3:   *" Local Interface:
4:   *" IMPORTING
5:   *"   REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM
6:   *"   REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/AOTYPES
7:   *"   REFERENCE(I_ALL_APFL_TABLES) TYPE TRXAS_TABCONTAINER
8:   *"   REFERENCE(I_APPTYPE_TAB) TYPE TRXAS_APPTYPE_TABS_WA
9:   *"   REFERENCE(I_APP_OBJECT) TYPE TRXAS_APPOBJ_CTAB_WA
10:  *" EXPORTING
11:  *"   VALUE(E_RESULT) LIKE SY-BINPT
12:  *"   TABLES
13:  *"     C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL
14:  *" EXCEPTIONS
15:  *"   PARAMETER_ERROR
16:  *"   RELEVANCE_DETERM_ERROR
17:  *"   STOP_PROCESSING
18:  *-
19:  * Top Include
20:  * TYPE-POOLS:trxas.
21:  *
22:  *
23:  *
24:  *
25:  *
26: ENDFUNCTION.
```

At the bottom of the code pane, it says "Scope: FUNCTION ZGTT\_SOEE\_DE\_GI\_REL1" and "ABAP | Ln 13 Col 48".

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

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

7-2: Choose activity **Define Used Business Process Types, Appl. Object Types and Event Types**



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

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

In the following:

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

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

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

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

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

Change View "Define Used Business Process Types": Overview			
Define Used Business Process 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
ESC_SHIPMT	<input checked="" type="checkbox"/> 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 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

Change View "Define Event Types": Overview		
New Entries		
Dialog Structure		
Define Used Business Process Types		
Business Process Type	Event Type	Description
ESC_SHIPMT	GTT_ARRIVAL_ACC_SO	Arrival Event for GTT Sample SO Acceptance System
ESC_SHIPMT	GTT_ARRIVAL_INT_SO	Arrival Event for GTT Sample so Integration System
ESC_SHIPMT	GTT_CHIN_ACC_SO	Check In Event for GTT Sample So Acceptance System
ESC_SHIPMT	GTT_CHIN_INT_SO	Check In Event for GTT Sample SO Integration System
ESC_SHIPMT	GTT_DEPART_ACC_SO	Departure Event for GTT Sample So Acceptance System
ESC_SHIPMT	GTT_DEPART_INT_SO	Departure Event for GTT Sample So Integration System
ESC_SHIPMT	GTT_LDED_ACC_SO	Loading End Event for GTT Sample SO Acceptance System
ESC_SHIPMT	GTT_LDED_INT_SO	Loading End Event for GTT Sample SO Integration System
ESC_SHIPMT	GTT_LDST_ACC_SO	Loading Start Event for GTT Sample SO Acceptance System
ESC_SHIPMT	GTT_LDST_INT_SO	Loading Start Event for GTT Sample SO Integration System
ESC_SHIPMT	TRA10_ROAD	Road Shipment for Transportation Visibility
ESC_SHIPMT	YSHIPMENT_ACC	Road Shipment for Transportation Visibility
ESC_SHIPMT	YSHIPMENT_INT	Road Shipment for Transportation Visibility
ESC_SHIPMT	ZGTT_SOF_ARRIVAL_ACC	Arrival Event for GTT SOF Acceptance System
ESC_SHIPMT	ZGTT_SOF_ARRIVAL_INT	Arrival Event for GTT SOF Integration System
ESC_SHIPMT	ZGTT_SOF_CHIN_ACC	Check In Event for GTT SOF Acceptance System
ESC_SHIPMT	ZGTT_SOF_CHIN_INT	Check In Event for GTT SOF Integration System
ESC_SHIPMT	ZGTT_SOF_DEPART_ACC	Departure Event for GTT SOF Acceptance System
ESC_SHIPMT	ZGTT_SOF_DEPART_INT	Departure Event for GTT SOF Integration System
ESC_SHIPMT	ZGTT_SOF_LDED_ACC	Loading End Event for GTT SOF Acceptance System
ESC_SHIPMT	ZGTT_SOF_LDED_INT	Loading End Event for GTT SOF Integration System
ESC_SHIPMT	ZGTT_SOF_LDST_ACC	Loading Start Event for GTT SOF Acceptance System

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

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

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

7-8: Check **GTT Relevant**

Bus. Proc. Type	ESC_SHIPMT
Event Type	ZGTT_SOF_CHIN_INT
Text	Check In Event

General Data    Control Tables    Global Track & Trace Relevance

Sequencing / Destination	
Seq. No.	10
HCI for GTT	ZGTTSOFINTE

Data Setup	
Event Function	ZGTT_SOF_EVNT_CHIN

Behavior
<input checked="" type="checkbox"/> GTT Relevant
<input type="checkbox"/> Stop ET Det.
<input type="checkbox"/> Appl. Log Deact

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

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

## Caution:

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

The image displays two screenshots of SAP configuration interfaces for defining business processes.

**Top Screenshot (Business Process Type: ESC\_SHIPMT):**

- Data Source for Events:**
  - Main Obj. Table: SHIPMENT\_HEADER\_NEW (highlighted with a red box)
  - Master Table: SHIPMENT\_HEADER\_OLD (highlighted with a red box)
- A callout box labeled "Event on Header level" points to the Main Obj. Table field.
- Reference Between Main and Master Table:**
  - First Field Reference from Main to Master Table
  - Second Field Reference from Main to Master Table

**Bottom Screenshot (Business Process Type: ESC\_DELIV):**

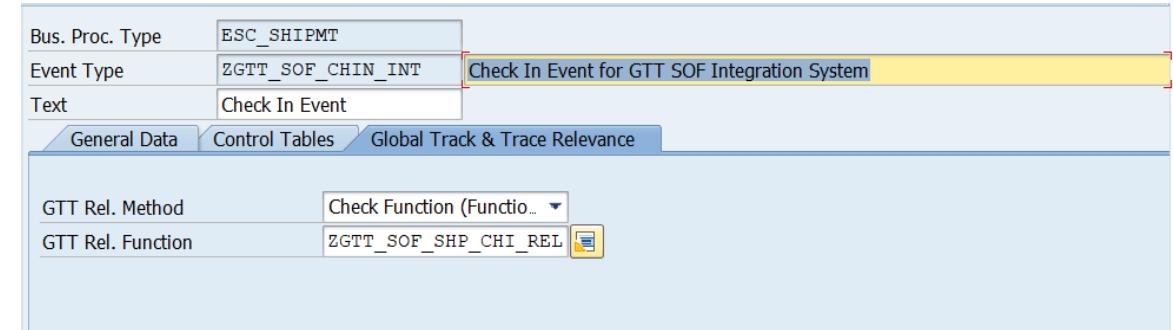
- Data Source for Events:**
  - Main Obj. Table: DELIVERY\_ITEM\_NEW (highlighted with a red box)
  - Master Table: DELIVERY\_HEADER\_NEW (highlighted with a red box)
- A callout box labeled "Event on Item level" points to the Main Obj. Table field.
- Reference Between Main and Master Table:**
  - First Field Reference from Main to Master Table
    - Uplink Field: VBELN
    - Uplink Mode: R
    - Uplink Target Fld: VBELN
    - Uplink Const: (empty)
  - Second Field Reference from Main to Master Table
    - Uplink Field: (empty)
    - Uplink Mode: (empty)
    - Uplink Target Fld: (empty)
    - Uplink Const: (empty)

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

7-10: In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need.

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

Click **Save**.



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

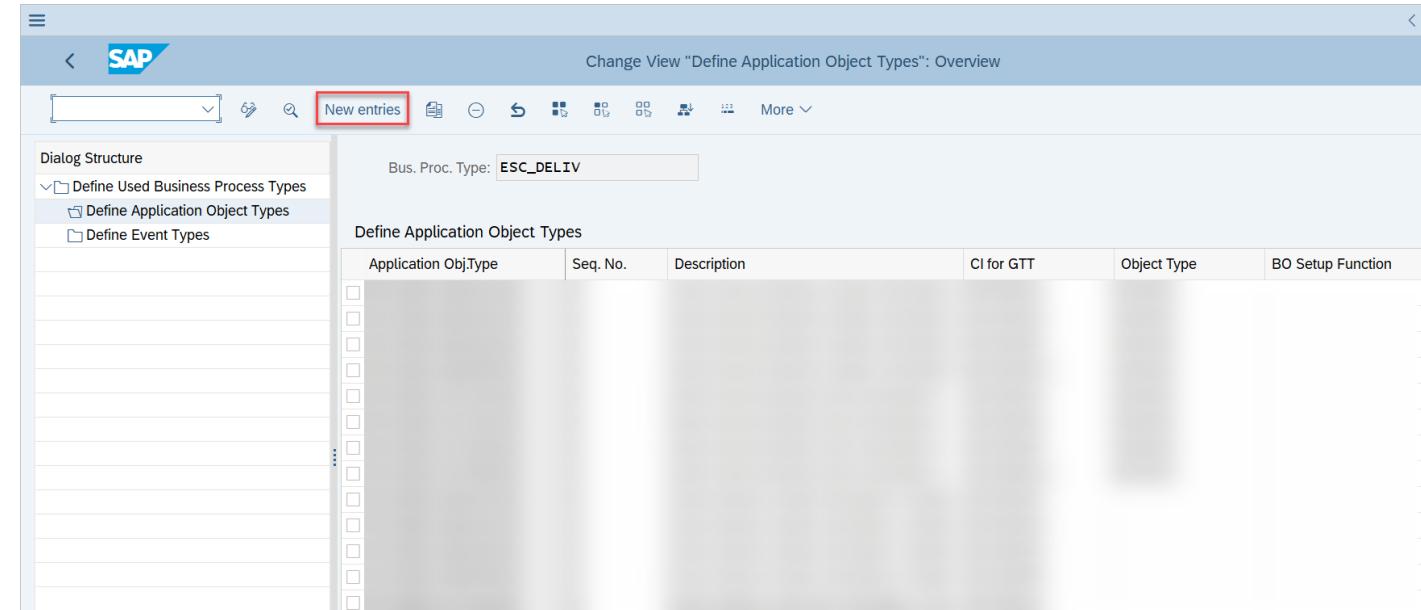
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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

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



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

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

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

7-16: Check **GTT Relevant**

The screenshot shows the SAP Fiori application configuration interface. The top section displays the following fields:

- Bus. Proc. Type: ESC\_SHIPMT
- Appl. Obj. Type: ZGTT\_SHP\_INT\_HD (highlighted with a red box)
- Text: Extract shipment header information to Global Track and Trace Integration (also highlighted with a red box)

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

The General Data tab contains several sections:

- Sequencing / Destination:** Seq. No.: 20, CI for GTT: ZGTTSOFINST (highlighted with a red box). The description "CI For GTT V2 Integration system Sales Order Sampl" is visible.
- Business Object Reference:** Object Type: [empty], BO Setup Fnct.: [empty] (with a small info icon).
