



SAP Logistics Business Network, Global Track and Trace Option Track Sales Orders - Deep Dive with SAP ERP Integration

Logistics Business Network
March 2021

PUBLIC

Objectives



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

- Learn what prerequisite is necessary for Global Track and Trace Option
- Learn how to maintain IDOC configurations in ERP for integration
- Learn how to maintain extractors in ERP for integration
- Learn how to download and implement sample ABAP codes from Github
- Learn how to customize own logic based on sample codes

Agenda

A Prerequisites

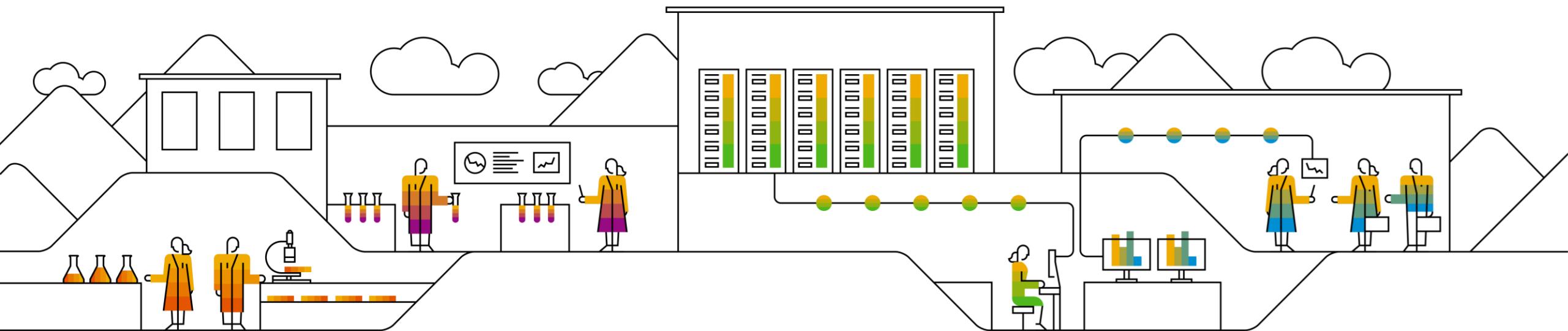
B Configuration and Implementation - Basic

 B1 IDOC Configuration

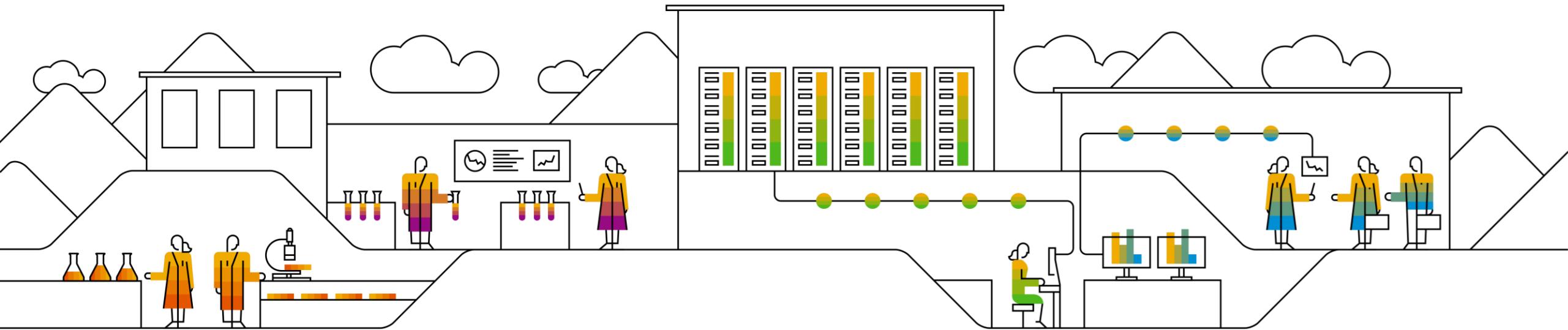
 B2 Extractor Configuration

C Download ABAP Code from GitHub

D Configuration and Coding Guide - Advanced



A) Prerequisites



STEP 1: Check the SAP Version

1-1: 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*. You can find this guide at <http://help.sap.com/gtt>.