- Behavior:** A checkbox labeled "GTT Relevant" is checked (highlighted with a red box). Other options include "Stop AO Determ." and "Appl. Log Deact".

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

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

## Caution:

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

The image contains two screenshots of SAP configuration screens. Both screens have a top bar with tabs: General Data, Control Tables (which is selected), Object Identification, Global Track & Trace Relevance, and Parameter Setup.

**Top Screen (Business Process Type: ESC\_SHIPMT):**

- Bus. Proc. Type: ESC\_SHIPMT
- Appl. Obj. Type: ZGTT\_SHP\_INT\_HD Extract shipment header information to Global Track and Trace Integration
- Text: [empty]

**Bottom Screen (Business Process Type: ESC\_DELIV):**

- Bus. Proc. Type: ESC\_DELIV
- Appl. Obj. Type: ZGTT\_DE\_INT\_ITEM Extract delivery order item information to Global Track and Trace Integration
- Text: Delivery Item

**Control Tables Tab Content (Both Screens):**

- Data Source for Created and Updated Objects:**
  - Main Obj. Table: SHIPMENT\_HEADER\_NEW (highlighted with a red box)
  - Master Table: [empty]
  - AOT on Header Level (highlighted with a red box)
- Data Source for Deleted Objects:**
  - Del.Obj. Table: SHIPMENT\_HEADER\_OLD (highlighted with a red box)
- Reference Between Main and Master Table:**
  - First Field Reference from Main to Master Table
- Data Source for Created and Updated Objects:**
  - Main Obj. Table: DELIVERY\_ITEM\_NEW (highlighted with a red box)
  - Master Table: DELIVERY\_HEADER\_NEW (highlighted with a red box)
  - AOT on Item Level (highlighted with a red box)
- Data Source for Deleted Objects:**
  - Del.Obj. Table: DELIVERY\_ITEM\_OLD (highlighted with a red box)
- Reference Between Main and Master Table:**
  - First Field Reference from Main to Master Table
    - Uplink Field: VBELN (highlighted with a red box)
    - Uplink Mode: R (highlighted with a red box)
    - Uplink Target Fld: VBELN (highlighted with a red box)
    - Uplink Const: [empty]

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

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

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

Bus. Proc. Type: ESC\_DELIV  
Appl. Obj. Type: ZGTT\_DE\_INT\_ITEM Extract delivery order item information to Global Track and Trace Integration  
Text: Delivery Item

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  
Second Field to Build Appl. Obj. ID

Cntrl Tab. Type: 1 Main Object Table  
AO ID Field: VBELN

Cntrl Tab. Type: 1 Main Object Table  
AO ID Field: POSNR

Determine AOID By Function

AOID Function:

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

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

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

The screenshot shows a configuration interface for a business process. At the top, there are three input fields: 'Bus. Proc. Type' (ESC\_DELIV), 'Appl. Obj. Type' (ZGTT\_DE\_INT\_ITEM), and 'Text' (Delivery Item). Below these, a note says 'Extract delivery order item information to Global Track and Trace Integration'. A horizontal navigation bar includes tabs for 'General Data', 'Control Tables', 'Object Identification', 'Global Track & Trace Relevance' (which is underlined in blue, indicating it is active), and 'Parameter Setup'. Under the active tab, two configuration fields are shown: 'GTT Rel. Method' (set to 'A Check Function (Function Module)') and 'GTT Rel. Function' (containing the value 'ZGTT\_SOF\_DEITM'). The 'GTT Rel. Function' field is highlighted with a red rectangular border.

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

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

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

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

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

Click **Save**.

The screenshot shows the SAP Fiori interface for parameter setup. At the top, there are fields for 'Bus. Proc. Type' (ESC\_DELIV), 'Appl. Obj. Type' (ZGTT\_DE\_INT\_ITEM) with a tooltip 'Extract delivery order item information to Global Track and Trace Integration', and 'Text' (Delivery Item). Below these are tabs for General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The Parameter Setup tab is active. Under 'Tracking ID Setup', the 'TrkID Method' dropdown is set to 'A Determine by Function', and the 'Trk.ID Function' dropdown is set to 'ZGTT\_TID\_DE\_ITEM'. A 'Tracking ID Fld:' input field is also present. Under 'Parameter Setup', the 'Ctrl Data Function' is 'ZGTT\_OTE\_DE\_ITEM', the 'Info Data Function' is empty, and the 'Planned Event Function' is 'ZGTT\_EE\_DE\_ITM'. All these settings are highlighted with a red border.

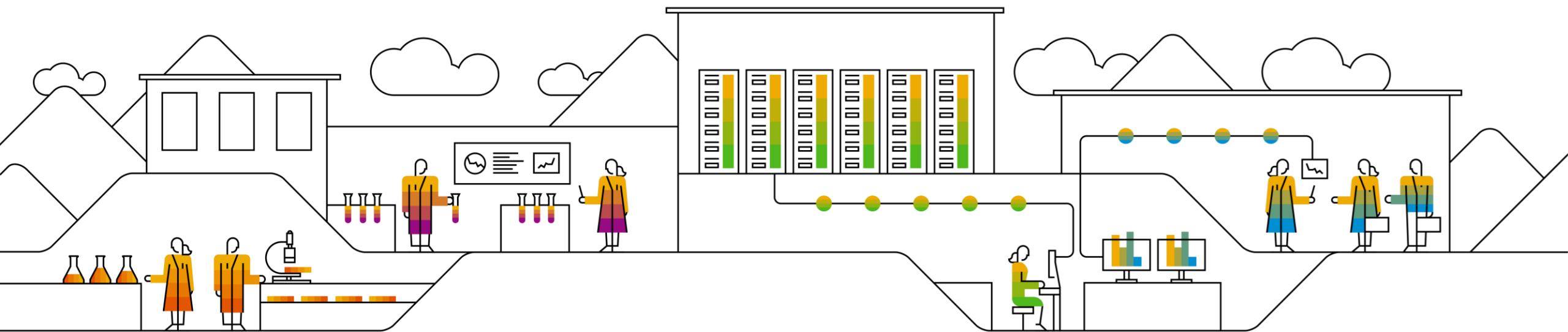
# C) Download ABAP Code from GitHub

**C1. Initial Download ABAP Code from GitHub (Only for TSOF)**

**C2. Update ABAP Code from GitHub (Only for TSOF)**

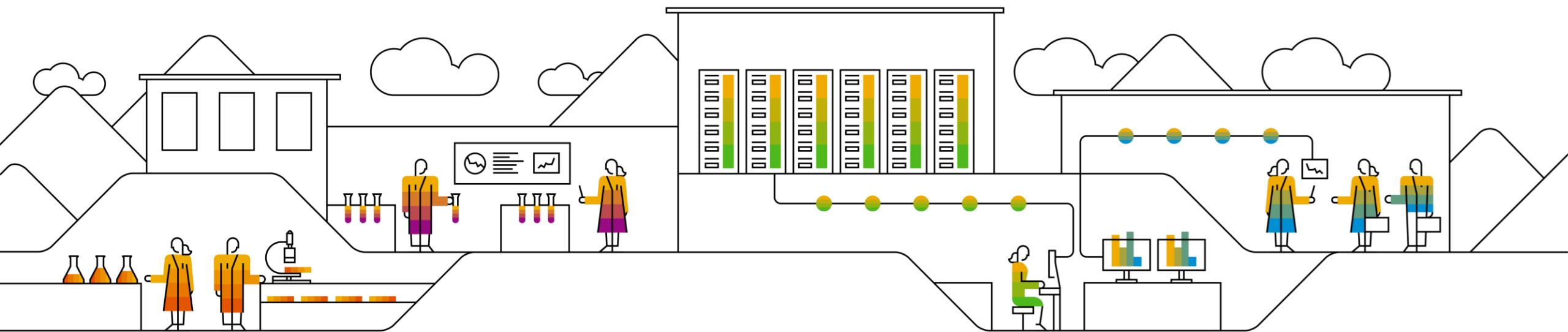
**C3. Download Another ABAP Code from GitHub (Only for TPOF)**

**C4. Initial Download ABAP Code from GitHub (Include TSOF/TPOF/TS)**



# C) Download ABAP Code from GitHub

## C1. Initial Download ABAP Code from GitHub (Only for TSOF)



# STEP 1: Install ABAPGit

You need to install ABAPGit before downloading the 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 consist 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

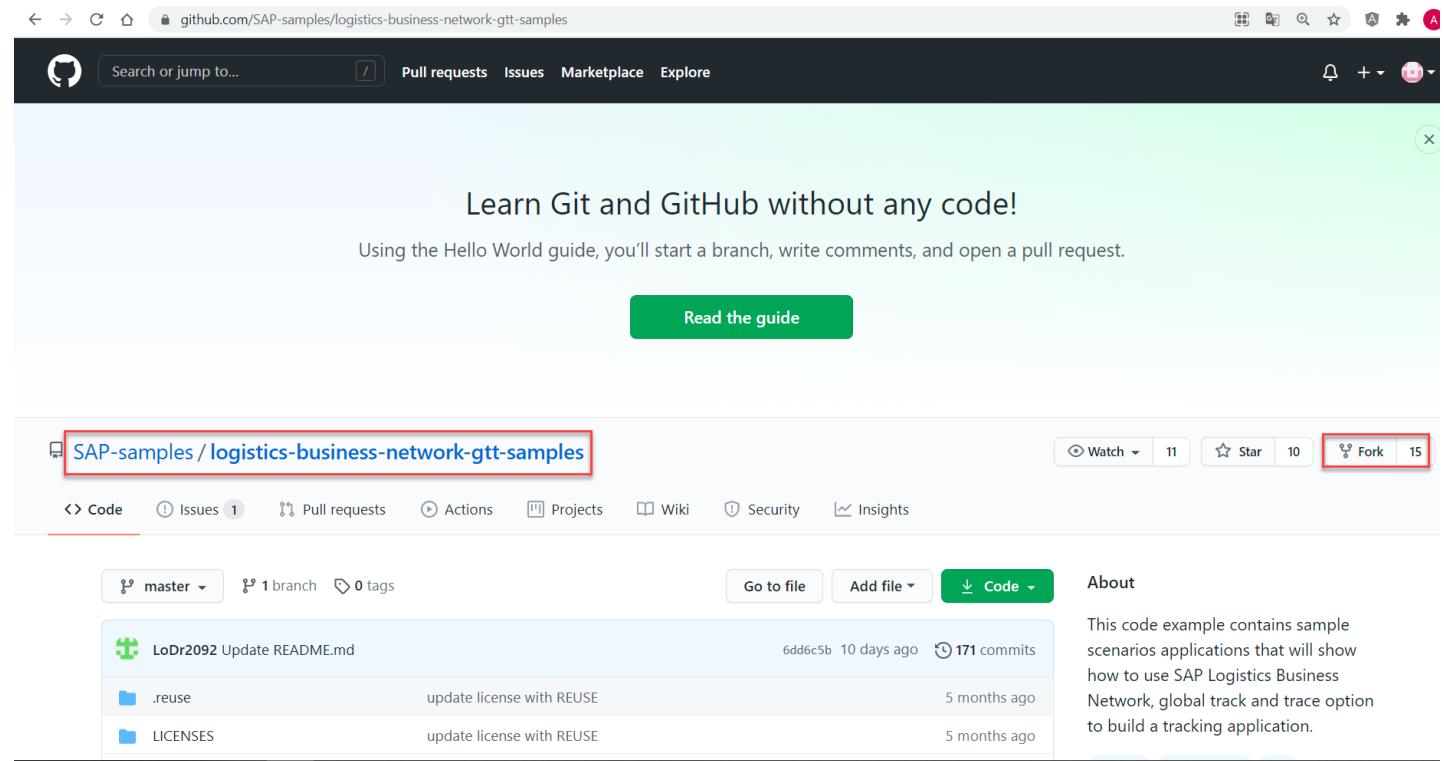
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT` ).

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 the user’s own repository.



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

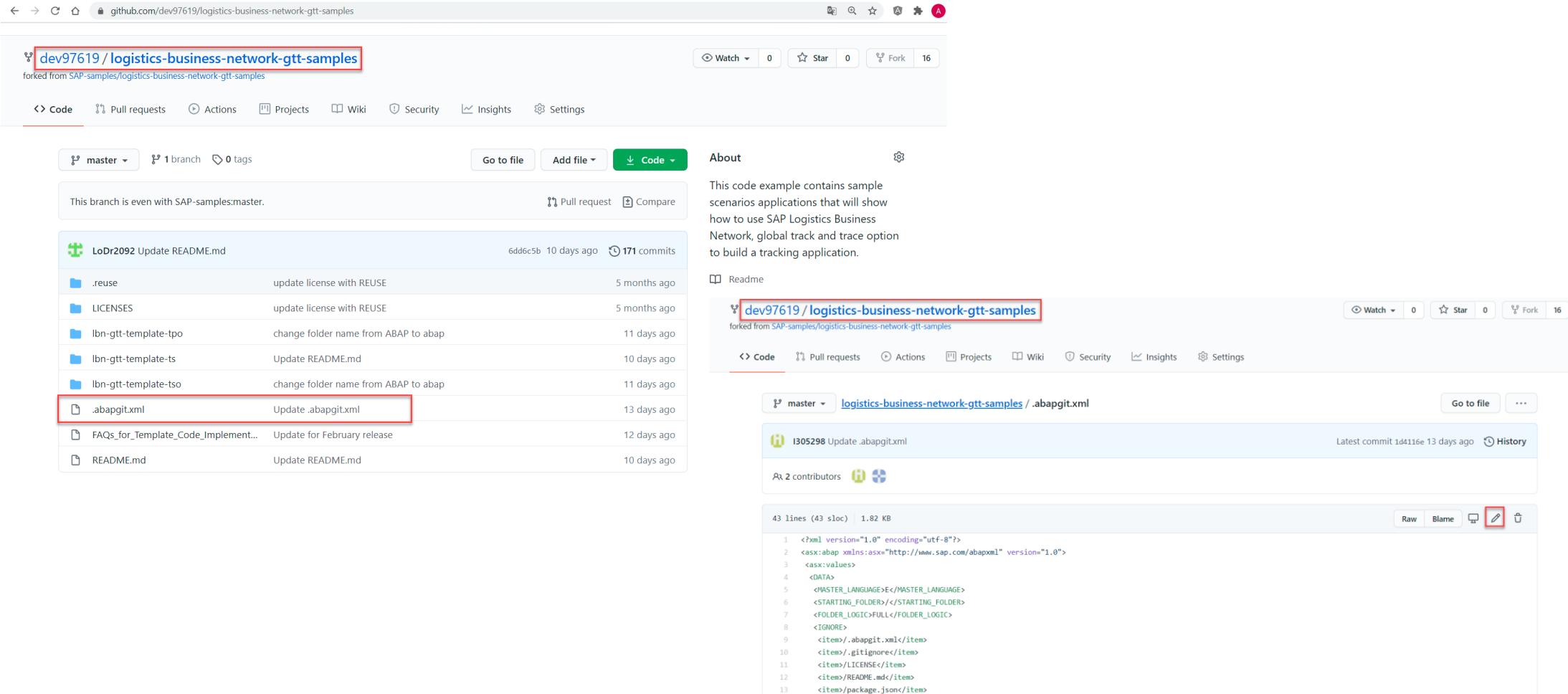
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit list includes several entries, with the last one, '.abapgit.xml', highlighted by a red box. The commit details show it was updated 13 days ago. To the right of the commit list, there is an 'About' section describing the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and build instructions. There are also sections for 'Readme', 'Releases', and 'Packages'.

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

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

3-2: Click  button to edit the file.



The screenshot shows a GitHub repository page for `dev97619/logistics-business-network-gtt-samples`. The repository has 16 forks. The main page displays a list of commits, including one from LoDr2092 that updated the README.md file. A commit from i305298 updated the `.abapgit.xml` file 13 days ago. The `.abapgit.xml` file content is shown in the code editor:

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

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

3-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with

"<STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>" as follows.

3-4: Commit change.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. On the right, a 'Commit changes' dialog is open over the '.abapgit.xml' file content.

**.abapgit.xml Content:**

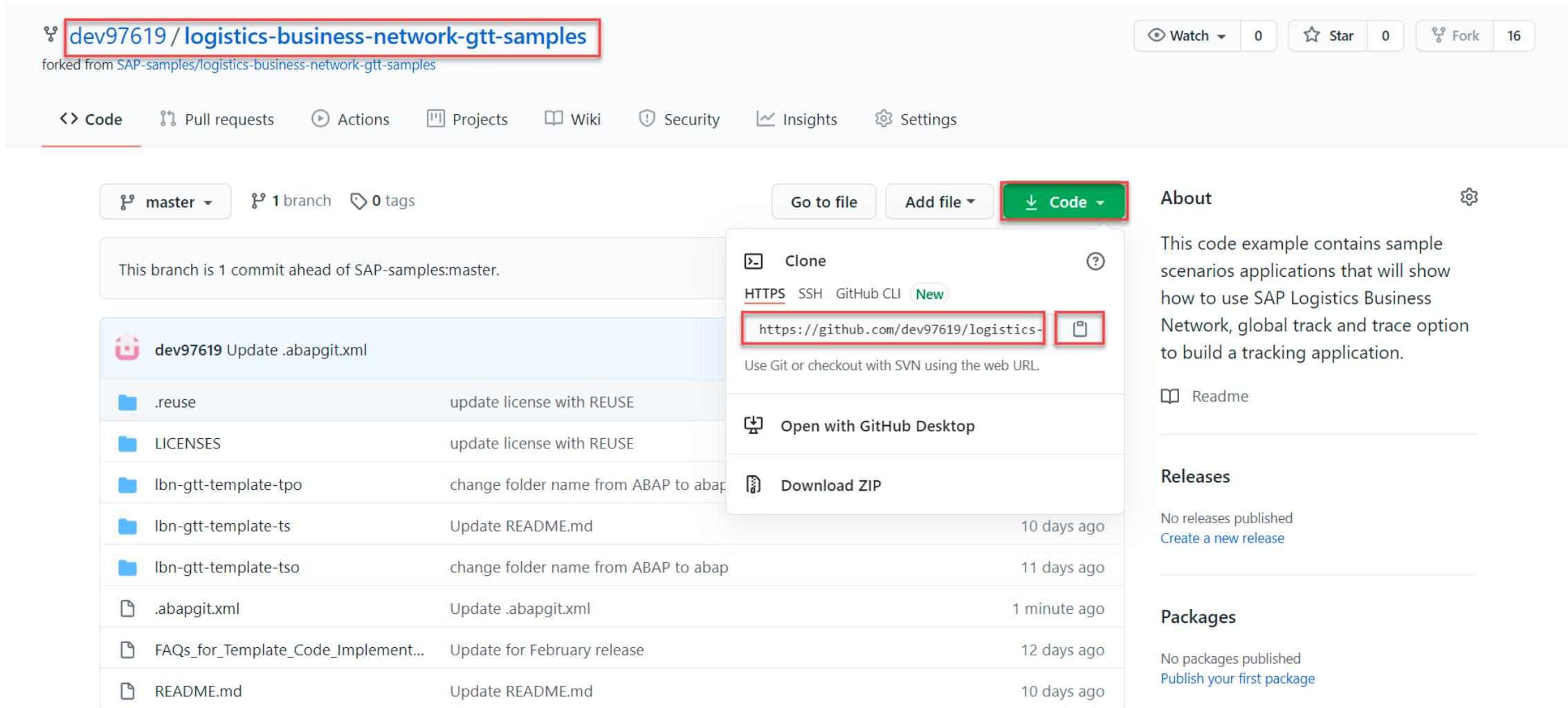
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3 <asx:values>
4 <DATA>
5 <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>
7 <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8 <IGNORE>
9 <item>/.abapgit.xml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

**Commit changes Dialog:**

- Input field: Update .abapgit.xml
- Text area: Add an optional extended description...
- Radio buttons:
  - Commit directly to the master branch.
  - Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)
- Buttons: Commit changes (green), Cancel (red)

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

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



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository has been forked from SAP-samples/logistics-business-network-gtt-samples. The 'Code' tab is selected. A dropdown menu is open over the 'Clone' link, with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples> highlighted and a copy icon () overlaid on it. The repository has 1 branch and 0 tags. The master branch is 1 commit ahead of SAP-samples:master. The .abapgit.xml file is listed in the commit history, along with other files like .reuse, LICENSES, and README.md. The repository has 0 stars, 0 forks, and 16 issues. The 'About' section describes the code example as containing sample scenarios for SAP Logistics Business Network, global track and trace options. The 'Readme' and 'Releases' sections are also visible.

dev97619 / logistics-business-network-gtt-samples

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

Ibn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

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

README.md Update README.md

Go to file Add file 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

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published Create a new release

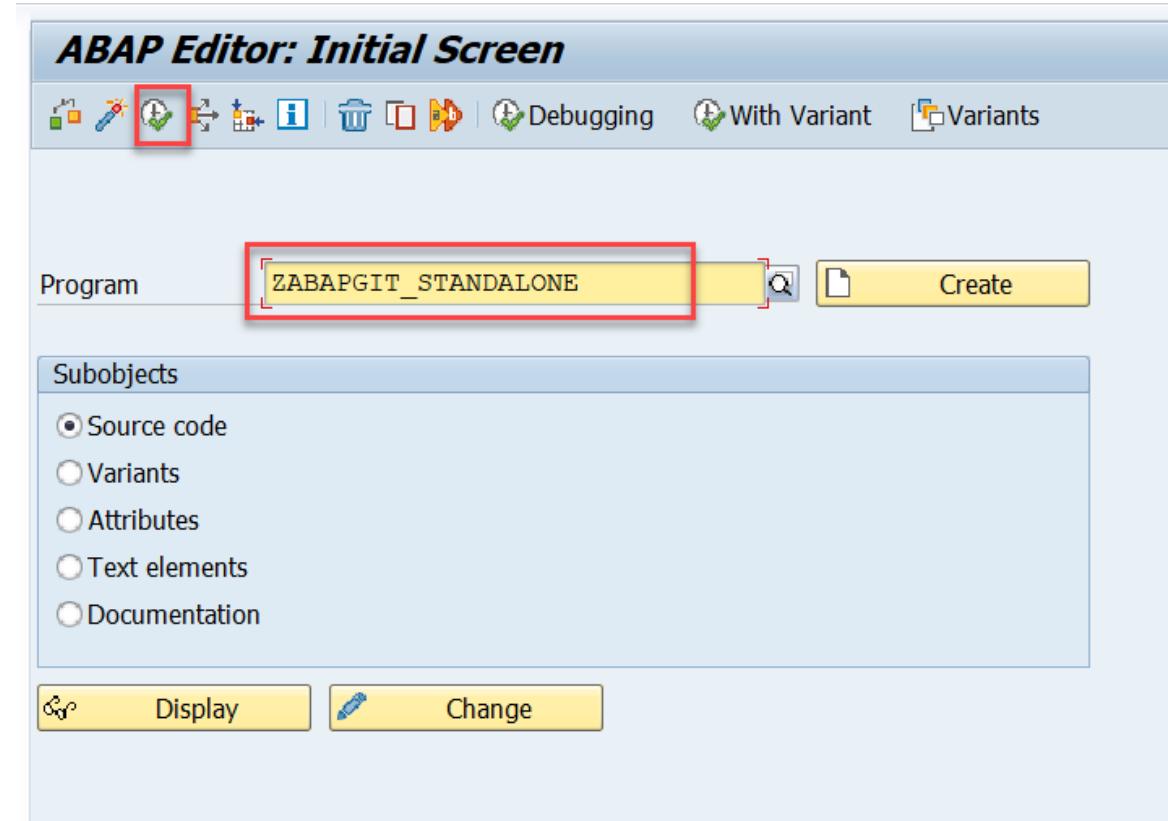
Packages

No packages published Publish your first package

## STEP 4: Download ABAP Code from GitHub

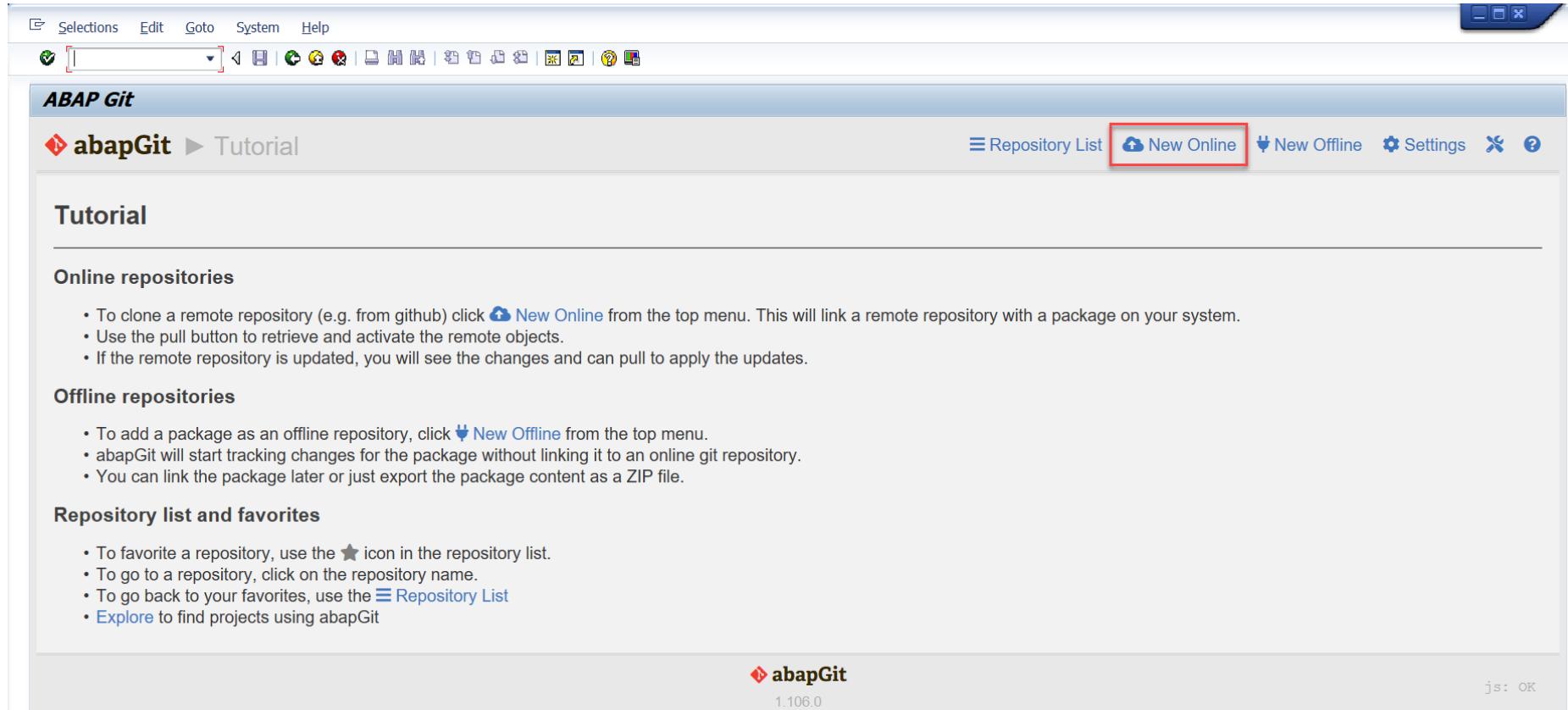
4-1: Enter T-code *SE38* and fill in the report name from STEP 1,  
*ZABAPGIT\_STANDALONE*.

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>

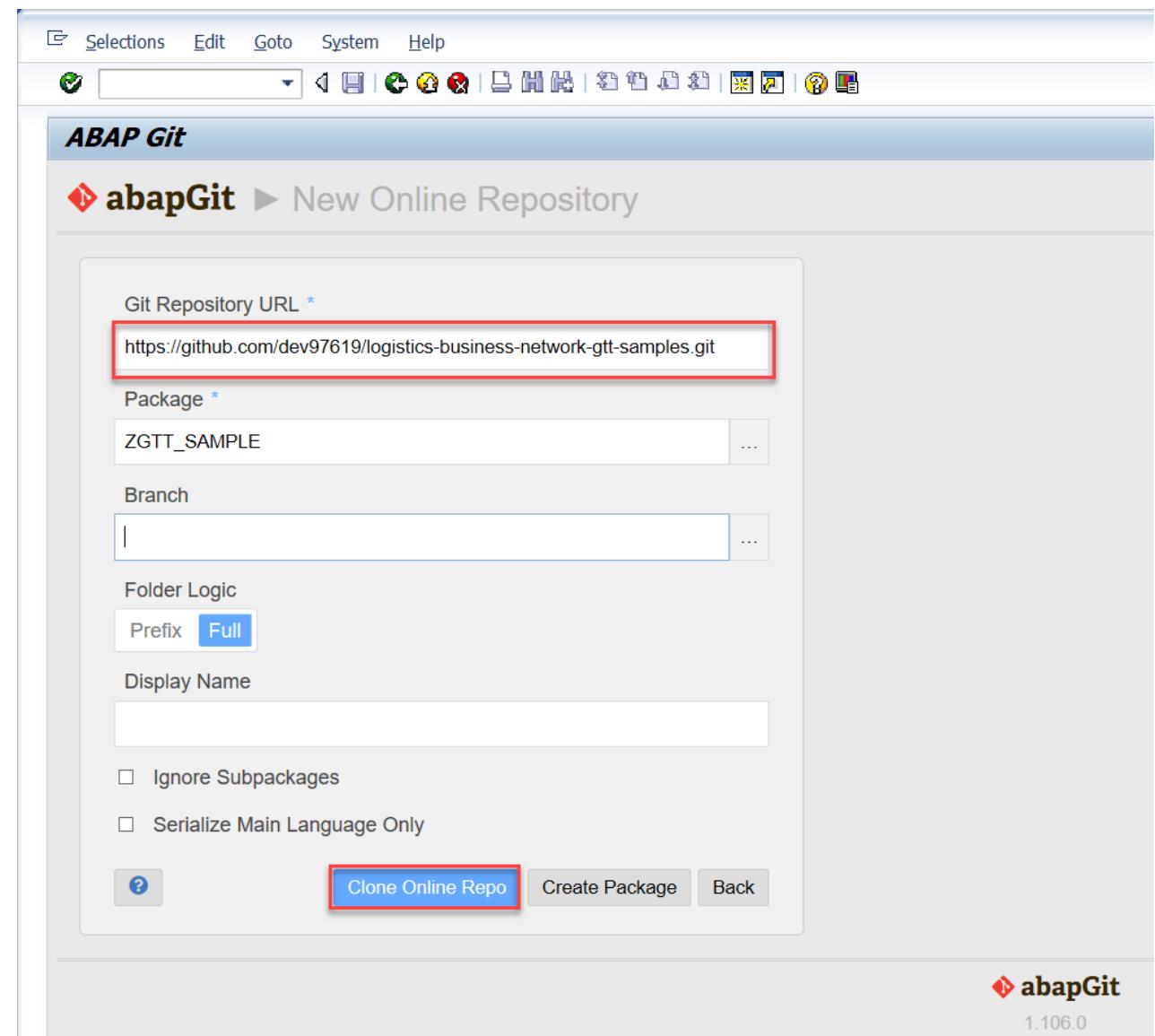
## Caution:

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.

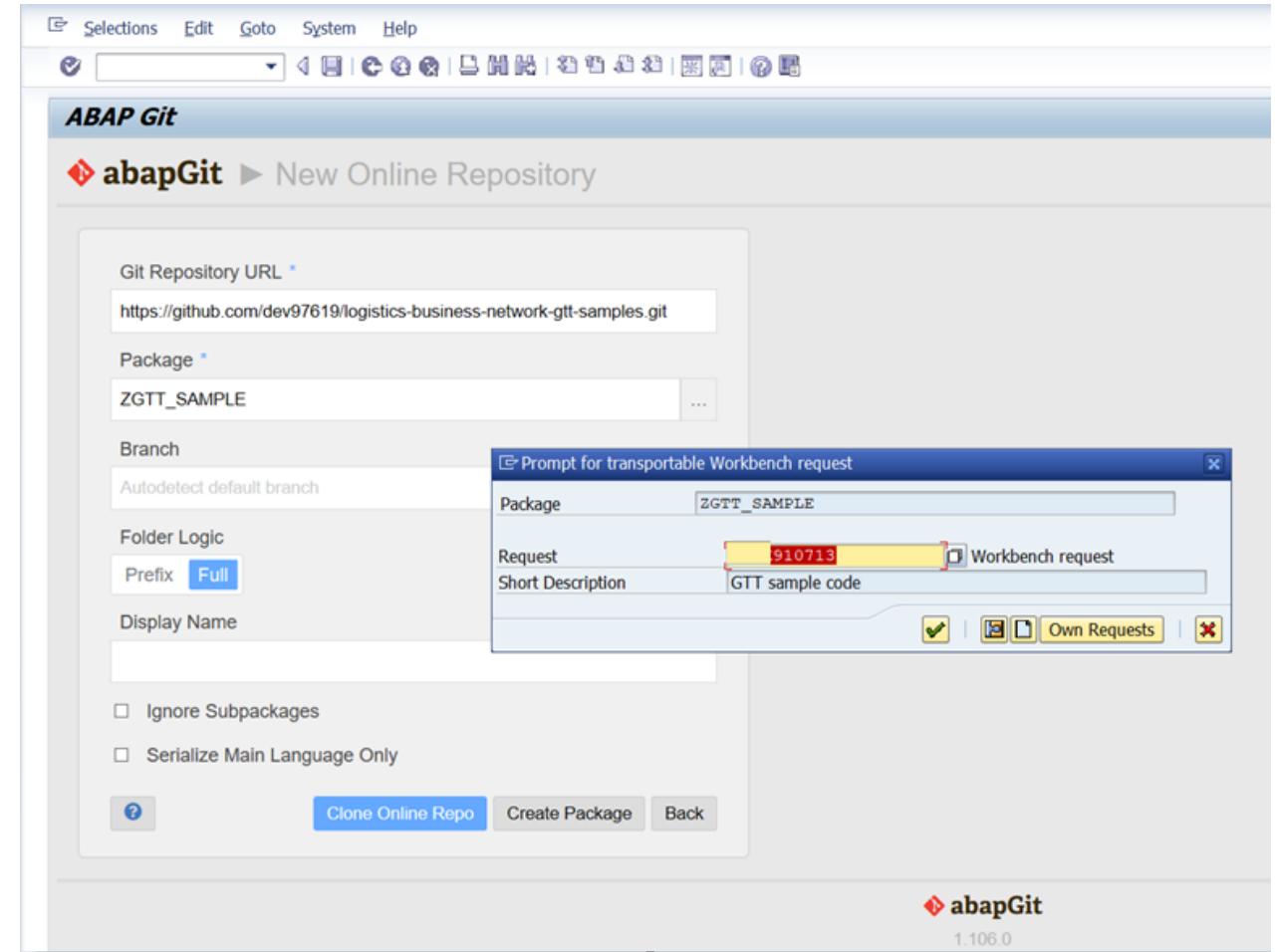
4-6: Set *Full* for **Folder Logic**

4-7: Click **Clone Online Repo** to download the code.



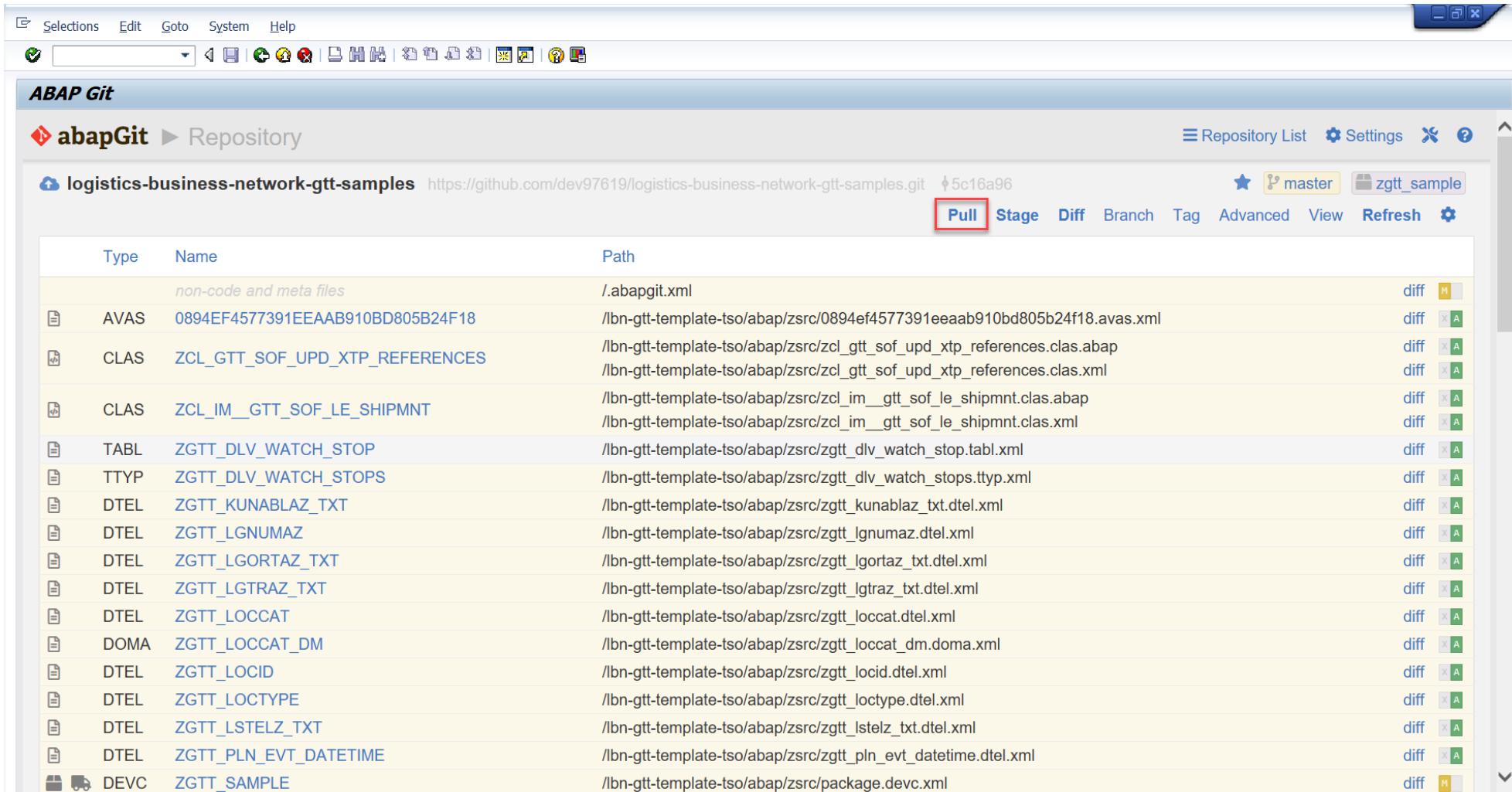
# STEP 4: Download ABAP Code from GitHub

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.

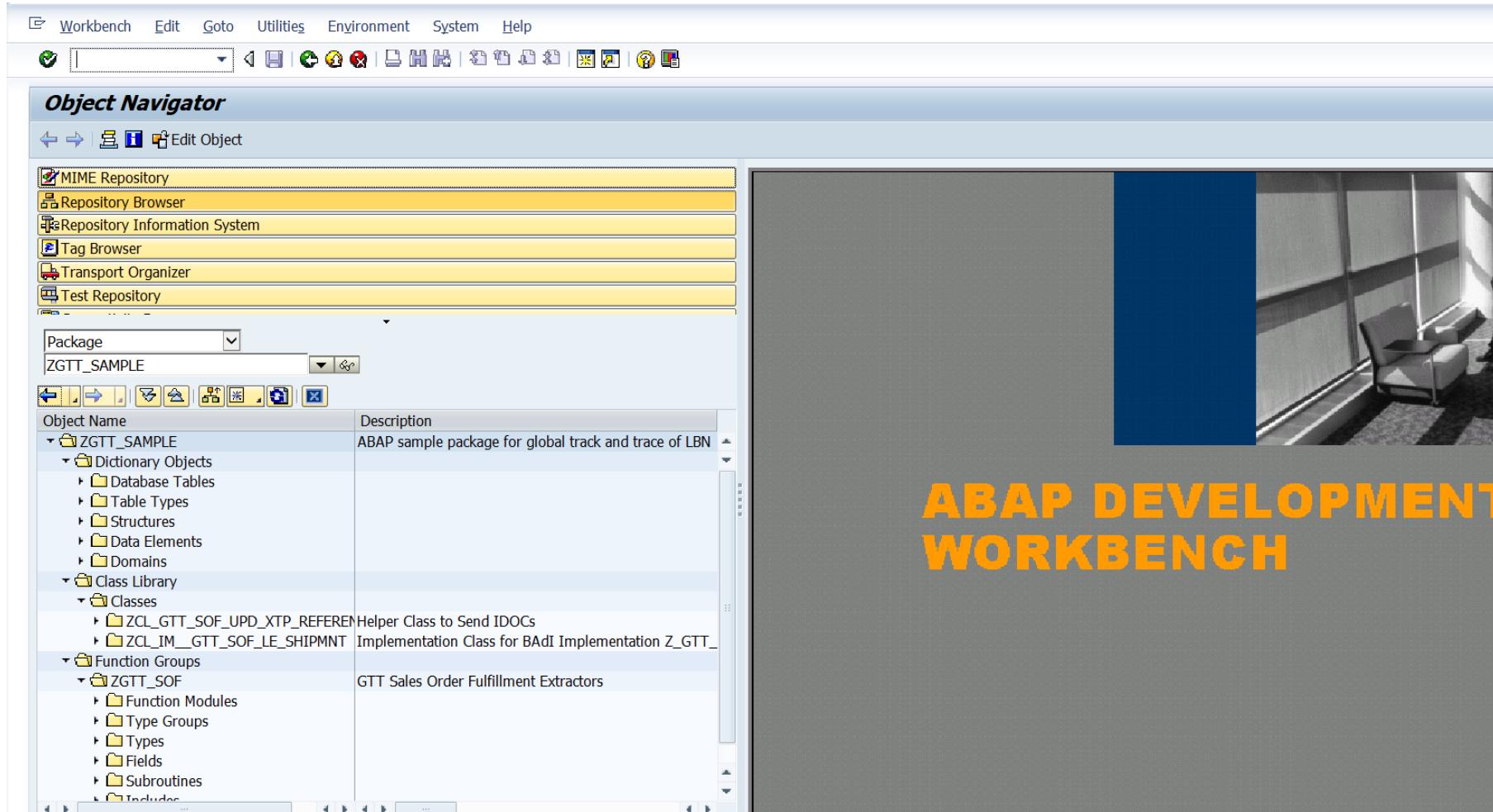


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a sub-header "abapGit ► Repository". The main area displays a table of files from the "logistics-business-network-gtt-samples" repository. The table has columns for Type, Name, and Path. The "Pull" button in the toolbar is highlighted with a red box. The table rows include:

Type	Name	Path	Actions
	non-code and meta files	/.abapgit.xml	diff
AVAS	0894EF4577391EEAAB910BD805B24F18	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	diff
CLAS	ZCL_GTT_SOUPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap	diff
		/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	diff
CLAS	ZCL_IM_GTT_SOFLESHIPMNT	/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap	diff
		/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	diff
TABL	ZGTT_DLW_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/zggt_dlw_watch_stop.tabl.xml	diff
TTYP	ZGTT_DLW_WATCH_STOPS	/lbn-gtt-template-tso/abap/zsrc/zggt_dlw_watch_stops.ttyp.xml	diff
DTEL	ZGTT_KUNABLZ_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_LGORAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgoraz_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
DEVC	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 TSOF)

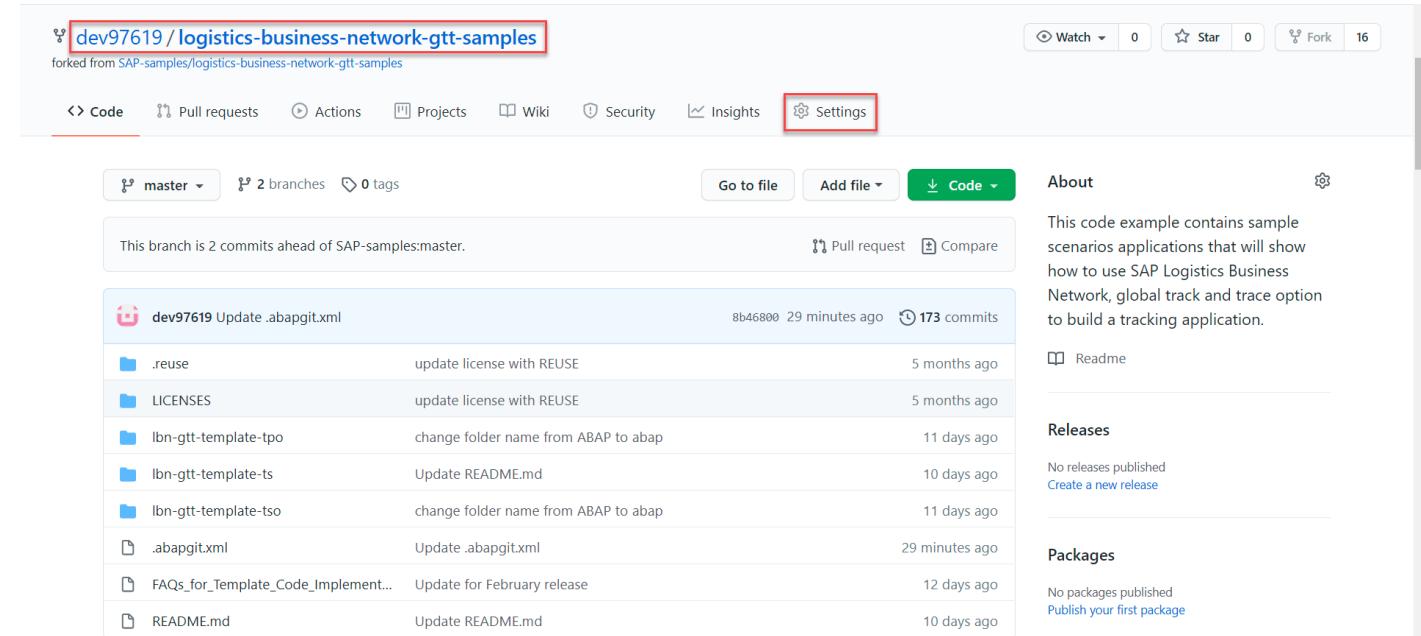


# STEP 1: Delete the User's Account Repository

1-1: Assume you've already installed the sample code of TSOF to your local SAP system with the version of the previous release.

In the latest release, there is some code changes in the 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 repository, and click "Settings".



# STEP 1: Delete the User's Account Repository

1-3: Scroll 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'. 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.

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

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

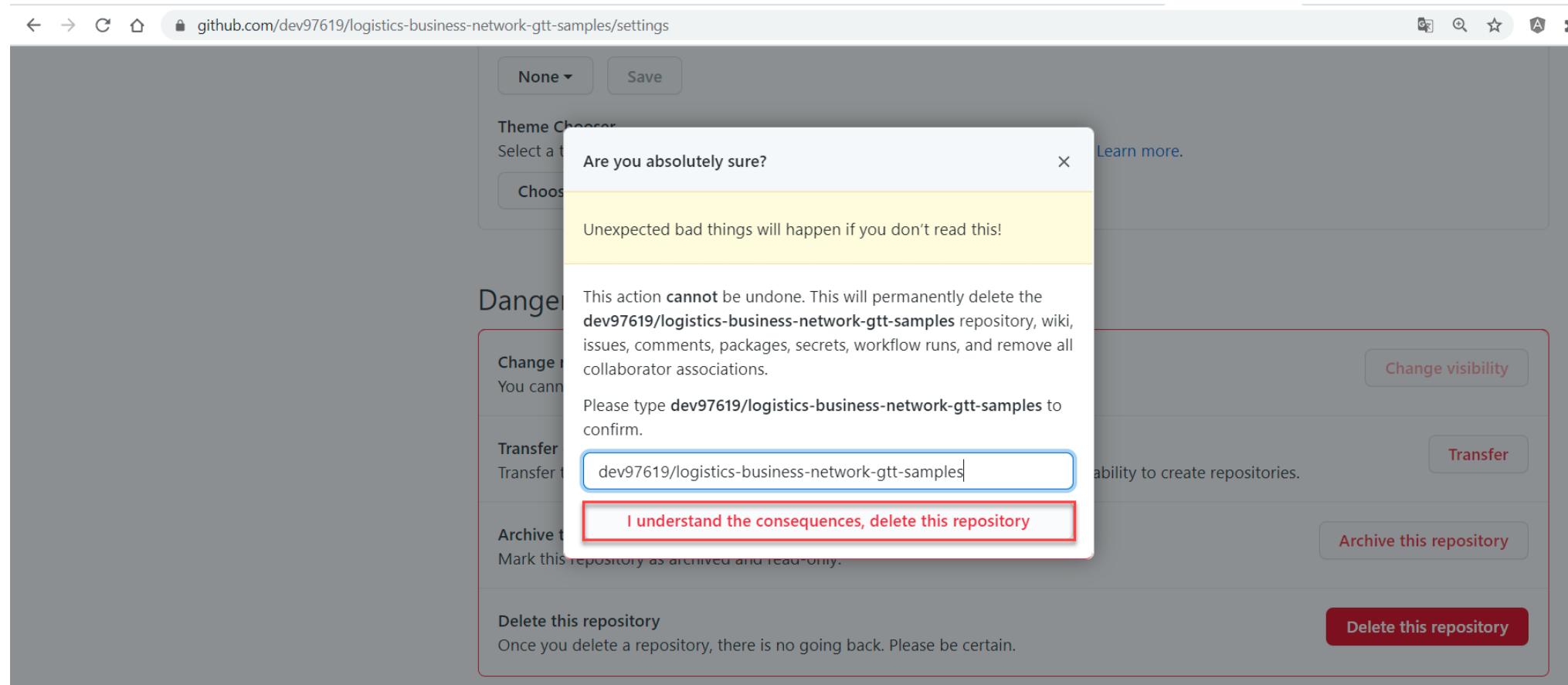
Transfer ownership  
Transfer this repository to another user or to an organization where you have the ability to create repositories. Transfer

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 warning messages. Follow the instructions then click the button "I understand the consequences, delete this repository".



# STEP 1: Delete the User's Account Repository

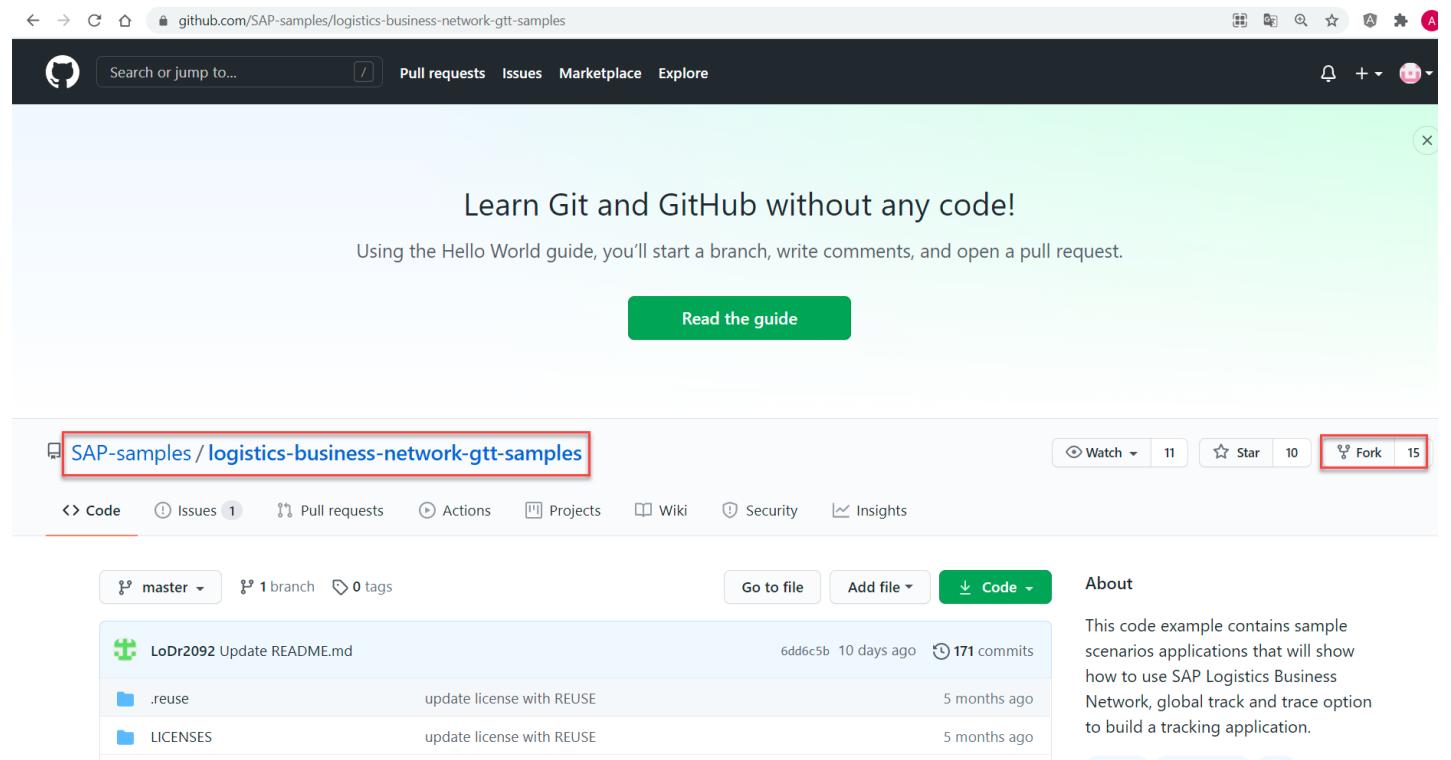
1-5: The user's account repository is deleted.

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 user profile. 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 header bar.

# 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 the user’s own repository.



# STEP 2: Fork Sample Code Repository

2-3: The newest version of the sample codes is 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 ▾

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

.abapgit.xml Update .abapgit.xml 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’

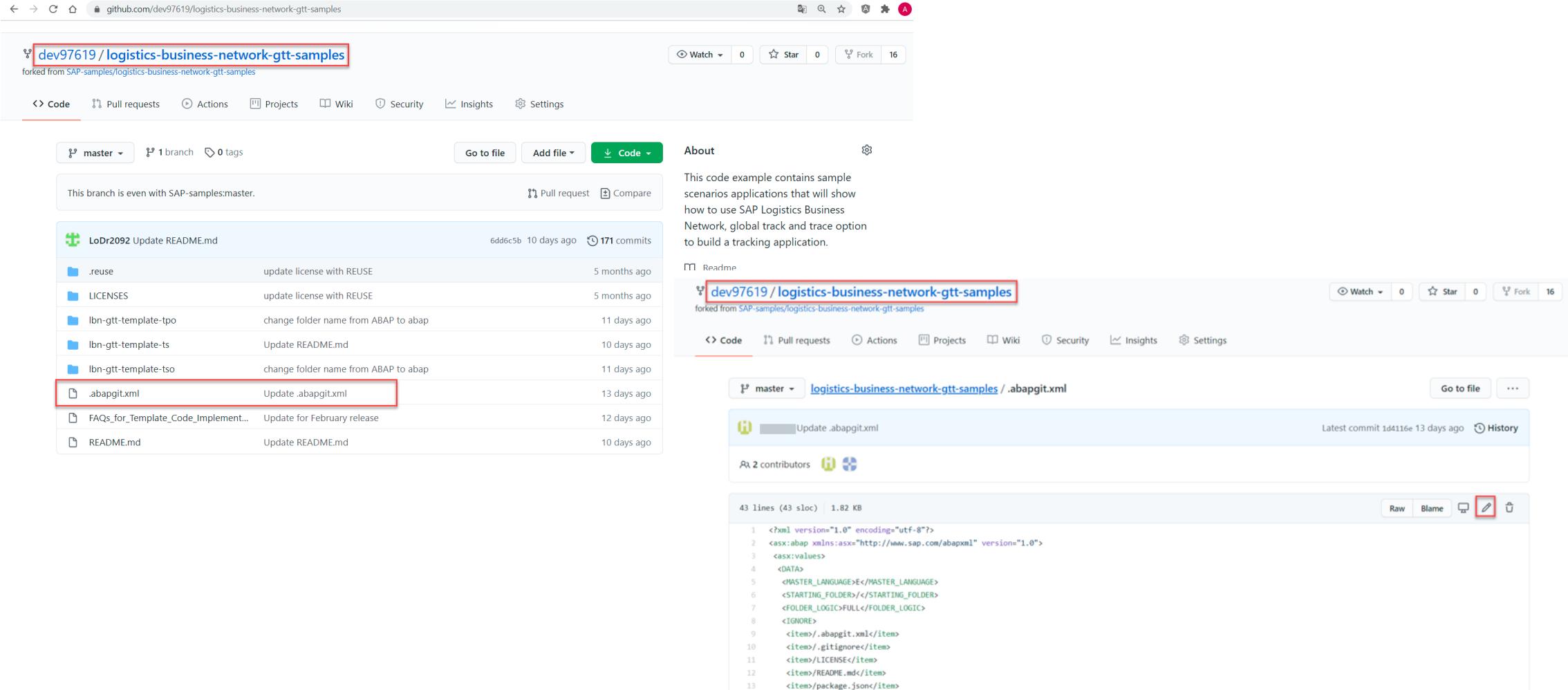
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one for '.abapgit.xml' which is highlighted with a red box. The commit details show it was updated 13 days ago. To the right of the code area, there is an 'About' section describing the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and build instructions. There are also sections for 'Readme', 'Releases', and 'Packages'.

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

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

3-2: Click  button to edit the file.



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected, showing the master branch with 1 branch and 0 tags. A commit by LoDr2092 titled 'Update README.md' is visible. Below it, a commit by the current user titled 'Update .abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgit.xml'. The repository has 171 commits in total. The 'About' section describes the code example as containing sample scenarios applications for SAP Logistics Business Network, global track and trace option to build a tracking application. The '.abapgit.xml' file content is displayed in a code editor, showing XML configuration for ABAP git. The file has 43 lines (43 sloc) and is 1.82 KB. The code editor includes buttons for 'Raw', 'Blame', and 'Edit' (with a red box around it).