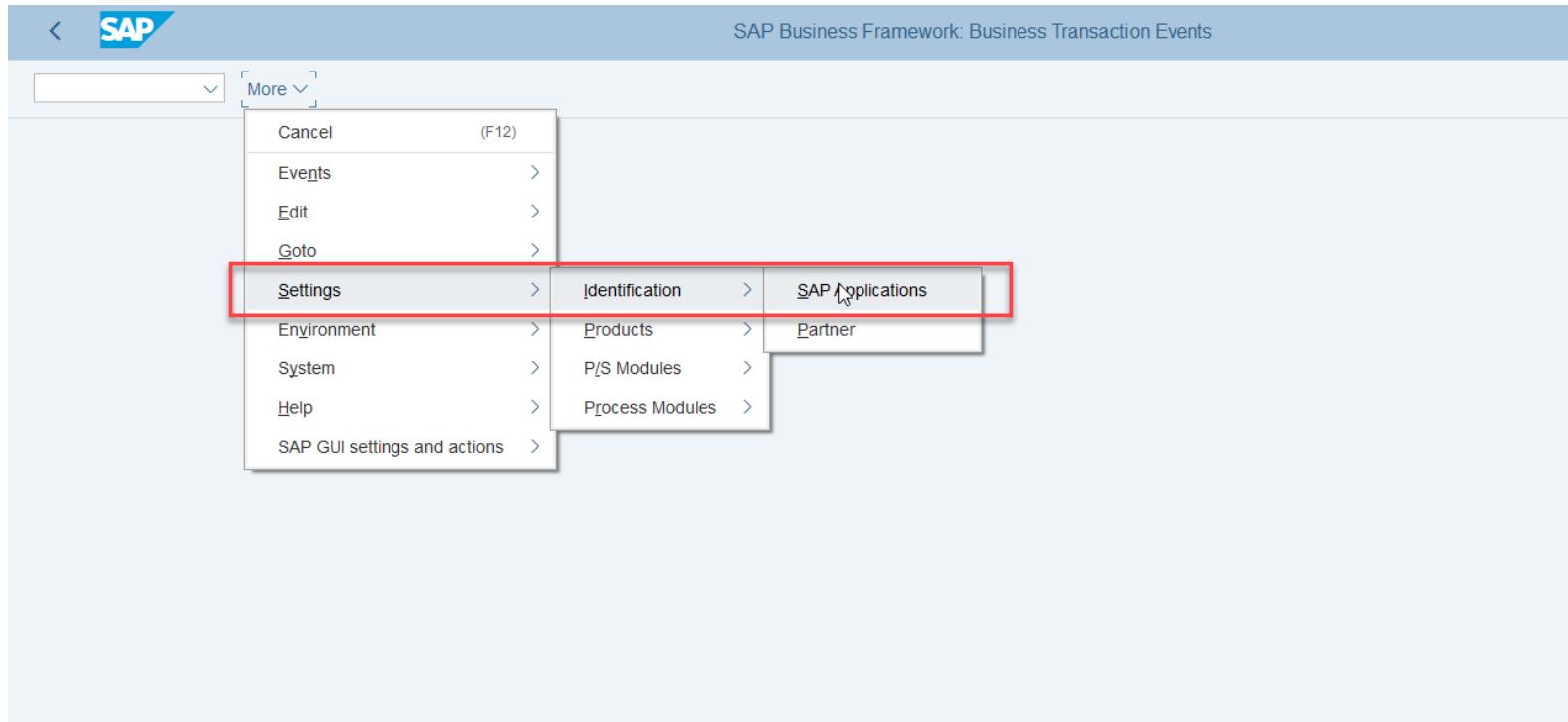
1-2 : The ABAP codes on Github to support sample apps for GTT Version 2 shall be implemented in S4 HANA 1909 SP03 on premise, which is not validated in lower release, and not applicable for ECC series of products.

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**

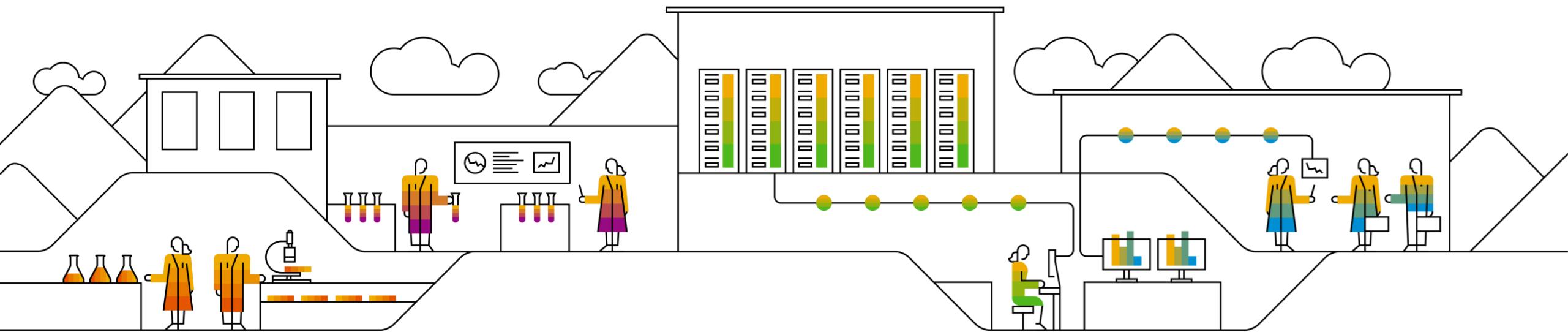
2-6: Click **Save**

The screenshot shows the SAP Fiori interface for the 'BTE Application Indicator' application. The title bar reads 'Change View "BTE Application Indicator": Overview'. The main area is a table with three columns: 'Appl.', 'A', and 'Text'. The 'Appl.' column lists application IDs, and the 'Text' column provides a brief description. The row for 'PI-EM' is highlighted with a red box around the entire row. In the 'A' column for PI-EM, there is a checkbox that is checked, indicating the application is active. Other rows in the table include PM (Instandhaltung), PM-BW (Instandhaltung-BW), PM-EQM (Instandhaltung, Equipment), PM-PAM (Instandhalt. Pool Asset Mgmt), PMA-PC (Product Compliance), PMAT (Produkt - Material), PMIPUR (PMI Anschluss Einkauf), PMPUSH (MAM Push), PP-BD (Production Planning MasterData), PP-DD (Demand Driven Replenishment), PP-MRP (Material Requirements Planning), PRICAT (Preiskatalog), PS-REP (Projektsystem), PSRVA (Produkt - Service), QBEXT (External Inspection Procurement), QBEXTP (External Inspection Production), QILPO (Inspection Lot Order Integr.), RDSVFI (Dgtl. Signature Validation FI), and RDSVM (Dgtl. Signature BP Check). At the bottom of the screen, there is a footer with the SAP logo and navigation links like 'New Entries', 'Copy As...', 'Delete', 'Undo Change', 'Select All', 'Select Block', 'Deselect All', 'More', 'Display', and 'Exit'. A 'Save' button is visible at the bottom right.