```
<?xml version="1.0" encoding="utf-8"?>
<sax:abap xmlns:sax="http://www.sap.com/abapxml" version="1.0">
<sax: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.mdc</item>
<item>/package.json</item>
```

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

3-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with

"<STARTING\_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING\_FOLDER>" as follows.

3-4: Commit change.

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples'. The repository name is highlighted with a red box. The commit dialog for the file '.abapgit.xml' is open, showing the XML code and a 'Commit changes' interface.

**Code View:**

```
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 Dialog:**

Commit changes

Update .abapgit.xml

Add an optional extended description...

-o- Commit directly to the master branch.

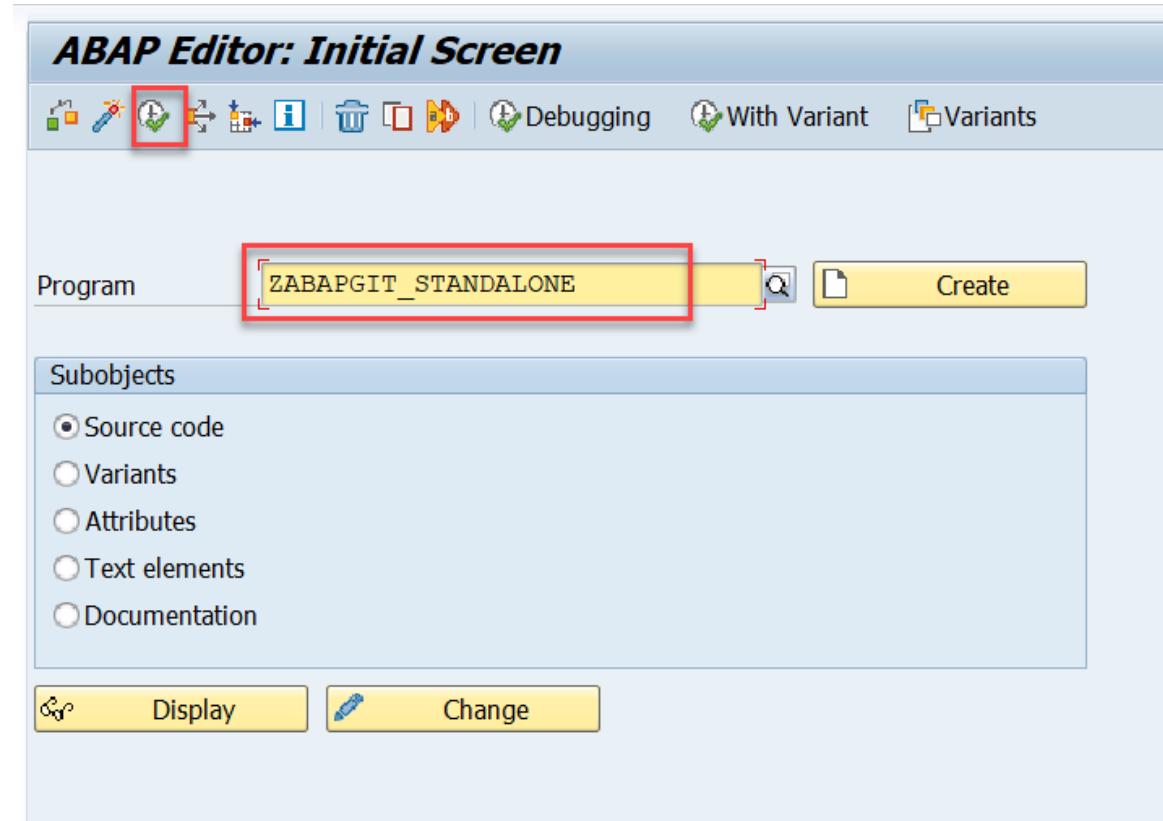
! Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

**Buttons:** Commit changes (highlighted with a red box), 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

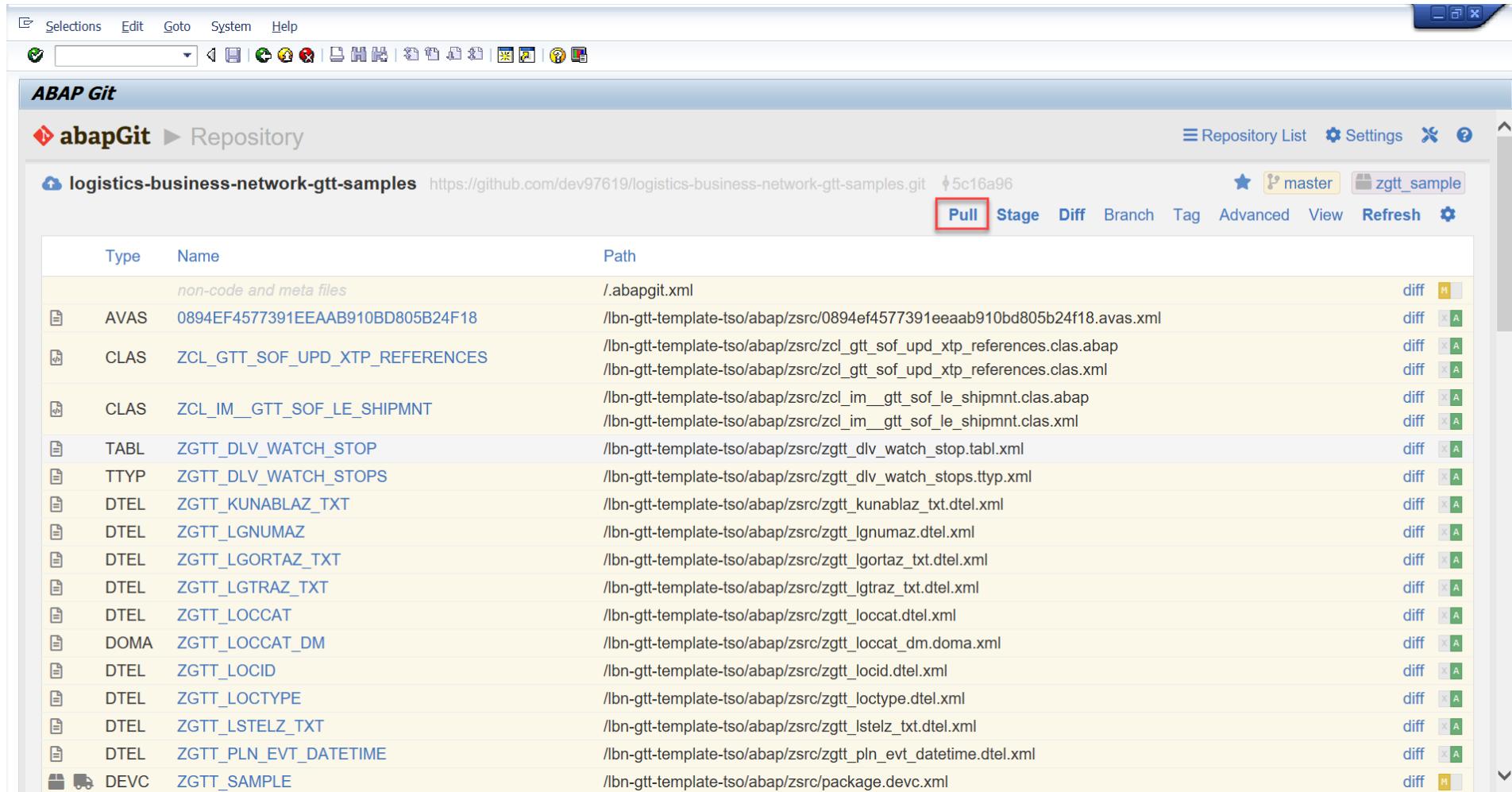
4-3: Check if the URL is changed or not after your recreation of repository copy. Access the TSOF Repository by clicking  button.



The screenshot shows the ABAP Git interface within an SAP application. The title bar includes standard SAP menu items: Selections, Edit, Goto, System, Help. Below the menu is a toolbar with various icons. The main area is titled "ABAP Git" and shows a "Repository List". A navigation bar at the top of the list area includes "abapGit" and "Repository List", along with buttons for "New Online", "New Offline", "Settings", and help. A filter bar below the navigation bar allows filtering by "Favorites" and "Detail". The repository list table has columns: Name, Url, Package, Branch, and Action. One row is visible: "logistics-business-network-gtt-samples" with Url "github.com/dev97619/logistics-business-network-gtt-samples.git", Package "zgtt\_sample", Branch "master", and Action buttons for "Check", "Stage", "Patch", "Settings", and a red-bordered ">". At the bottom of the list area, there is a footer with the "abapGit" logo and version "1.106.0", and a status message "js: OK".

# 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 and a menu bar with 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the toolbar, the title bar says 'ABAP Git' and 'abapGit Repository'. The main area displays a table of files in the 'logistics-business-network-gtt-samples' repository. The table has columns for 'Type', 'Name', and 'Path'. The 'Pull' button in the toolbar is highlighted with a red box. The table data is as follows:

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_SOUPD_XTP_REFERENCES	//lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	A A
CLAS	ZCL_IM_GTT_SOFL_SHIPMNT	//lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	A A
TABL	ZGTT_DLW_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zgtt_dlw_watch_stop.tabl.xml	A
TTYP	ZGTT_DLW_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zgtt_dlw_watch_stops.ttyp.xml	A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_kunabla_z_txt.dtel.xml	A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgnumaz.dtel.xml	A
DTEL	ZGTT_LGORATAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgorataz_txt.dtel.xml	A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgtraz_txt.dtel.xml	A
DTEL	ZGTT_LOCAT	//lbn-gtt-template-tso/abap/zsrc/zgtt_locat.dtel.xml	A
DOMA	ZGTT_LOCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zgtt_locat_dm.doma.xml	A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zgtt_locid.dtel.xml	A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zgtt_loctype.dtel.xml	A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lstelz_txt.dtel.xml	A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zgtt_pln_evt_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 (Only for TPOF)



# STEP 1: Fork Sample Code Repository

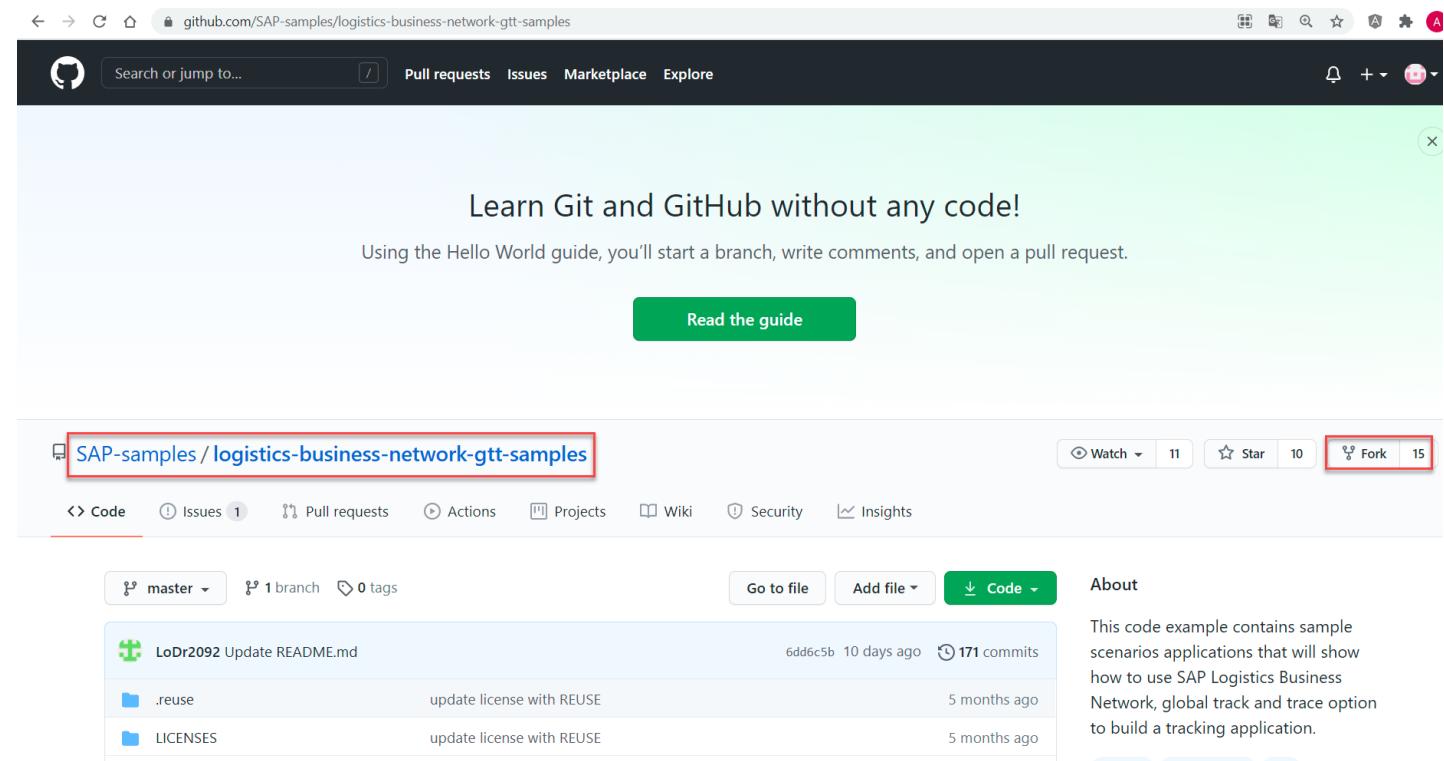
## Prerequisite:

You must have already completed procedure C1 and have installed ABAPGit and the sample code of TSOF to your local SAP system.

To install the TPOF do the following:

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

1-2. Click the “Fork” button, it will copy the newest version of sample codes into the user’s account and meanwhile it will navigate to the user’s own repository.



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

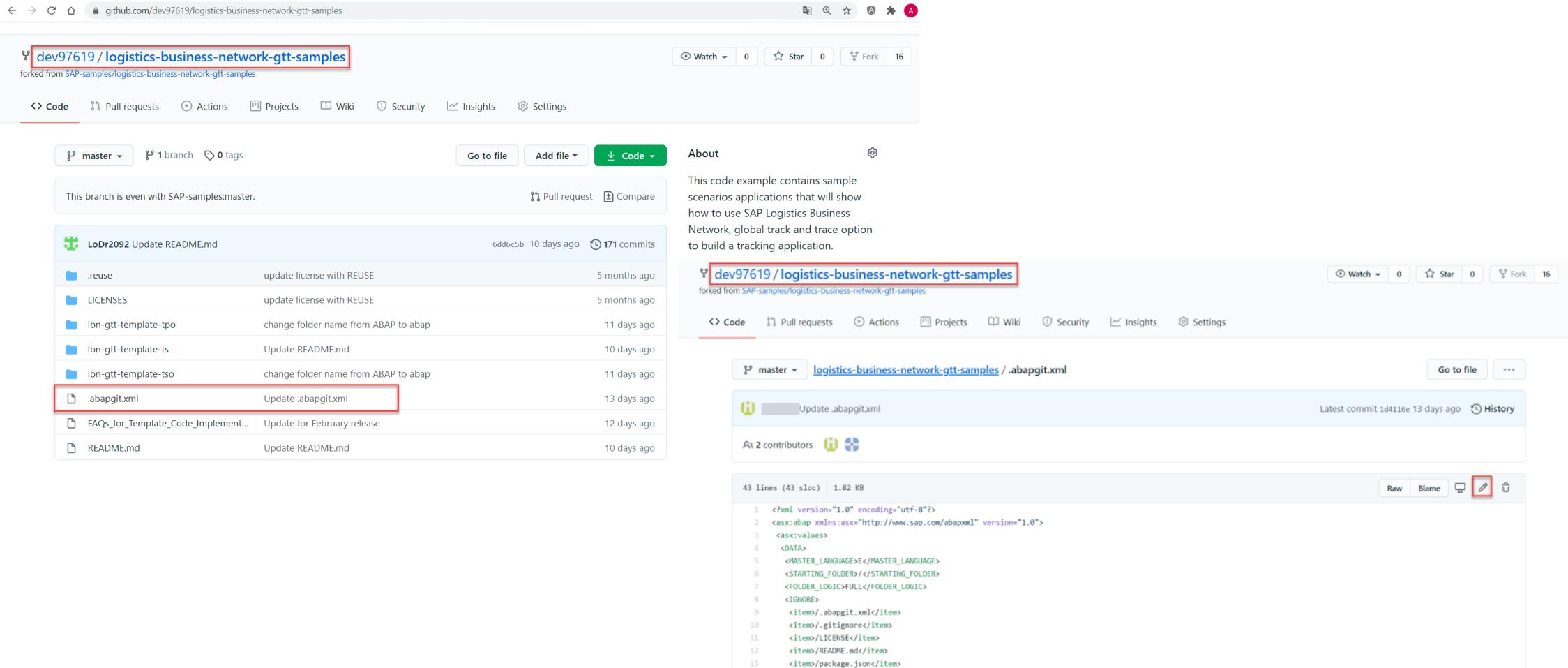
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one for '.abapgit.xml' which is highlighted with a red box. The commit details show it was updated 13 days ago. To the right of the code area, there is an 'About' section describing the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and build instructions. There are also sections for 'Readme', 'Releases', and 'Packages'.

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

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

2-2: Click  button to edit the file.