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



STEP 1: Define RFC Connection for GTT

1-1: Log on to the business client

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

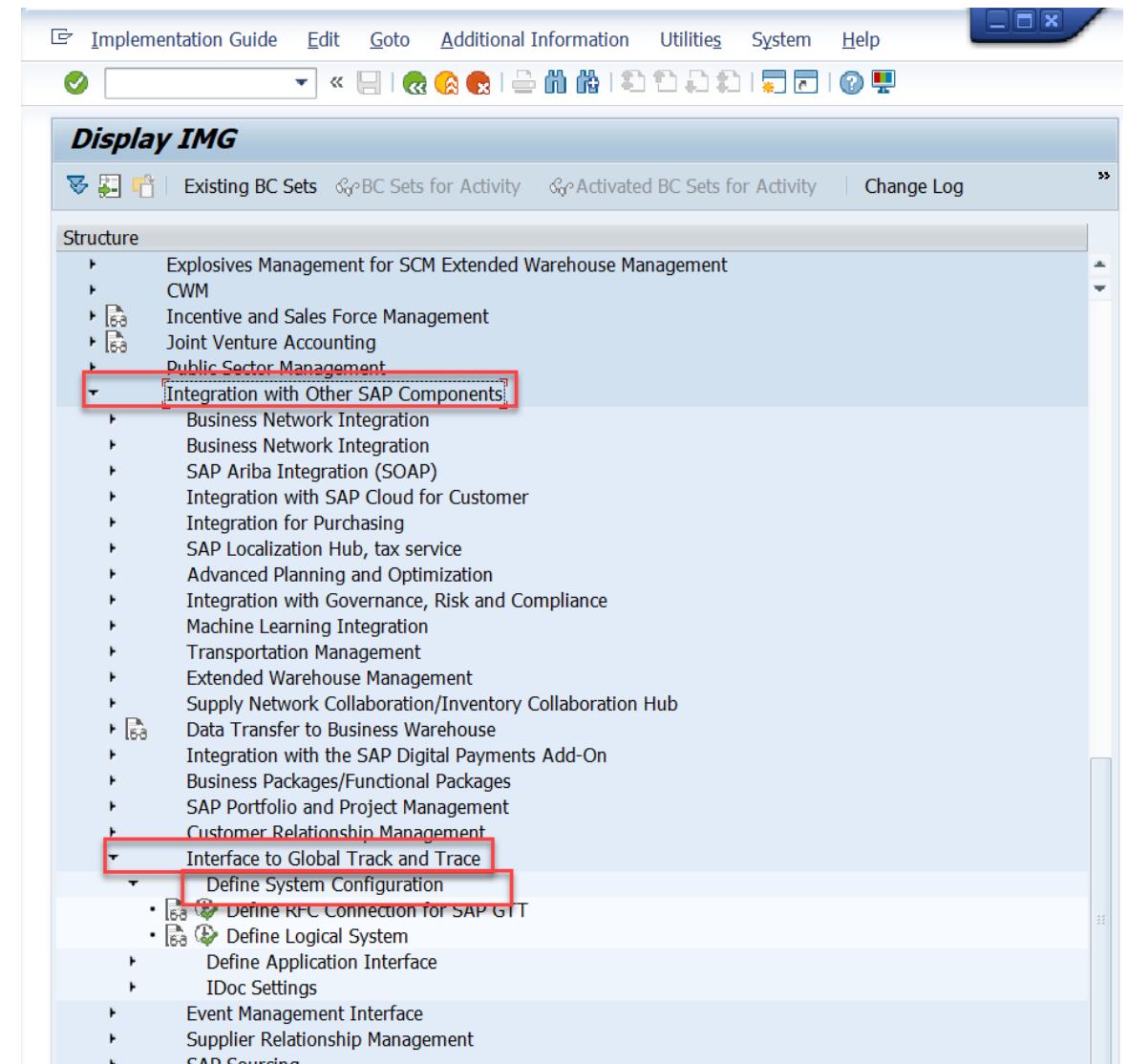
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

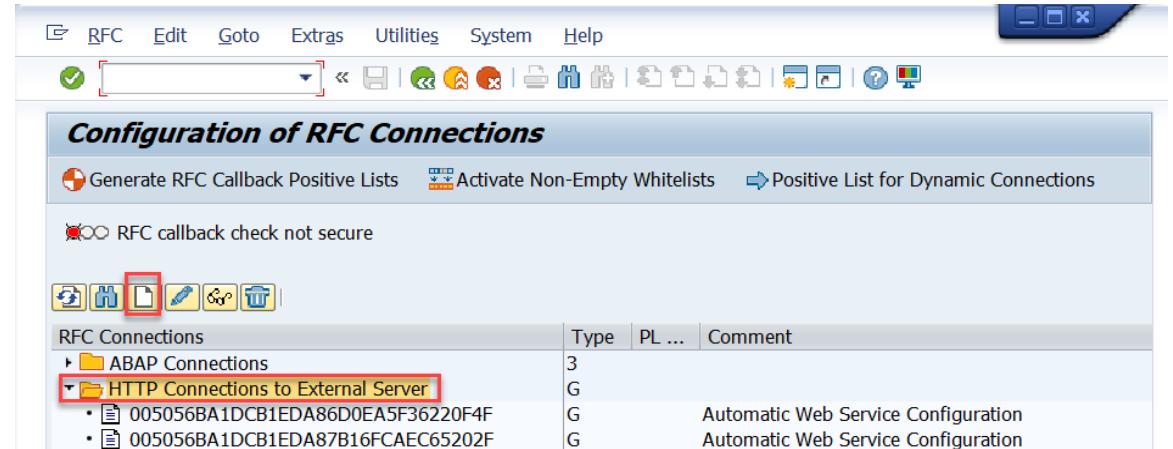
Define RFC Connection for SAP GTT



STEP 1: Define RFC Connection for GTT

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

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



STEP 1: Define RFC Connection for GTT

1-7: Enter a description

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

For example, the URL of solution owners is as follows :