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

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <DATA>
      <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
      <STARTING_FOLDER>/</STARTING_FOLDER>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
      <IGNORE>
        <item>/.abapgit.xml</item>
        <item>/.gitignore</item>
        <item>LICENSE</item>
        <item>README.md</item>
        <item>package.json</item>
      </IGNORE>
    </DATA>
  </asx:values>
</asx:abap>
```

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

2-3: Replace the line "<STARTING\_FOLDER>/</STARTING\_FOLDER>" with

"<STARTING\_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING\_FOLDER>" as follows.

2-4: Commit change.

The screenshot shows a GitHub commit dialog for the file '.abapgit.xml' in the repository 'logistics-business-network-gtt-samples'. The file content is displayed on the left, and the commit message is on the right. A red box highlights the line of code being modified.

**.abapgit.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>
```

**Commit changes:**

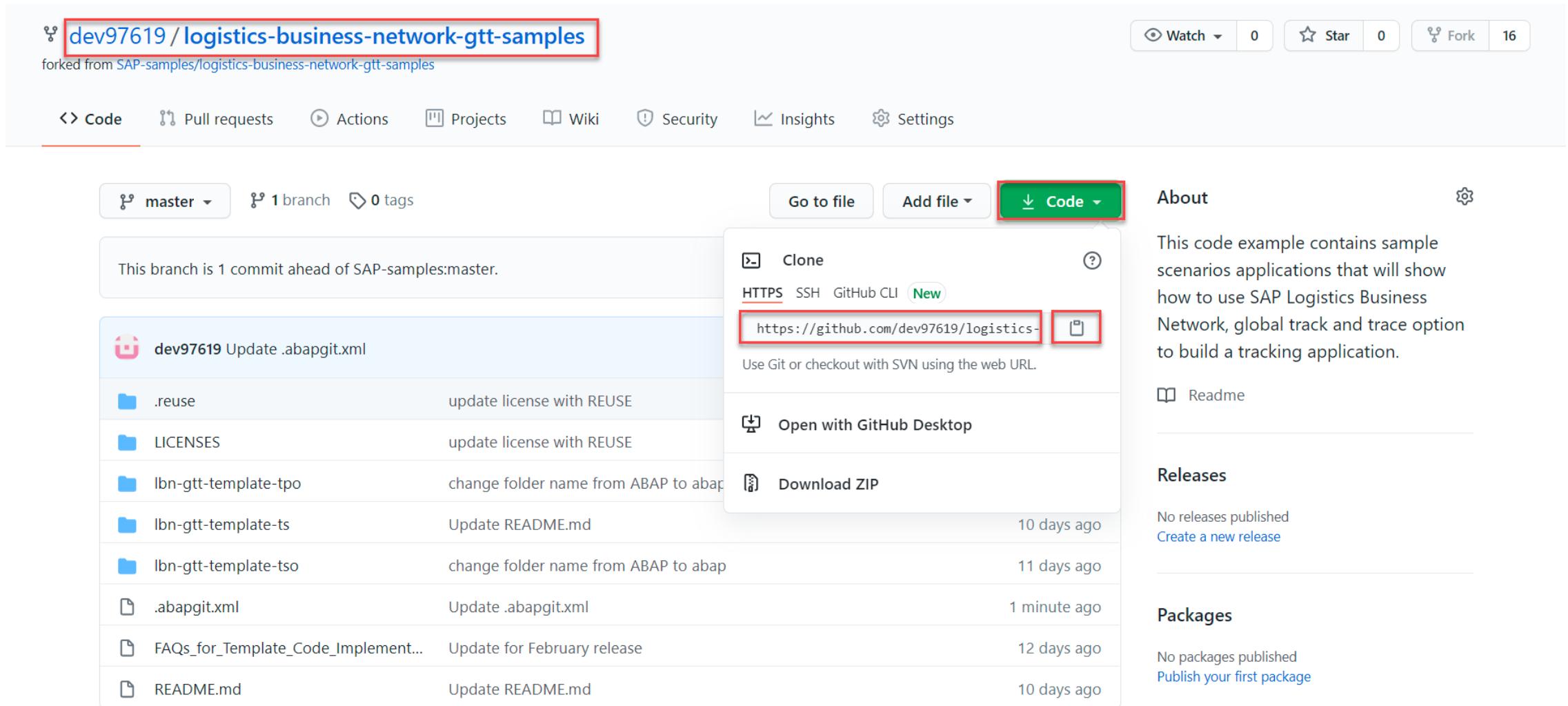
Update .abapgit.xml  
Add an optional extended description...

-o Commit directly to the master branch.  
 ↗ Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

**Buttons:** Commit changes (highlighted with a red border) and Cancel

## 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 been forked from SAP-samples/logistics-business-network-gtt-samples. The 'Code' tab is selected. A dropdown menu is open over the 'Clone' link, with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples> highlighted and a copy icon () overlaid on it. The repository has 1 branch and 0 tags. The master branch is 1 commit ahead of SAP-samples:master. The commit history includes several changes to the '.abapgit.xml' file and other documentation files. The 'About' section describes the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options. The 'Readme' and 'Releases' sections are also visible.

Code example:

```
https://github.com/dev97619/logistics-business-network-gtt-samples
```

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

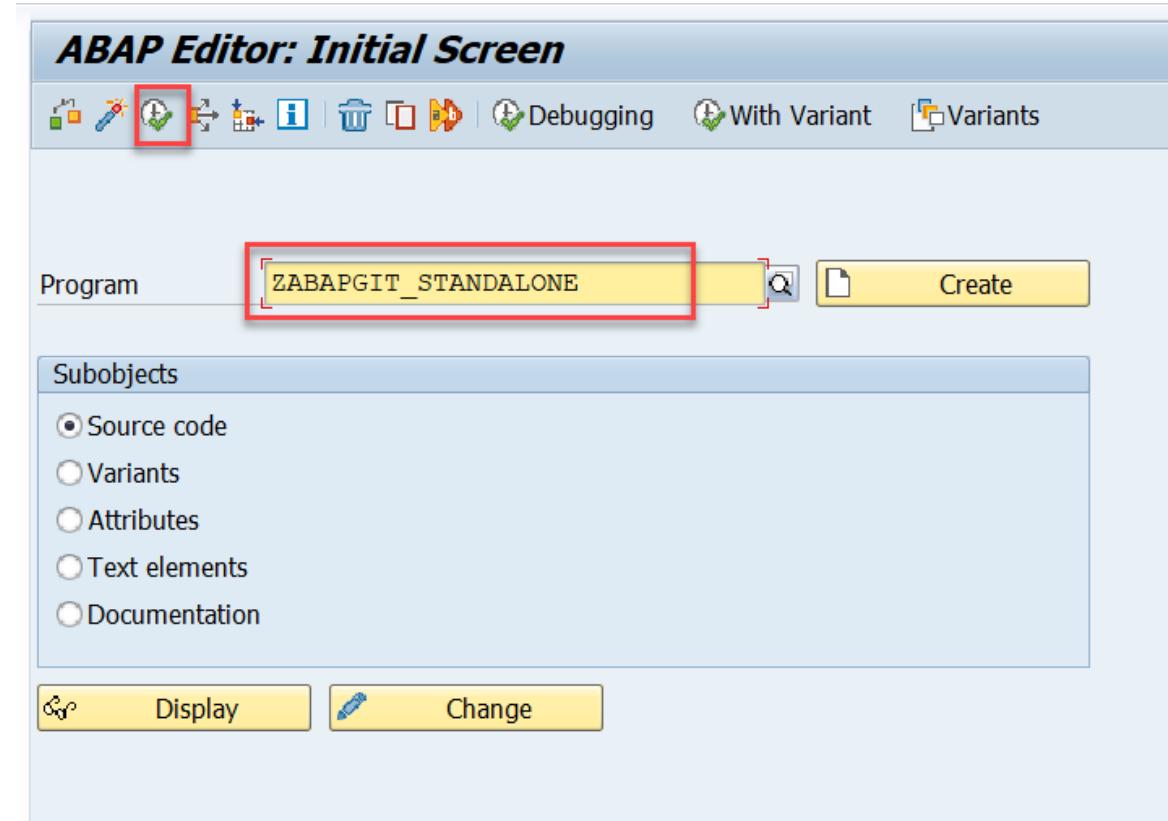
Releases

Packages

## STEP 3: Remove TSOF 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 TSOF Repository in ABAPGit

3-3: Access the TSOF Repository by clicking  button.



The screenshot shows the ABAPGit interface with the following details:

- Toolbar:** Selections, Edit, Goto, System, Help.
- Repository List:** **abapGit** ► Repository List
- Filter:**  ✓ Only Favorites | ✓ Detail
- Table Headers:** Name, Url, Package, Branch, Action
- Table Data:**

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 
- Footer:** abapGit 1.106.0 js: OK

# STEP 3: Remove TSOF Repository in ABAPGit

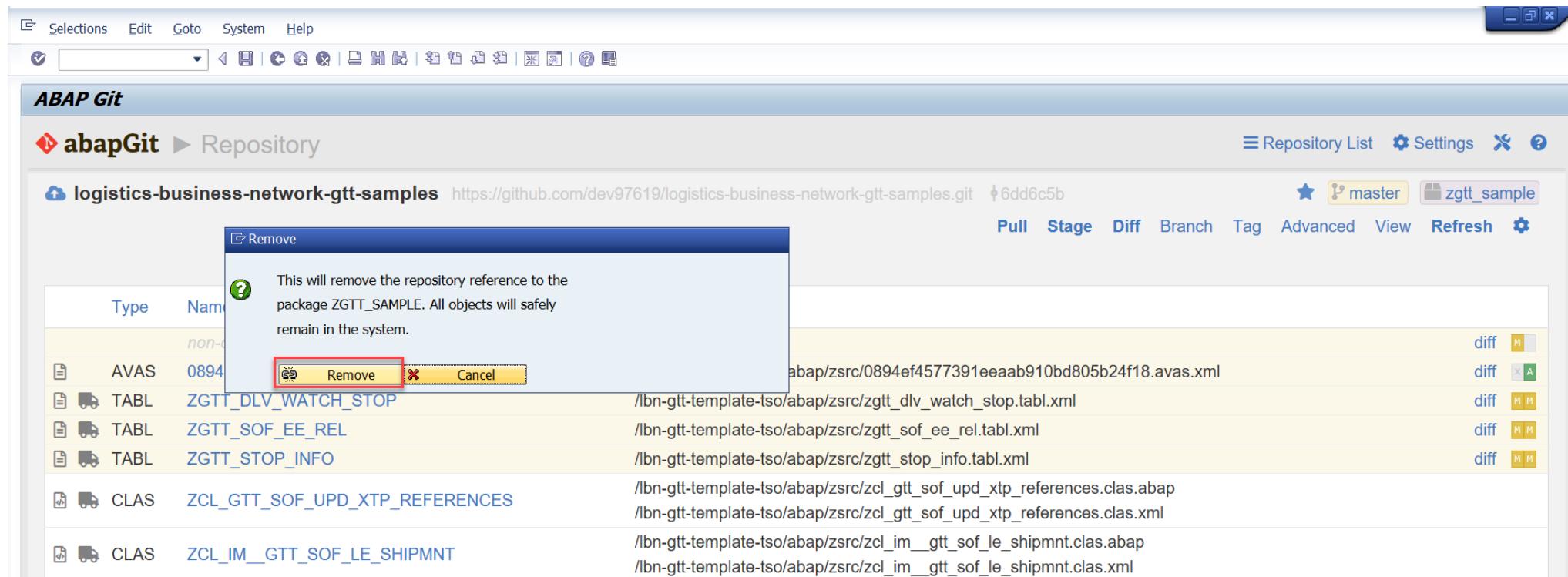
3-4: Under the “Advanced” menu, choose and click “Remove”.

The screenshot shows the ABAP Git interface for managing repositories. The repository listed is "logistics-business-network-gtt-samples" with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples.git>. The commit hash shown is 6dd6c5b. The toolbar at the top includes Selections, Edit, Goto, System, Help, and various icons for file operations. Below the toolbar is a menu bar with ABAP Git, abapGit, Repository, and other options. The main area displays a table of objects with columns for Type, Name, and Path. The "Advanced" menu is open, showing a list of options: Reset Local (Force Pull), Checkout commit, Background Mode, Change Remote, Make Off-line, Force Stage, Transport to Branch, Add all objects to transport request, Syntax Check, Run Code Inspector, Update Local Checksums, Beta - Data, Remove, and Uninstall. The "Remove" option is highlighted with a red box.

Type	Name	Path
<i>non-code and meta files</i>		
AVAS	0894EF4577391EEAB910BD805B24F18	./abapgit.xml
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeab910bd/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
CLAS	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 TSOF Repository in ABAPGit

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

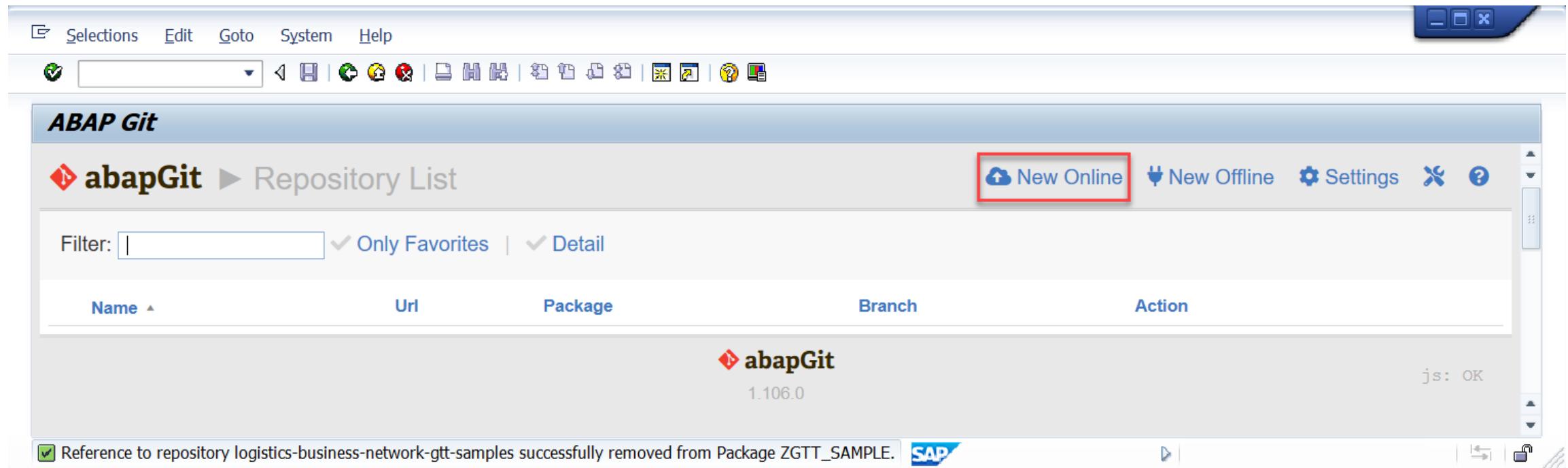


3-6: After repository removal you will see the following message:



# STEP 4: Download TPOF Code from GitHub

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



# STEP 4: Download TPOF 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>

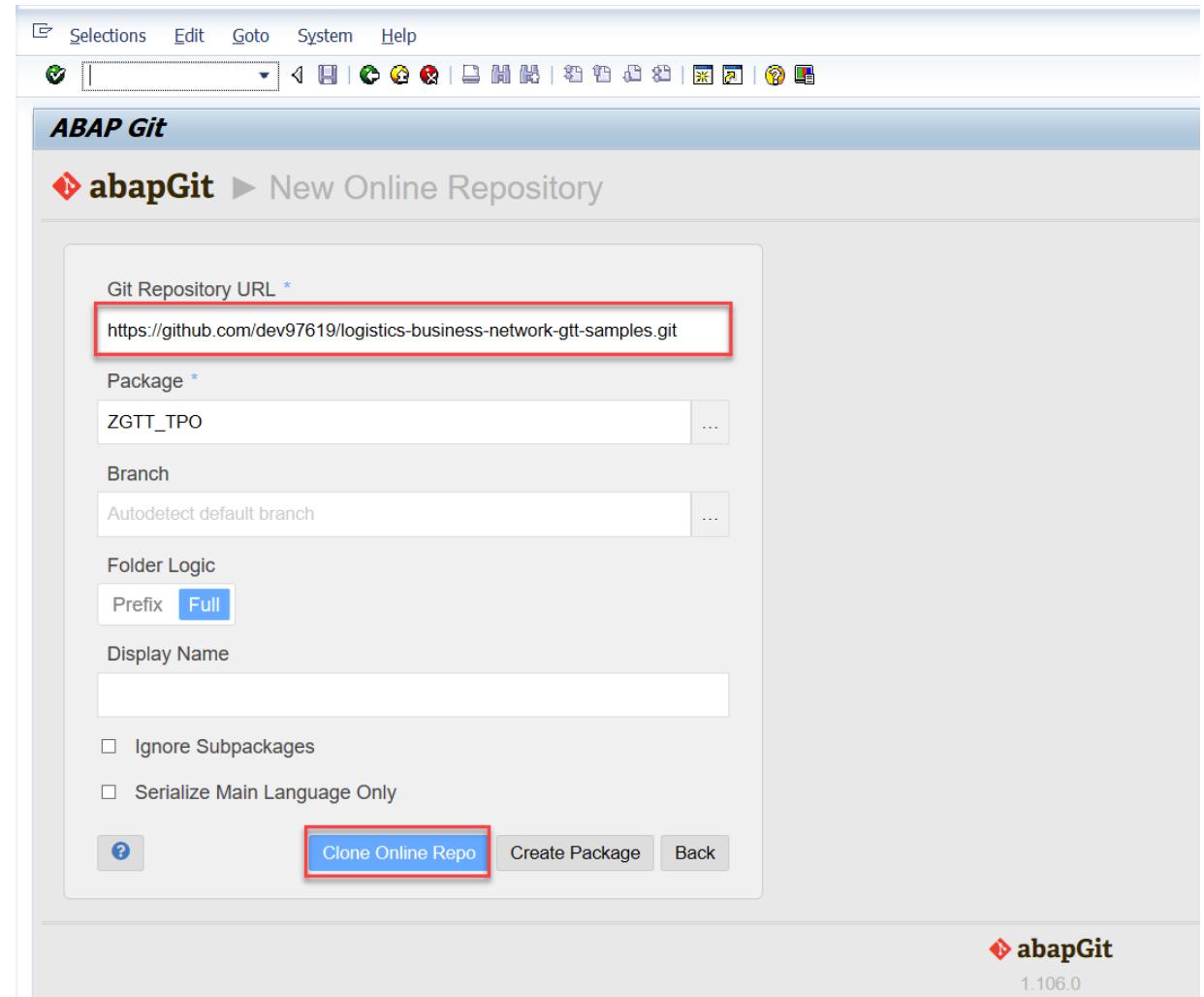
## Caution:

This URL is the user's account 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.

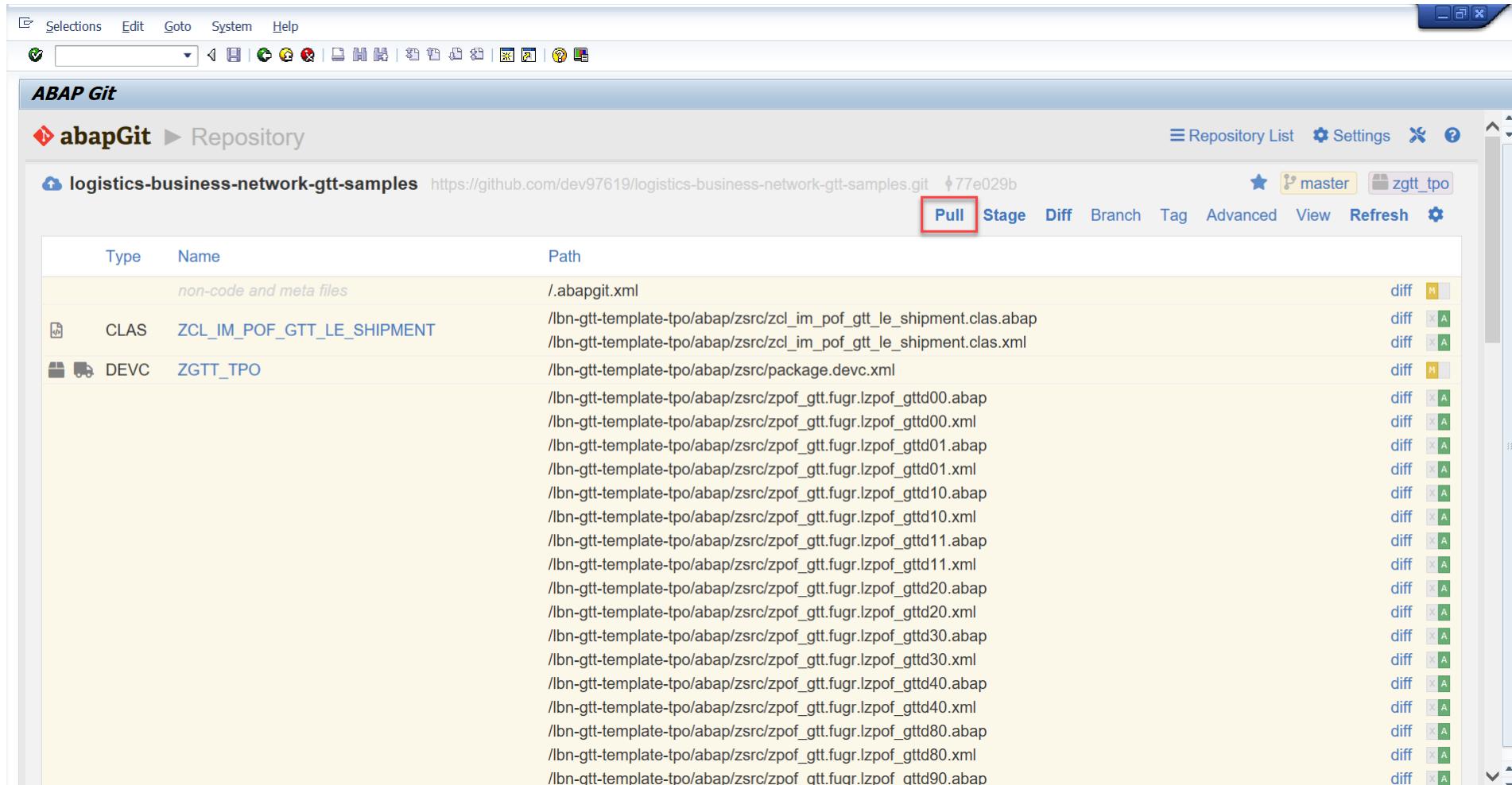
4-4: Set *Full* for **Folder Logic**

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 and a menu bar with 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the toolbar, the title bar says 'ABAP Git' and 'abapGit Repository'. Underneath, it shows the repository 'logistics-business-network-gtt-samples' with the URL 'https://github.com/dev97619/logistics-business-network-gtt-samples.git' and a commit hash '77e029b'. To the right of the URL are icons for a star, a master branch, and a tag labeled 'zgtt\_tpo'. A navigation bar below the title bar includes 'Repository List', 'Settings', 'Pull' (which is highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a gear icon. The main area is a table with columns 'Type', 'Name', and 'Path'. It lists several files: a non-code file '.abapgit.xml', a class 'ZCL\_IM\_POF\_GTT\_LE\_SHIPMENT' with two associated XML files, and a package 'ZGTT\_TPO' containing numerous ABAP and XML files related to GTT (Global Transportation Template) logic. Each file entry includes a 'diff' link and a small status indicator icon.