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

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

Port: `443`

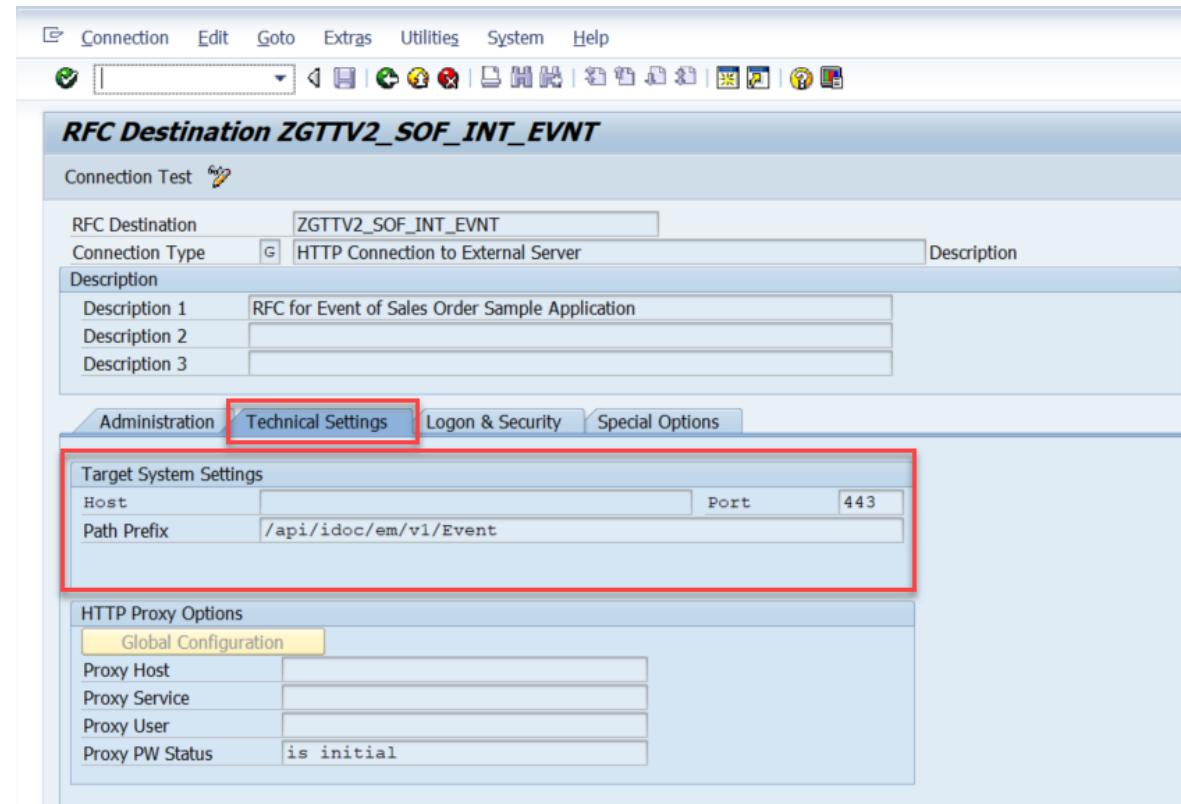
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

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 GTT

1-9: In the **Logon & Security** tab, enter the Logon information.

For basic authentication, the GTT technical user / password is needed. You can get this from your GTT administrator.