# C) Download ABAP Code from GitHub

## C4. Initial Download ABAP Code from GitHub (include TSOF/TPOF/TS)



# STEP 1: Install ABAPGit

You need to install ABAPGit before downloading the 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

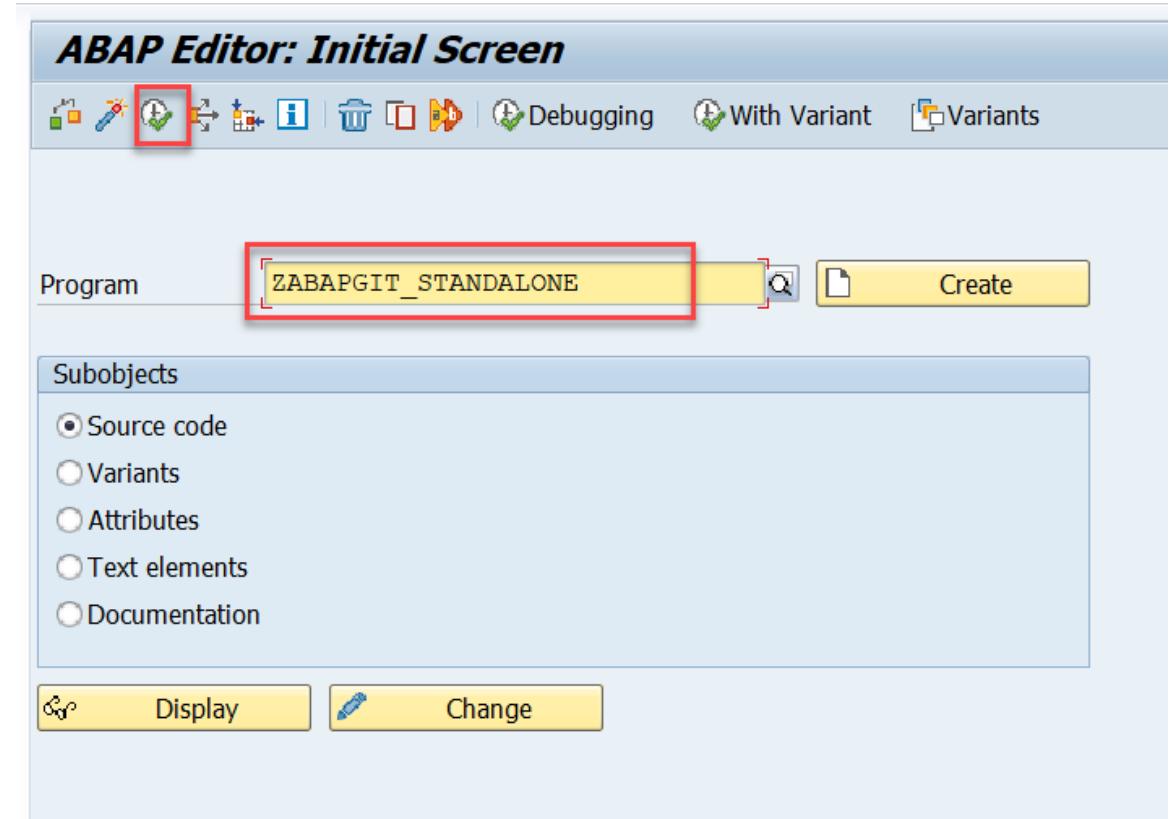
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

## STEP 2: Download ABAP Code

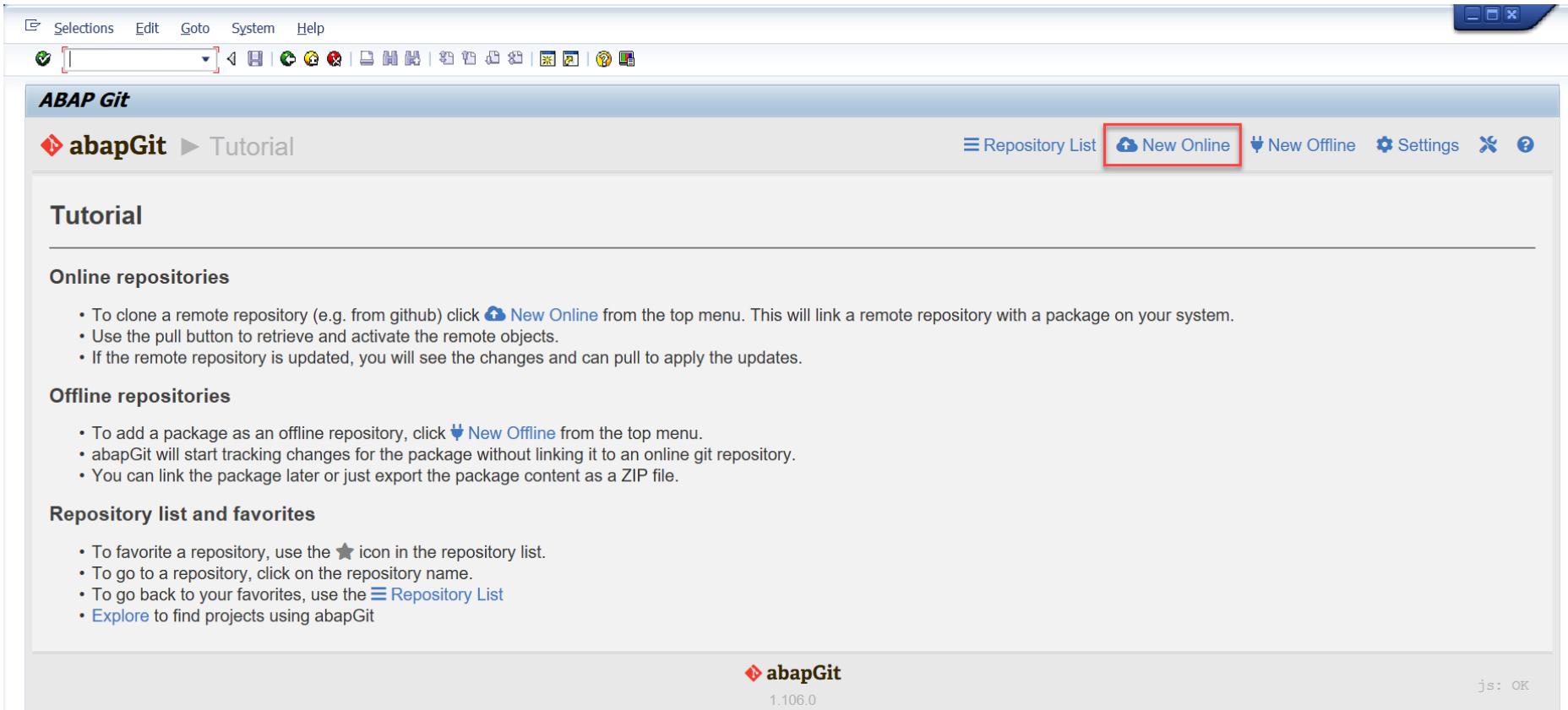
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,  
**ZABAPGIT\_STANDALONE**.

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



# STEP 2: Download ABAP Code

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



The screenshot shows the abapGit application window. At the top, there is a menu bar with 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the menu is a toolbar with various icons. The main title bar says 'ABAP Git'. Underneath it, the title 'abapGit ► Tutorial' is displayed. On the right side of the title bar, there are several buttons: 'Repository List', 'New Online' (which is highlighted with a red box), 'New Offline', 'Settings', and others. The main content area is titled 'Tutorial'. It contains sections for 'Online repositories' and 'Offline repositories', each with a bulleted list of instructions. At the bottom, there is a footer with the 'abapGit' logo, the version '1.106.0', and the text 'js: OK'.

**Online repositories**

- To clone a remote repository (e.g. from github) click **New Online** from the top menu. This will link a remote repository with a package on your system.
- Use the pull button to retrieve and activate the remote objects.
- If the remote repository is updated, you will see the changes and can pull to apply the updates.

**Offline repositories**

- To add a package as an offline repository, click **New Offline** from the top menu.
- abapGit will start tracking changes for the package without linking it to an online git repository.
- You can link the package later or just export the package content as a ZIP file.

**Repository list and favorites**

- To favorite a repository, use the **★** icon in the repository list.
- To go to a repository, click on the repository name.
- To go back to your favorites, use the **Repository List**
- [Explore](#) to find projects using abapGit

## STEP 2: Download ABAP Code

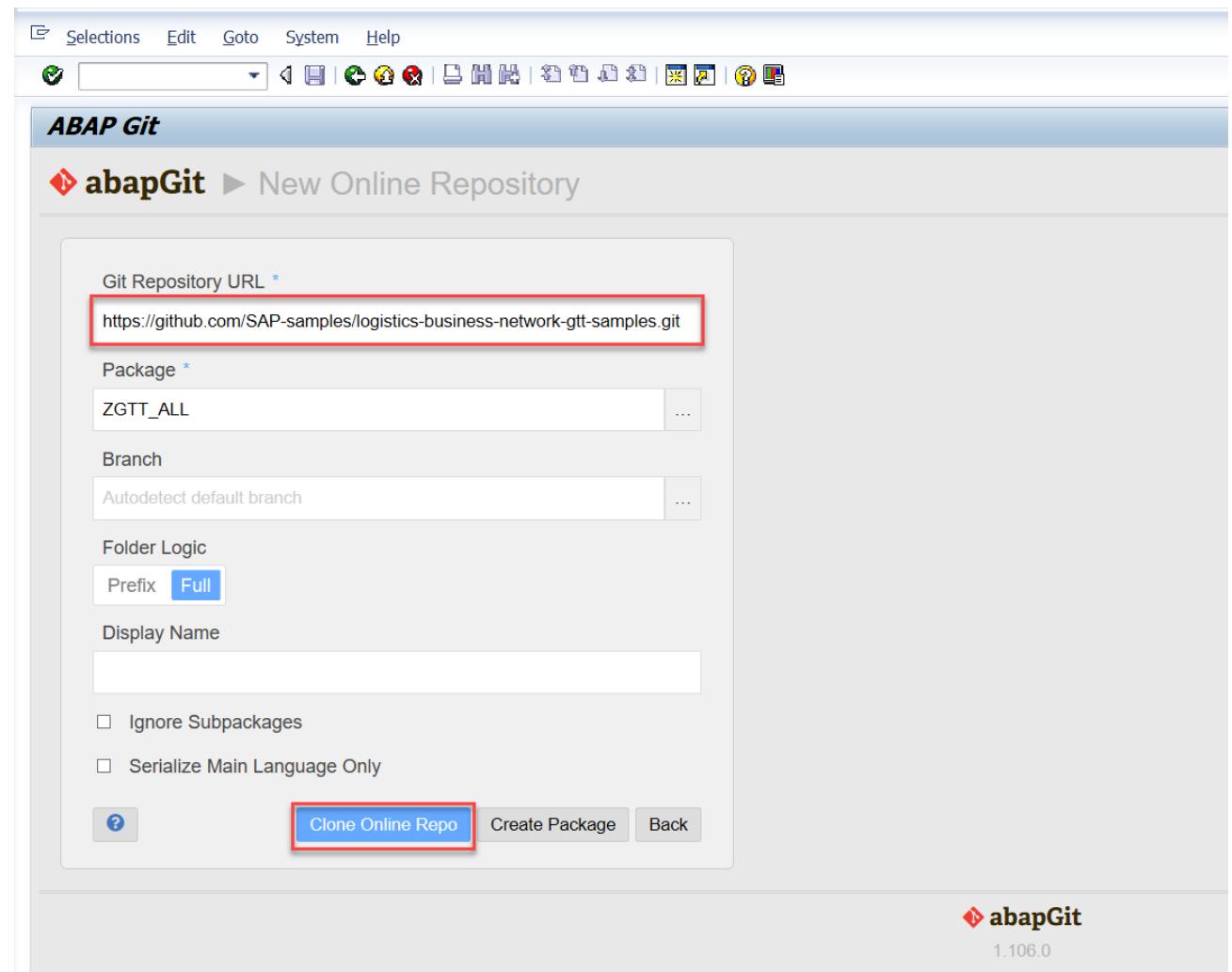
2-4: Fill in the **Git Repository URL**:

<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.

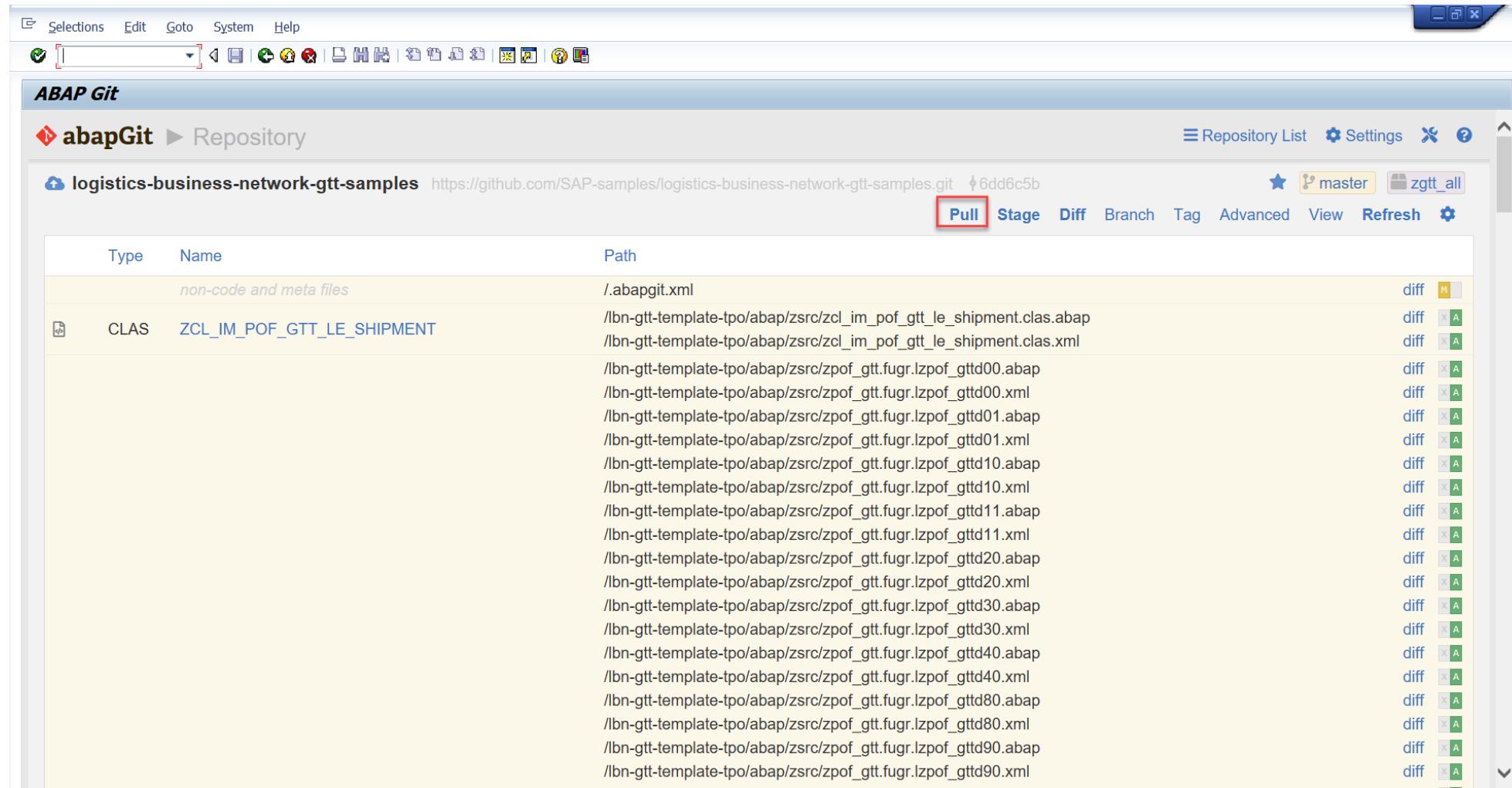
2-6: Set *Full* for **Folder Logic**

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.

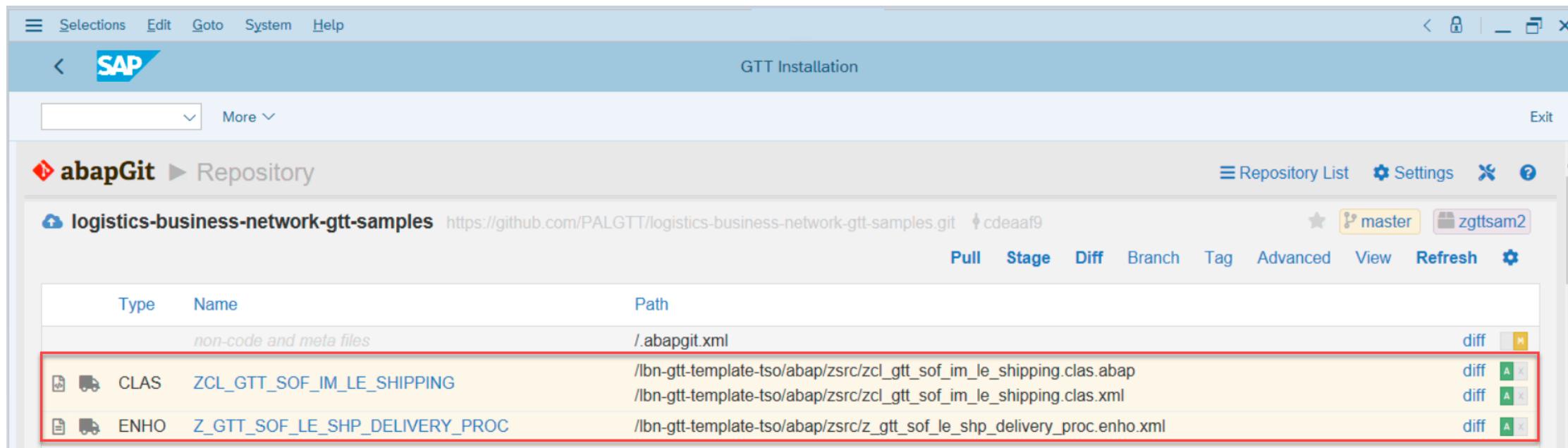


The screenshot shows the ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation: "abapGit > Repository". Underneath is a sub-header for the repository "logistics-business-network-gtt-samples" with its URL and a commit hash. To the right of the sub-header are buttons for "master" branch and "zgtt\_all" tag. A navigation bar below the sub-header includes "Pull", "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The main area is a table with columns "Type", "Name", and "Path". The "Type" column shows mostly "non-code and meta files" and one entry for "CLAS". The "Name" column lists file names like ".abapgit.xml", "ZCL\_IM\_POF\_GTT\_LE\_SHIPMENT", and various ABAP and XML files under the path "/bn-gtt-template-tpo/abap/zsrc/". The "Path" column shows the full file paths. To the right of each row are "diff" buttons and small colored boxes indicating changes (yellow for M, blue for A).

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

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

**Symptom:** If the user updates the ABAP code by report ZABAPGIT\_STANDALONE, there will be a code difference as below:



The screenshot shows the SAP GTT Installation interface. At the top, it says "GTT Installation". Below that, it shows a GitHub repository "logistics-business-network-gtt-samples" with the URL <https://github.com/PALGTT/logistics-business-network-gtt-samples.git>. The commit hash is cdeaaf9. The branch is master, and the tag is zgttsam2. The interface has tabs for Pull, Stage, Diff, Branch, Tag, Advanced, View, Refresh, and Settings. A red box highlights the "Diff" tab. The main table lists files and their paths:

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

This is because the enhancement implementation Z\_GTT\_SOFTWARE\_SHIPPING\_DELIVERY\_PROC is already obsolete and removed from the GitHub, the report ZABAPGIT\_STANDALONE cannot remove the object which was already deleted in GitHub.

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

## **Solution:**

### **Option 1:**

1-1) Deactivate the BADI implementation.

### **Option 2:**

2-1) Delete the enhancement implementation Z\_GTT\_SOF\_LE\_SHP\_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: Remotely Deleted Object Cannot be Synchronized to the Local Object

For option 1: Use transaction code SE19 and deactivate the BADI implementation.

The screenshot shows two SAP application windows. The top window is titled "BAdI Builder: Initial Screen for Implementations". It has a search bar and several menu items: Check, Delete implementation, Copy implementation, Rename implementation, Application help, and More. Below the menu is a section titled "Edit Implementation" with a radio button for "New BAdI" selected. The "Enhancement Implementation" field contains the value "Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC", which is highlighted with a red box. The bottom window is titled "Enhancement Implementation Z\_GTT\_SOF\_LE\_SHP\_DELIVERY\_PROC Display". It also has a search bar and menu items: Previous Object, Next Object, Display >> Change, Other Object..., Check, Activate, Where-Used List, Display Object List, and Fullscreen On/Off. The "Implementation Elements" tab is selected. In the "Properties" section, the "BAdI Implementation" field is set to "Z\_GTT\_SOF\_IM\_LE\_SHIPPING" and the "Description" field is "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list". Under "Runtime Behavior", there is a checked checkbox labeled "Implementation is active" and a note below it stating "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 the same as the name which is defined in the corresponding model in Manage Models application in SAP Business Network Global Track and Trace Version 2.

The image shows two screenshots of the SAP Business Network Global Track and Trace Version 2 interface.