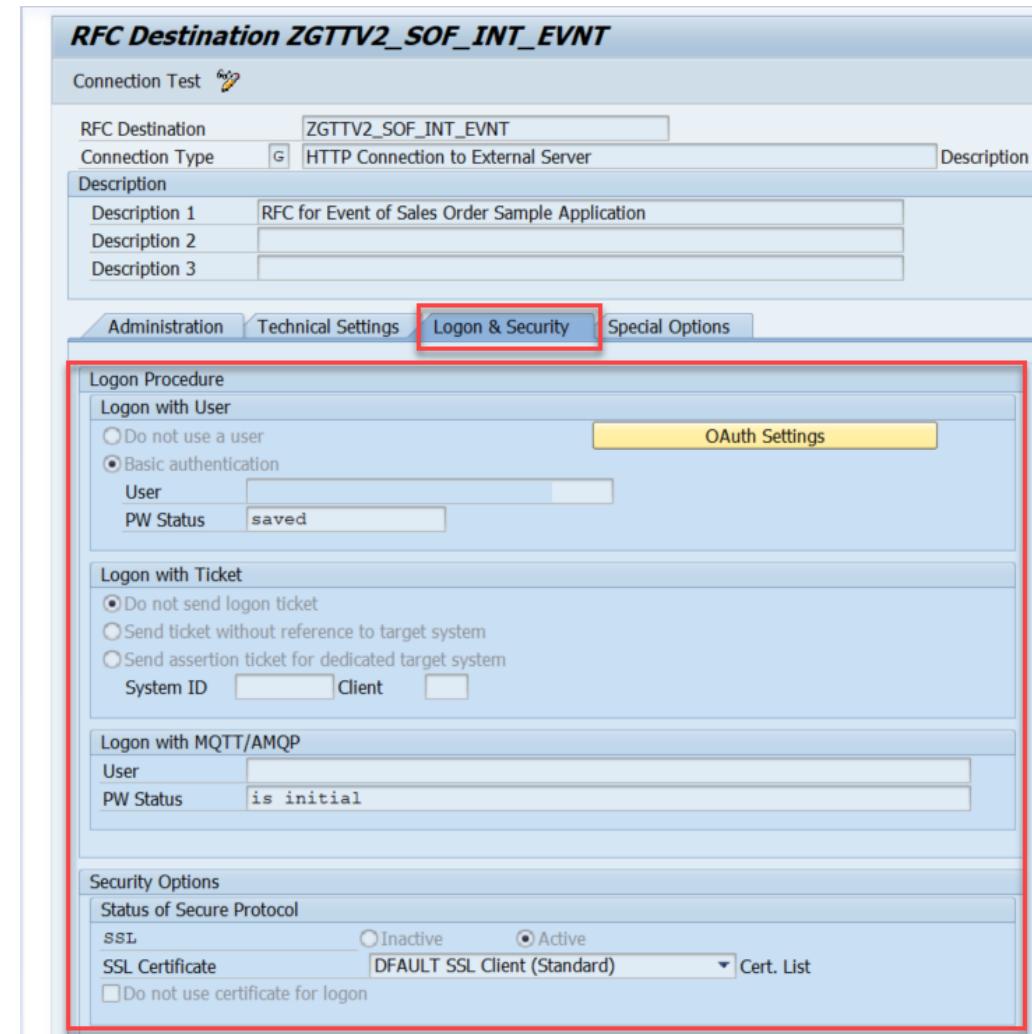
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