**Left Screenshot: Define Application Object Types - Details**

- Dialog Structure:** Define Used Business Process, Define Application Object Type (selected), Define Event Types.
- Business Process Type:** ESC\_SORDER
- Application Object Type:** ZGTT\_SO\_INT\_HD (highlighted with a red box)
- Description:** Extract sales order header information to Global Track and Trace Integration
- Text:** Sales Order Header
- Object Identification Tab:** Selected tab.
- Method for determination of AOID:** AOID Method: Determine from Field
- Application Object ID Source:**
  - First Field to Build Appl. Obj. ID: Cntrl Tab. Type: 1 Main Object Table, AO ID Field: VBELN
  - Second Field to Build Appl. Obj. ID: Cntrl Tab. Type: (empty), AO ID Field: (empty)
- Determine AOID By Function:** (empty)

**Right Screenshot: Manage Models - IDOC Integration**

- Tracked Process:** SalesOrder
- Integration Switch:** ON
- Tracked Process Mapping:** ERP Object Type: Others, Application Object Type: ZGTT\_SO\_INT\_HD (highlighted with a red box)
- Tracked Process / Events (2):**

Name	IDOC	Event Code
SalesOrderEvent	E1EHPAO	
Event Types	E1EVMDR02	
- User Model Fields:**

Field	IDOC Segment	IDOC Field
salesOrderNo	E1EHPCP	YN_SO_NO
shipToPartyId	E1EHPCP	YN_SO_SHIPTO
netValue	E1EHPCP	YN_NET_VALUE
currency	E1EHPCP	YN_NET_VALUE_CURRENCY

## 2: Maintain Tracking ID Type

In the AOT you maintained, make sure the name of Tracking ID Type is as same as the name defined in the corresponding process type of the model in Manage Models app in SAP Business Network Global Track and Trace Version 2.

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

The image consists of two side-by-side screenshots from SAP.

**SAP AOT Screenshot:** This screenshot shows the "Display View 'Define Application Object Types': Details" screen. It includes sections for "Bus. Proc. Type" (ESC\_SORDER), "Appl. Obj. Type" (ZGTT\_SO\_INT\_HD), and "Text" (Sales Order Header). Under "Parameter Setup", the "Tracking ID Setup" section is highlighted, showing "TrkID Method: B Determine from Field", "Tr.ID Tab. Type: 1 Main Object Table", "Trk.ID Function: ZGTT\_OTE\_SO\_HD", and "Tr. ID Code Set: SALES\_ORDER". The "SALES\_ORDER" field is highlighted with a red box.

**SAP Business Network Global Track and Trace Version 2 Screenshot:** This screenshot shows the "Model Details" view for a model named "SOF". It lists various tracked processes like SalesOrder, SalesOrderItem, Delivery, and DeliveryItem. In the "User Model Fields" section, there is a table with columns for Name, Description, Type, Required, and Value. A modal dialog titled "Edit Tracked Process" is open for the "SalesOrder" field. The "Name" field contains "SalesOrder", and the "Tracking Id Type" field contains "SALES\_ORDER". Both of these fields are also highlighted with red boxes.

### 3: Make the customization logic in the function modules and assign them to the extractor function.

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

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

Function	Function Module	Description
ZGTT_SOF_DEHDR	ZGTT_SOF_OTE_DE_HDR_REL	Extractor for relevance determination for Delivery Order Header
ZGTT_SOF_DEITM	ZGTT_SOF_OTE_DE_ITM_REL	Extractor for relevance determination for Delivery Order Items
ZGTT_SOF_SHPHDR	ZGTT_SOF_OTE_SHP_HDR_REL	Extractor for relevance determination for Shipment
ZGTT_SOF_SOHDR	ZGTT_SOF_OTE_SO_HDR_REL	Extractor for relevance determination for Sales Order Header
ZGTT_SOF_SOITM	ZGTT_SOF_OTE_SO_ITM_REL	Extractor for relevance determination for Sales Order Items

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

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

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

# 4: Sample Codes for Sales Order Fulfillment Application

To support the Sales Order Fulfillment Application, the sample codes in Github covers the following cases by function group ZGTT\_SOF:

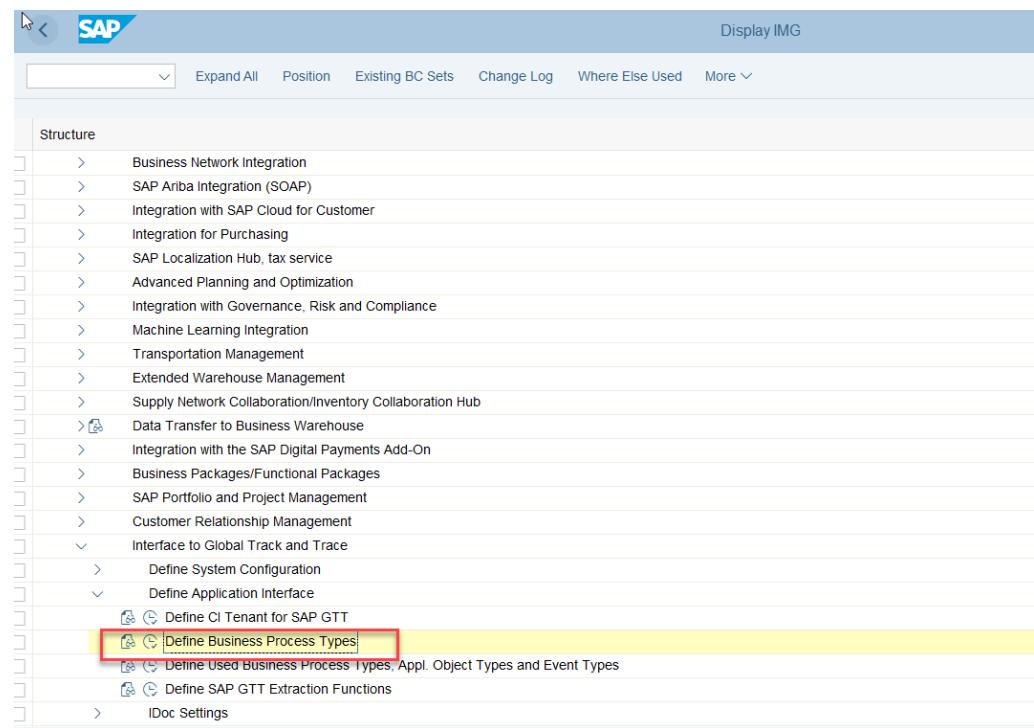
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_HD	Function for setup of control parameters of delivery header
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_ITM	Function for setup of control parameters of delivery item
Control Parameter Extractors	ESC_SHIPMT	ZGTT_SOF_OTE_SHP_HD	Function for setup of control parameters of shipment
Control Parameter Extractors	ESC_SORDER	ZGTT_SOF_OTE_SO_HD	Function for setup of control parameters of sales order header
Control Parameter Extractors	ESC_SORDER	ZGTT_SOF_OTE_SO_ITM	Function for setup of control parameters of sales order item
Event Data Extractors	ESC_DELIV	ZGTT_SOF_EE_DE_GI	SOF Extractor: Actual Event of Goods Issue
Event Data Extractors	ESC_DELIV	ZGTT_SOF_EE_DE_PACKING	SOF Extractor: Actual Event of Packing
Event Data Extractors	ESC_DELIV	ZGTT_SOF_EE_DE_PICKING	SOF Extractor: Actual Event of Picking
Event Data Extractors	ESC_DELIV	ZGTT_SOF_EE_DE POD	SOF Extractor: Actual Event of POD
Event Data Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_ARRIVAL	SOF Extractor: Actual Event of Arrival
Event Data Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_CHECKIN	SOF Extractor: Actual Event of Check In
Event Data Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_DEPARTURE	SOF Extractor: Actual Event of Departure
Event Data Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_LOAD_END	SOF Extractor: Actual Event of Loading End
Event Data Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_LOAD_START	SOF Extractor: Actual Event of Loading Start
GTT relevance function of AOT	ESC_DELIV	ZGTT_SOF_OTE_DE_HDR_REL	Extractor for relevance determination for Delivery Order Header
GTT relevance function of AOT	ESC_DELIV	ZGTT_SOF_OTE_DE_ITM_REL	Extractor for relevance determination for Delivery Order Items
GTT relevance function of AOT	ESC_SHIPMT	ZGTT_SOF_OTE_SHP_HDR_REL	Extractor for relevance determination for Shipment
GTT relevance function of AOT	ESC_SORDER	ZGTT_SOF_OTE_SO_HDR_REL	Extractor for relevance determination for Sales Order Header
GTT relevance function of AOT	ESC_SORDER	ZGTT_SOF_OTE_SO_ITM_REL	Extractor for relevance determination for Sales Order Items
GTT relevance function of Event Type	ESC_DELIV	ZGTT_SOF_EE_DE_GI_REL	Extractor for relevance determination for Goods Issue Event
GTT relevance function of Event Type	ESC_DELIV	ZGTT_SOF_EE_DE_PACKING_REL	Extractor for relevance determination for Packing Event
GTT relevance function of Event Type	ESC_DELIV	ZGTT_SOF_EE_DE_PICKING_REL	Extractor for relevance determination for Picking Event
GTT relevance function of Event Type	ESC_DELIV	ZGTT_SOF_EE_DE POD_REL	Extractor for relevance determination for POD Event
GTT relevance function of Event Type	ESC_SHIPMT	ZGTT_SOF_EE_SHP_ARRIVAL_REL	Extractor for relevance determination for Arrival Event
GTT relevance function of Event Type	ESC_SHIPMT	ZGTT_SOF_EE_SHP_CHECKIN_REL	Extractor for relevance determination for CheckIn Event
GTT relevance function of Event Type	ESC_SHIPMT	ZGTT_SOF_EE_SHP_DEPARTURE_REL	Extractor for relevance determination for Departure Event
GTT relevance function of Event Type	ESC_SHIPMT	ZGTT_SOF_EE_SHP_LOAD_END_REL	Extractor for relevance determination for Loading End Event
GTT relevance function of Event Type	ESC_SHIPMT	ZGTT_SOF_EE_SHP_LOAD_START_REL	Extractor for relevance determination for Loading Start
Planned Event Extractors	ESC_DELIV	ZGTT_SOF_EE_DE HD	SOF Extractor: Planned Event for Delivery Header of Outbound Delivery
Planned Event Extractors	ESC_DELIV	ZGTT_SOF_EE_DE_ITM	SOF Extractor: Planned Event for Delivery Item of Outbound Delivery
Planned Event Extractors	ESC_SHIPMT	ZGTT_SOF_EE_SHP_HD	SOF Extractor: Planned Event for Shipment
Tracking ID Extractors	ESC_DELIV	ZGTT_ADD_TRACKID_OTE_DEITEM	Function for setup of tracking IDs of delivery item
Tracking ID Extractors	ESC_SHIPMT	ZGTT_ADD_TRACKID_OTE_SHPHDR	Function for setup of tracking IDs of shipment
Tracking ID Extractors	ESC_SORDER	ZGTT_ADD_TRACKID_OTE_SOITEM	Function for setup of tracking IDs of sales order item

# 5: Available Contexts for the extractors' modules

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

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

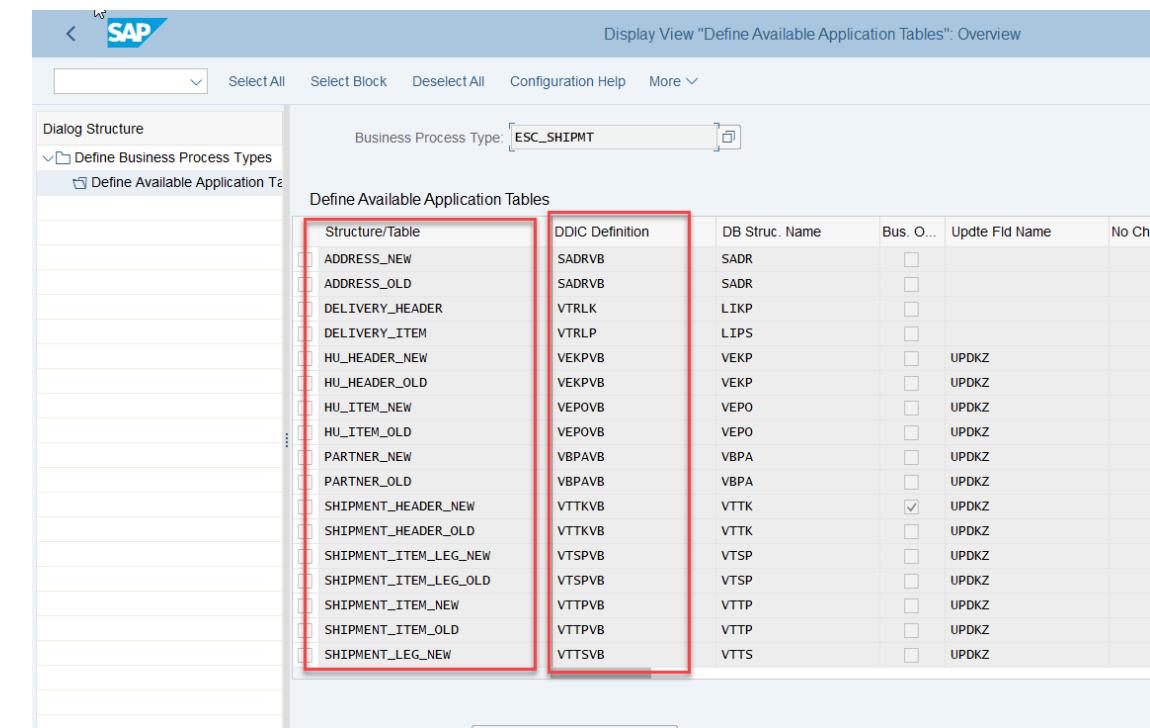
5-3: Please select the Business Process Types to find all the context tables and their structure info.



Display IMG

Structure

- > Business Network Integration
- > SAP Ariba Integration (SOAP)
- > Integration with SAP Cloud for Customer
- > Integration for Purchasing
- > SAP Localization Hub, tax service
- > Advanced Planning and Optimization
- > Integration with Governance, Risk and Compliance
- > Machine Learning Integration
- > Transportation Management
- > Extended Warehouse Management
- > Supply Network Collaboration/Inventory Collaboration Hub
- > Data Transfer to Business Warehouse
- > Integration with the SAP Digital Payments Add-On
- > Business Packages/Functional Packages
- > SAP Portfolio and Project Management
- > Customer Relationship Management
- < Interface to Global Track and Trace
  - > Define System Configuration
  - < Define Application Interface
    - Define CI Tenant for SAP GTT
    - Define Business Process Types** (highlighted with a red box)
    - Define Used Business Process Types, Appl. Object Types and Event Types
    - Define SAP GTT Extraction Functions
  - > IDoc Settings



Display View "Define Available Application Tables": Overview

Dialog Structure

Business Process Type: **ESC\_SHIPMT**

Structure/Table	DDIC Definition	DB Struc. Name	Bus. O...	Updt Fld Name	No Ch...
ADDRESS_NEW	SADRVB	SADR			
ADDRESS_OLD	SADRVB	SADR			
DELIVERY_HEADER	VTRLK	LTKP			
DELIVERY_ITEM	VTRLP	LIPS			
HU_HEADER_NEW	VEKPB	VEKP			UPDKZ
HU_HEADER_OLD	VEKPB	VEKP			UPDKZ
HU_ITEM_NEW	VEPOVB	VEPO			UPDKZ
HU_ITEM_OLD	VEPOVB	VEPO			UPDKZ
PARTNER_NEW	VBPAVB	VBPA			UPDKZ
PARTNER_OLD	VBPAVB	VBPA			UPDKZ
SHIPMENT_HEADER_NEW	VTTKVB	VTTK			UPDKZ
SHIPMENT_HEADER_OLD	VTTKVB	VTTK			UPDKZ
SHIPMENT_ITEM_LEG_NEW	VTSPVB	VTSP			UPDKZ
SHIPMENT_ITEM_LEG_OLD	VTSPVB	VTSP			UPDKZ
SHIPMENT_ITEM_NEW	VTPPB	VTPP			UPDKZ
SHIPMENT_ITEM_OLD	VTPPB	VTPP			UPDKZ
SHIPMENT_LEG_NEW	VTTSVB	VTTS			UPDKZ

# 6: Coding Tips in the GTT relevance function modules

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

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

See sample code of function: *ZGTT\_SOF\_OTE\_SHP\_HDR\_REL*

The screenshot shows the SAP Function Builder interface with the title bar "Function Builder: Display ZGTT\_SOF\_OTE\_SHP\_HDR\_REL". The main area displays the ABAP source code for the function module. The code is annotated with red boxes highlighting specific sections of logic:

- A red box surrounds the first section of code, lines 28 to 40, which checks if the main table is a Shipment and performs a create operation if it is.
- A red box surrounds the second section of code, lines 41 to 45, which reads the main object table (Shipment - VTTK) and assigns it to a local variable.
- A red box surrounds the third section of code, lines 46 to 50, which checks the relevance of AOT in OTE.
- A red box surrounds the final section of code, lines 51 to 55, which handles the update of the VTTK table and performs delivery assignment checks.

```
28     <ls_xvttk>      TYPE vttkb.
29
30     *<1> Check if Main table is Shipment or not.
31     IF i_app_object-maintabdef >* go_bpt_shipment_header_new.
32       PERFORM create_logtable_ao_rel
33         TABLES c_logtable
34           USING i_app_object-maintabdef
35             space
36             i_app_obj_types-trelfunc
37             i_app_object-appobjtype
38             i_appls.
39             RAISE parameter_error.
40     ELSE.
41       Read Main Object Table (Shipment - VTTK)
42       ASSIGN i_app_object-maintabdef->* TO <ls_xvttk>.
43     ENDIF.
44
45     *<2> Check Relevance of AOT IN OTE
46     IF i_app_object-maintabdef = go_bpt_relevance_shp.
47       PERFORM check_act_relevance_shp
48         USING <ls_xvttk>
49         CHANGING lv_act_relevance.
50       CHECK lv_act_relevance IS NOT INITIAL.
51
52     When shipment is newly created, check relevance of GTT: only when delivery has been assigned.
53     IF <ls_xvttk>-updtk = 0.
54       PERFORM check_delivery_assignment
55         USING i_all_appl_tables
56       c...
```

## 7: Coding Tips in the Tracking ID function modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E\_TRACKIDDATA*.
3. The Tracking ID Type need to be the same as the definition in the process type of model in Manage Models app.
4. 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. The tracking ID for its own process type needs to be filled for each process update.
6. In case of tracking ID deletion, the field *ACTION* shall be filled with 'D'.

See sample code of function:

*ZGTT\_ADD\_TRACKID\_OTE\_SHPHDR*

The screenshot shows the SAP Function Builder interface with the title bar "Function Builder: Display ZGTT\_ADD\_TRACKID\_OTE\_SHPHDR". Below the title bar, there are tabs: Attributes, Import, Export, Changing, Tables, Exceptions, and Source Code. The "Source Code" tab is selected. The code area displays the ABAP source code for the function module. A red box highlights the logic for handling different values of *ls\_xvttk->vsart*. The code uses concatenation and APPEND statements to build the *e\_trackiddata* structure based on the value of *ls\_xvttk->vsart*. Another red box highlights the logic for handling updates where *ls\_xvttk->vsart* is not initial, reading from table *lt\_yvttk* to update *ls\_yvttk*.

```
76 e_trackiddata-trxcod = 'SHIPMENT_ORDER'.
77 e_trackiddata-trxid = <ls_xvttk>-tknum.
78 CONCATENATE '01' sy-datum sy-uzeit INTO e_trackiddata-start_date.
79 e_trackiddata-end_date = '09991231000000'.
80 e_trackiddata-timzon = lv_tzone.
81 e_trackiddata-msrid = space.
82 APPEND e_trackiddata.
83
84 e_trackiddata-trxcod = 'RESOURCE'.
85 CLEAR e_trackiddata-start_date.
86 CLEAR e_trackiddata-end_date.
87 CLEAR e_trackiddata-timzon.
88
89 IF <ls_xvttk>-updkz = 'I'.
90   IF <ls_xvttk>-vsart = '01' AND <ls_xvttk>-extil IS NOT INITIAL.
91     CONCATENATE <ls_xvttk>-tknum <ls_xvttk>-extil INTO e_trackiddata-trxid.
92     e_trackiddata-msrid = space.
93     APPEND e_trackiddata.
94   ELSEIF <ls_xvttk>-vsart = '04' AND <ls_xvttk>-signi IS NOT INITIAL.
95     CONCATENATE <ls_xvttk>-tknum <ls_xvttk>-signi INTO e_trackiddata-trxid.
96     e_trackiddata-msrid = space.
97     APPEND e_trackiddata.
98   ENDIF.
99
100 ELSEIF <ls_xvttk>-updkz = 'U'.
101   READ TABLE lt_yvttk INTO ls_yvttk INDEX 1.
102   CHECK ls_yvttk IS NOT INITIAL.
103
104 IF <ls_xvttk>-vsart <> ls_yvttk-vsart.
105   IF <ls_xvttk>-vsart = '01' AND <ls_xvttk>-extil IS NOT INITIAL.
```

## 8: Coding Tips in the Control Parameter function modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E\_CONTROL\_DATA*.
3. SAP Business Network Global Track and Trace v2 asks for full transport for all the control parameters, which means that all the fields needs to be extracted in all cases, no matter whether their values have been changed.
4. To fill in the composition (table) fields defined in Manage Model app, use the parameter field *PARAMINDEX* to specify the line number. 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 app. It's not recommended to fill a code list type field to clear a composition even if it's a key field.**
6. The field with fixed name 'ACTUAL\_BUSINESS\_DATETIME' and 'ACTUAL\_BUSINESS\_TIMEZONE' are mandatory fields to be transported for event handling sequencing in SAP Business Network Global Track and Trace Version 2.
7. The fields with fixed names 'ACTUAL\_TECHNICAL\_TIMEZONE' and 'ACTUAL\_TECHNICAL\_DATETIME' are optional and recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in SAP Business Network Global Track and Trace via TP request, which leads to an error)
8. In Manage Models app, click tab *IDOC Integration* to map the parameter names and model field names.
9. For DATE or DATETIME fields, when the source value is initial like '00000000' '0000000000000000', then please ensure to only enable *PARAMNAME* and *PARAMINDEX* in the extractor code, not enable *VALUE* for IDOC sending.
10. For amount field which has reference currency, ensure to call BAPI 'BAPI\_CURRENCY\_CONV\_TO\_EXTERNAL' using the reference currency to make the amount tracked correctly by SAP Business Network Global Track and Trace Version 2. The BAPI will output the conversion result in 4 decimals as fixed, which needs additional rounding in the extractor if the corresponding field defined in the tracking model is less than 4 decimals.
11. In the shipment extractor, add the prefix LBN# into the fields SERVICE AGENT LBN ID for integration with Visibility Providers.

See sample code of function: *ZGTT\_SOF\_OTE\_SHP\_HD*

# 8: Coding Tips in the Control Parameter Function Modules

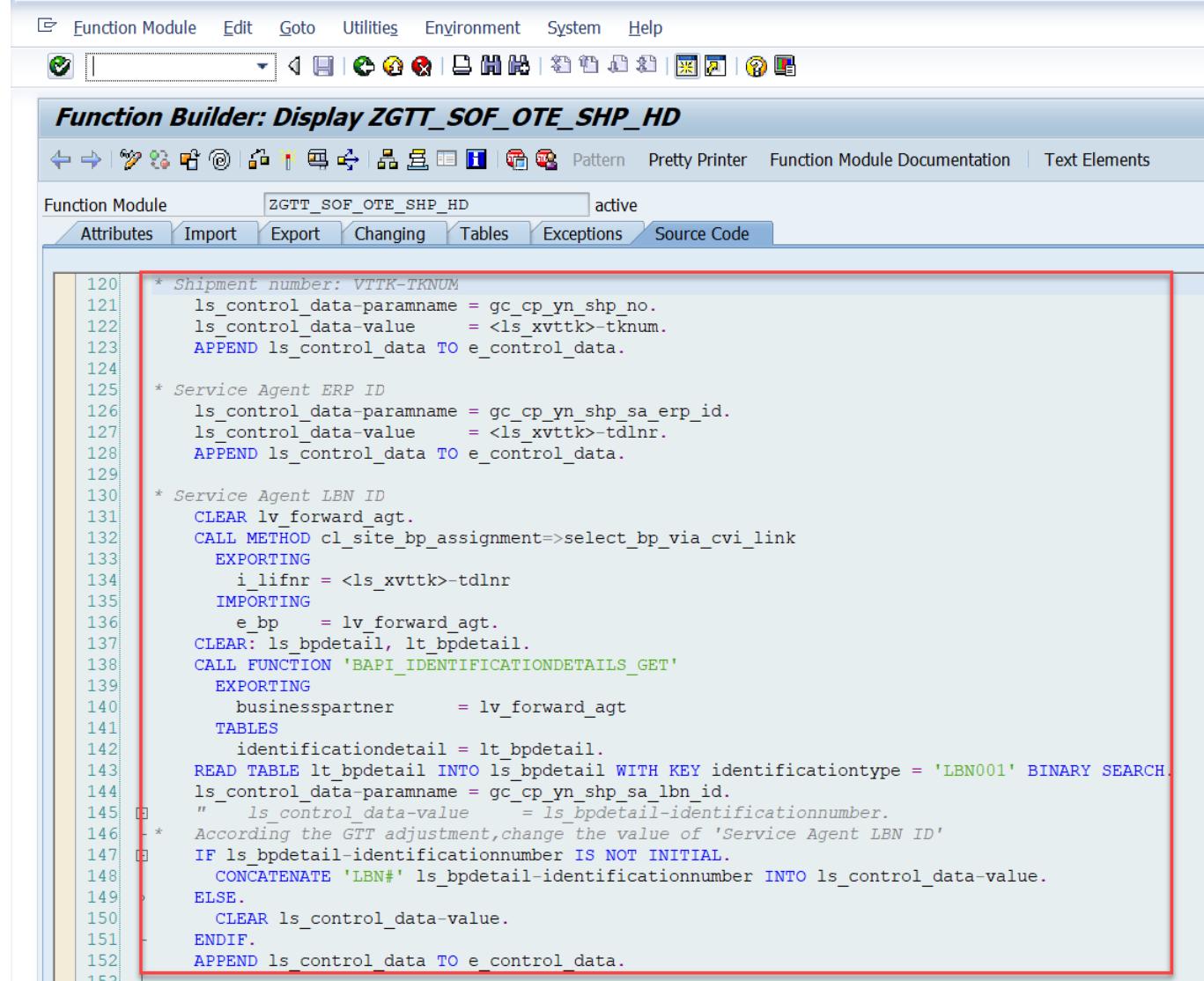
Fields mapping is set up in the Manage Models app in the IDOC Integration section:

The screenshot shows the SAP Manage Models app interface. At the top, there's a header with the SAP logo, a 'Model Details' dropdown set to 'Internal - Test', and user icons for help and profile. Below the header, the model name 'sof' is shown with a status of 'Active'. A sub-header indicates the model is for 'Sales Order Fulfillment'. The main navigation bar includes tabs for 'Tracked Process', 'Field Type Pool', 'Event Type Pool', 'Code List', 'IDOC Integration' (which is currently selected), 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. On the left, a sidebar displays 'Tracked Process / Events (26)' with a table showing rows for 'Tracked Process' (ShipmentEvent, E1EHPAO) and various 'Event Types' (LoadingStart, POD, Departure, Arrival, LoadingEnd). On the right, under 'IDOC Integration', there's a 'Tracked Process Mapping' section with dropdowns for 'ERP Object Type: Others' and 'Application Object Type: ZGTT\_SHP\_INT\_HD'. An 'Integration Switch' is turned 'ON'. A large table titled 'User Model Fields' is highlighted with a red border, listing fields like shipmentNo, serviceAgentLbnd, transportationMode, etc., along with their IDOC Segment and Field names.

User Model Fields		
Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCP	YN_SHP_NO
serviceAgentLbnd	E1EHPCP	YN_SHP_SA_LBN_ID
transportationMode	E1EHPCP	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPCP	YN_SHP_CONTAIN_DGOODS
forwardingAgentTrackingId	E1EHPCP	YN_SHP_FA_TRACKING_ID
> stops		
shippingType	E1EHPCP	YN_SHP_SHIPPING_TYPE
> resourceTPs		

# 8: Coding Tips in the Control Parameter Function Modules

Main logic of shipment is implemented in function module ZGTT\_SOF\_OTE\_SHP\_HD



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT\_SOF\_OTE\_SHP\_HD". The function module "ZGTT\_SOF\_OTE\_SHP\_HD" is active. The "Source Code" tab is selected. The code is highlighted with a red box around the main logic starting from line 120.

```
120 * Shipment number: VTTR-TKNUM
121   ls_control_data-paramname = gc_cp_yn_shp_no.
122   ls_control_data-value     = <ls_xvttk>-tknum.
123   APPEND ls_control_data TO e_control_data.
124
125 * Service Agent ERP ID
126   ls_control_data-paramname = gc_cp_yn_shp_sa_erp_id.
127   ls_control_data-value     = <ls_xvttk>-tdlnr.
128   APPEND ls_control_data TO e_control_data.
129
130 * Service Agent LBN ID
131   CLEAR lv_forward_agt.
132   CALL METHOD cl_site_bp_assignment=>select_bp_via_cvi_link
133     EXPORTING
134       i_lifnr = <ls_xvttk>-tdlnr
135     IMPORTING
136       e_bp    = lv_forward_agt.
137   CLEAR: ls_bpdetail, lt_bpdetail.
138   CALL FUNCTION 'BAPI_IDENTIFICATIONDETAILS_GET'
139     EXPORTING
140       businesspartner      = lv_forward_agt
141     TABLES
142       identificationdetail = lt_bpdetail.
143     READ TABLE lt_bpdetail INTO ls_bpdetail WITH KEY identificationtype = 'LBN001' BINARY SEARCH.
144     ls_control_data-paramname = gc_cp_yn_shp_sa_lbn_id.
145     "   ls_control_data-value     = ls_bpdetail-identificationnumber.
146   * According the GTT adjustment, change the value of 'Service Agent LBN ID'
147   IF ls_bpdetail-identificationnumber IS NOT INITIAL.
148     CONCATENATE 'LBN#' ls_bpdetail-identificationnumber INTO ls_control_data-value.
149   ELSE.
150     CLEAR ls_control_data-value.
151   ENDIF.
152   APPEND ls_control_data TO e_control_data.
```

# 9: Coding Tips in the Planned Event function modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E\_EXPEVENTDATA*.
3. As default except no change made on the model configuration, SAP Business Network Global Track and Trace version 2 asks for full transport for all the planned events, which means that all the events needs to be extracted in all cases, no matter whether their values have been changed. If nothing is transported, the planned events will be removed in SAP Business Network Global Track and Trace Version 2.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT\_EXP\_DATETIME* is optional, but need to be filled with relevant time zone *EVT\_EXP\_TZONE* together if it needs to be transported.
6. The field *LOC\_ID1* is optional, but need to be filled with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* app in SAP Business Network Global Track and Trace Version 2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

Name	IDOC	Event Code
<strong>Tracked Process</strong>		
ShipmentEvent	E1EHPAO	
<strong>Event Types</strong>		
LoadingStart	E1EVMHDR02	LOAD_BEGIN
POD	E1EVMHDR02	POD
Departure	E1EVMHDR02	DEPARTURE
Arrival	E1EVMHDR02	ARRIV_DEST
LoadingEnd	E1EVMHDR02	LOAD_END
Checkin	E1EVMHDR02	CHECK_IN

# 9: Coding Tips in the Planned Event Function Modules

Main logic of shipment Planned Events is implemented in function module ZGTT\_SOF\_EE\_SHP\_HD

The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT\_SOF\_EE\_SHP\_HD". The function module "ZGTT\_SOF\_EE\_SHP\_HD" is active. The "Source Code" tab is selected. The code is as follows:

```
125 * Planned Load-Start
126   ls_expeventdata-milestone    = 'LOAD_BEGIN'.
127 *
128 * Get Planned Load-Start datetime
129   PERFORM set_local_timestamp
130     USING      <ls_xvttk>-dplbg
131       <ls_xvttk>-uplbg
132     CHANGING ls_expeventdata-evt_exp_datetime.
133     APPEND ls_expeventdata TO e_expeventdata.
134
135 * Planned Load-End
136   ls_expeventdata-milestone    = 'LOAD_END'.
137 *
138 * Get Planned Load-End datetime
139   PERFORM set_local_timestamp
140     USING      <ls_xvttk>-dplen
141       <ls_xvttk>-uplen
142     CHANGING ls_expeventdata-evt_exp_datetime.
143     APPEND ls_expeventdata TO e_expeventdata.
144
145 CLEAR lt_stops.
146 CALL FUNCTION 'ZGTT_GET_STOPS_FROM_SHIPMENT'
147   EXPORTING
148     iv_tknum      = <ls_xvttk>-tknum
149     it_vtts_new  = lt_xvtt
150   IMPORTING
151     et_stops     = lt_stops.
152
153 LOOP AT lt_stops INTO ls_stop.
154   IF ls_stop-locat = 'S'.
155     ls_expeventdata-milestone    = 'DEPARTURE'.
156   ELSE.
157     ls_expeventdata-milestone    = 'ARRIV_DEST'.
158   ENDIF.
```

# 10: Coding Tips in the Event Data function modules

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

1. Make sure that the Main / Master tables 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. Add two technical parameters with fixed names ‘ACTUAL\_TECHNICAL\_TIMEZONE’ and ‘ACTUAL\_TECHNICAL\_DATETIME’ which are recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in SAP Business Network Global Track and Trace via TP request, which leads to an error)
5. If the event has reference table information, fill in the table *CT\_TRACKREFERENCES*.
6. The field *CT\_TRACKINGHEADER-SRCCOD*, *SRCID*, *SRCTX* is used for event reason transport.
7. In *Manage Models* app, click tab *IDOC Integration* to map the user-defined parameter names and model field names.

See sample code of function: *ZGTT\_SOF\_EE\_DE\_PICKING*

# 10: Coding Tips in the Event Data Function Modules

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

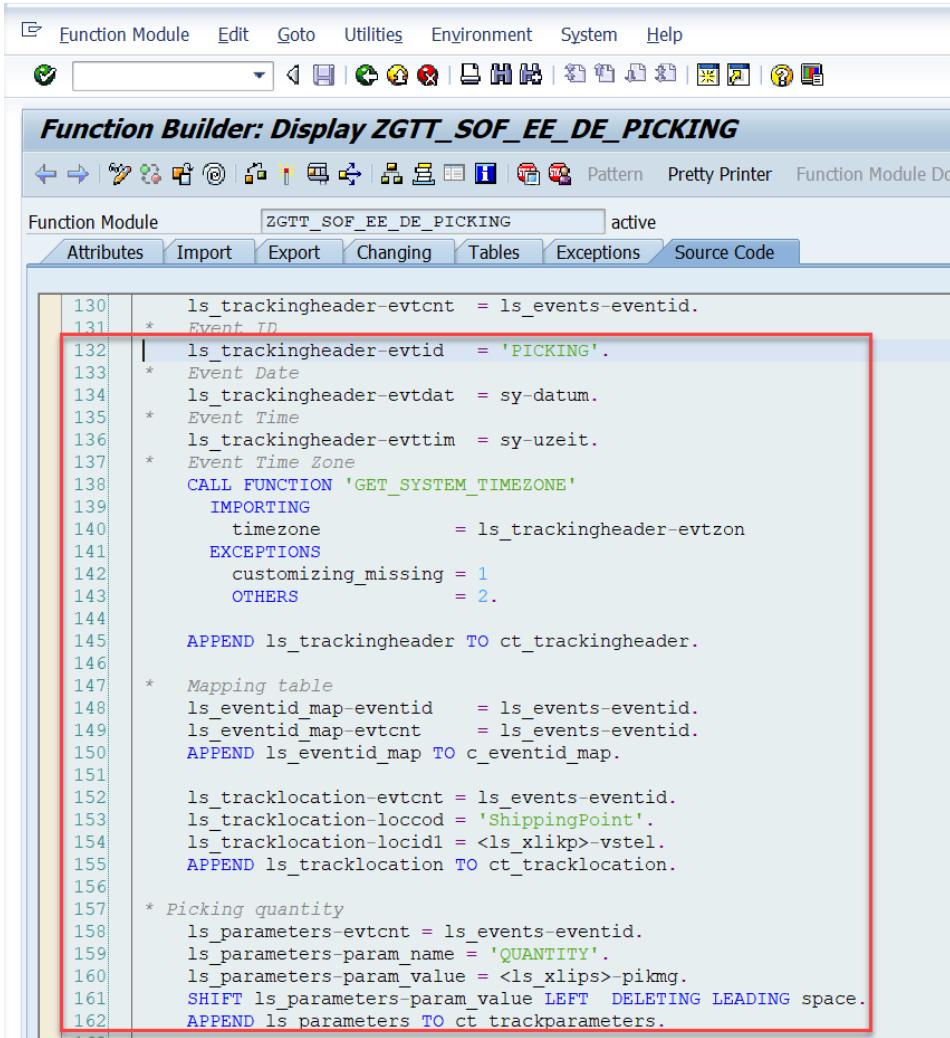
The screenshot shows the SAP Model Details interface for the 'Internal - Test' model. The 'sof' entry is marked as 'Active'. The 'IDOC Integration' tab is selected. The 'Tracked Process' dropdown is set to 'DeliveryItem'. The 'Integration Switch' is turned 'ON'. The 'Tracked Process Mapping' section shows 'ERP Object Type: Others' and 'Application Object Type: ZGTT\_DE\_INT\_ITEM'. The 'Tracked Process / Events (4)' table lists four entries: 'Tracked Process' (row 1), 'DeliveryItemEvent' (row 2), 'Event Types' (row 3), and 'Picking' (row 4). The 'User Model Fields' table maps fields like 'quantity' to IDOC segments and fields.

Name	IDOC	Event Code
Tracked Process		
DeliveryItemEvent	E1EHPAO	
Picking	E1EVMPAR02	PICKING
Packing	E1EVMPAR02	PACKING
DeliveryItemPOD	E1EVMPAR02	DLV POD

Field	IDOC Segment	IDOC Field
quantity	E1EVMPAR	QUANTITY

# 10: Coding Tips in the Event Data Function Modules

Main logic of delivery item picking event is implemented in function module `ZGTT_SOF_EE_DE_PICKING`



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT\_SOF\_EE\_DE\_PICKING". The function module name "ZGTT\_SOF\_EE\_DE\_PICKING" is active. The code editor displays the following ABAP code:

```
130 ls_trackingheader-evtcnt = ls_events-eventid.
131 * Event ID
132 | ls_trackingheader-evtid = 'PICKING'.
133 * Event Date
134 ls_trackingheader-evtdat = sy-datum.
135 * Event Time
136 ls_trackingheader-evttim = sy-uzeit.
137 * Event Time Zone
138 CALL FUNCTION 'GET_SYSTEM_TIMEZONE'
139   IMPORTING
140     timezone      = ls_trackingheader-evtzon
141   EXCEPTIONS
142     customizing_missing = 1
143     OTHERS          = 2.
144
145 APPEND ls_trackingheader TO ct_trackingheader.
146
147 * Mapping table
148 ls_eventid_map-eventid = ls_events-eventid.
149 ls_eventid_map-evtcnt = ls_events-eventid.
150 APPEND ls_eventid_map TO c_eventid_map.
151
152 ls_tracklocation-evtcnt = ls_events-eventid.
153 ls_tracklocation-loccod = 'ShippingPoint'.
154 ls_tracklocation-locidl = <ls_xlikp>-vstel.
155 APPEND ls_tracklocation TO ct_tracklocation.
156
157 * Picking quantity
158 ls_parameters-evtcnt = ls_events-eventid.
159 ls_parameters-param_name = 'QUANTITY'.
160 ls_parameters-param_value = <ls_xlips>-pikmg.
161 SHIFT ls_parameters-param_value LEFT DELETING LEADING space.
162 APPEND ls_parameters TO ct_trackparameters.
```

# 11: Enhancement codes for cross-processes tracking

The Sales Order Fulfillment application asks for cross-processes tracking, which is used in below cases:

- When the shipment process is updated and transported to SAP Business Network Global Track and Trace, the preceding delivery and item process, and their planned events needs to be updated and transported to SAP Business Network Global Track and Trace.

**IMPORTANT:** To enable cross-processes tracking, please update the below sample codes after downloading:

- Replace your Delivery AOT type name in Method *BEFORE\_UPDATE* of BADI implementation *Z\_GTT\_SOF\_LE\_SHIPMNT*

The screenshot shows the SAP Business Add-In Builder interface for the "Display Implementation Z\_GTT\_SOF\_LE\_SHIPMNT". The "Properties" tab is selected. Key configuration details include:

- Implementation Name: Z\_GTT\_SOF\_LE\_SHIPMNT (marked as active)
- Implementation Short Text: GTT - Enhancement to update the impacted delivery orders
- Definition Name: BADI\_LE\_SHIPMENT
- Runtime Behavior: Implementation will be called
- Interface Name: IF\_EX\_BADI\_LE\_SHIPMENT
- Name of Implementing Class: ZCL\_IM\_GTT\_SOF\_LE\_SHIPMNT

In the Method section, three methods are listed:

Method	Implementation Type	Description
AT_SAVE	ABAP ABAP code	Process Shipments During "At Save" Context
BEFORE_UPDATE	ABAP ABAP code	Process Shipments During "Before Update" Context
IN_UPDATE	ABAP ABAP code	Process Shipments During "In Update" Context

A red box highlights the "BEFORE\_UPDATE" method. Below the table, there is a field for the Default Implementation Class.

# 11: Enhancement codes for cross-processes tracking

The cross processes tracking scenarios cover below:

## **Shipment -> Delivery and Delivery Item:**

### 1\ Tracking ID (Delta Transport)

- Case: Shipment Create / Delete with Delivery
- Case: Shipment Assign / Unassign Delivery

### 2\ Shipment Composition (Full Transport)

- Case: Shipment Create / Delete with Delivery
- Case: Shipment Assign / Unassign Delivery

### 3\ Planned Event in Delivery (Full Transport)

- Case: Shipment Create / Delete with Delivery / with stage
- Case: Shipment Assign / Unassign Delivery / with stage
- Case: Stage Assign / Unassign Delivery
- Case: Stage Insert / Delete
- Case: Stage Location Update
- Case: Stage Planned Datetime Update

### 4\ Planned Event in Delivery Item (Full Transport)

- Case: Shipment Create / Delete with Delivery / with stage
- Case: Shipment Assign / Unassign Delivery / with stage
- Case: Stage Assign / Unassign Delivery
- Case: Stage Insert / Delete
- Case: Stage Location Update
- Case: Stage Planned Datetime Update

# 12: Known Issues

## 1. Planned Event Extension not enabled

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

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

## 2. IDOC sequencing issue

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

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

- 8: Coding Tips in the Control Parameter Function Modules
- 10: Coding Tips in the Event Data Function Modules
- 13: Solution of IDOC sequencing issue

# 13: Solution of IDOC Sequencing Issue

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

## 2. Create CI tenant.

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

## 3. Create RFC destination

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

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

The image contains two side-by-side screenshots of SAP GUI interfaces.

**SAP Global Track & Trace Definitions:** This screenshot shows a table titled "SAP Global Track & Trace Definitions". It has four columns: "CI for Global Track & Trace", "CI Log. System", "SAP Track & Trace Version", and "Description". There is one entry: "ZGTTSOFIN2" in the first column, "ZLSGTTINT" in the second, "GTT2.0 Logistics Business N..." in the third, and "CI For GTT V2 Integration system Sales Order Sample APP" in the fourth.

**RFC Destination ZGTTV2\_SOF\_INT2:** This screenshot shows the configuration of an RFC destination. The "RFC Destination" field is set to "ZGTTV2\_SOF\_INT2". The "Connection Type" is "HTTP Connection to External Server". The "Description" section contains three fields: "Description 1" (RFC for Tracked Process of Sales Order Sample Application), "Description 2" (empty), and "Description 3" (empty). At the bottom, there are tabs for "Administration", "Technical Settings", "Logon & Security", and "Special Options". In the "Target System Settings" section, the "Path Prefix" field is highlighted with a red box and contains the value "/api/idoc/em/v1/TrackedProcessAndEvent".

# 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.