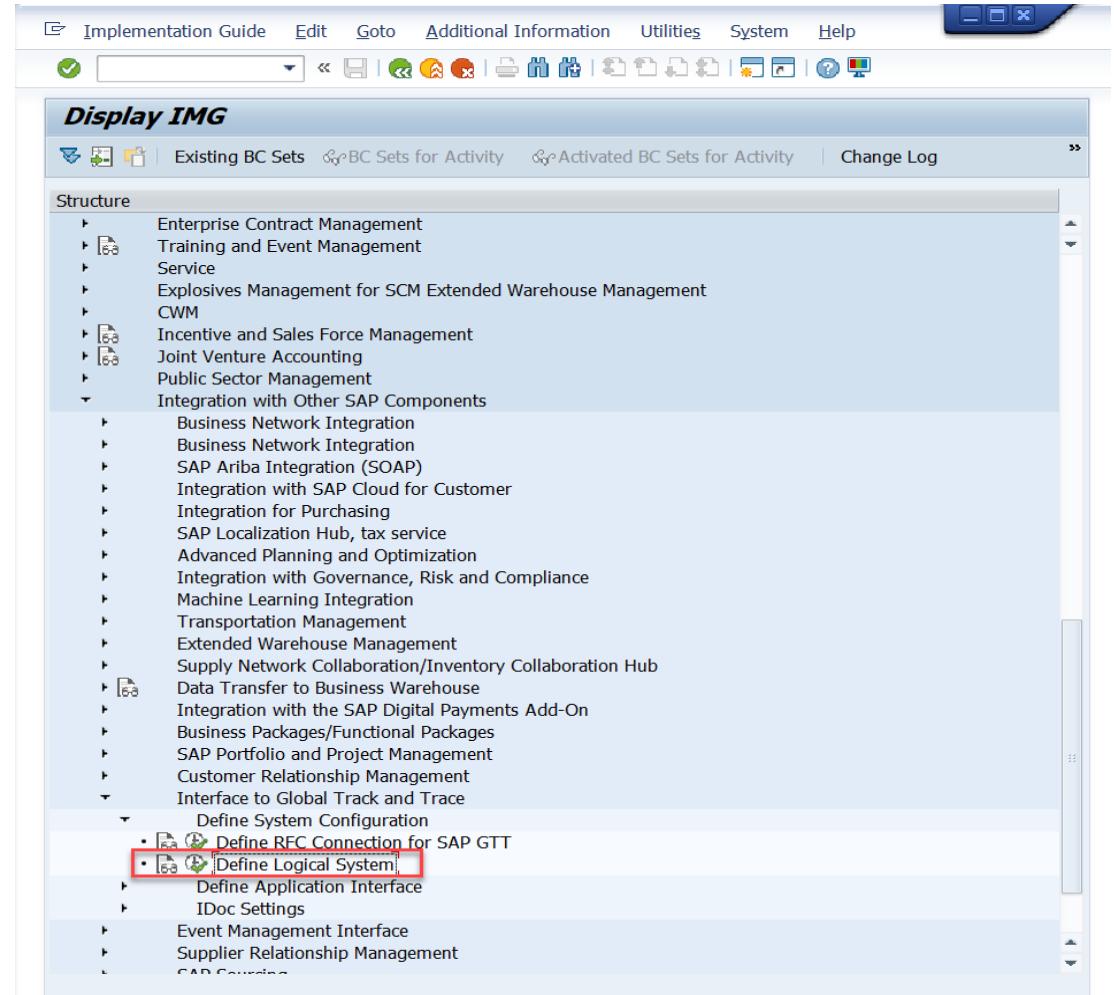
- one for event and
- the other for tracked process.



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**.

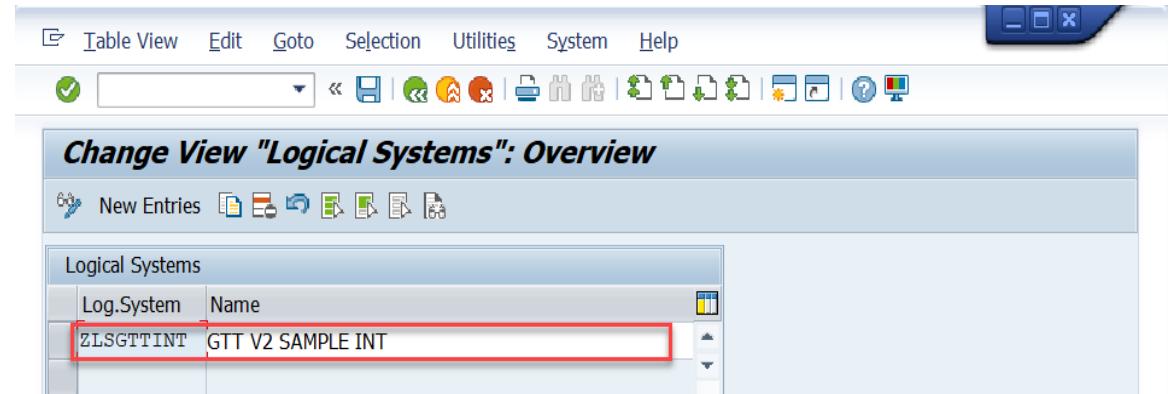


STEP 2: Define Logical System

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

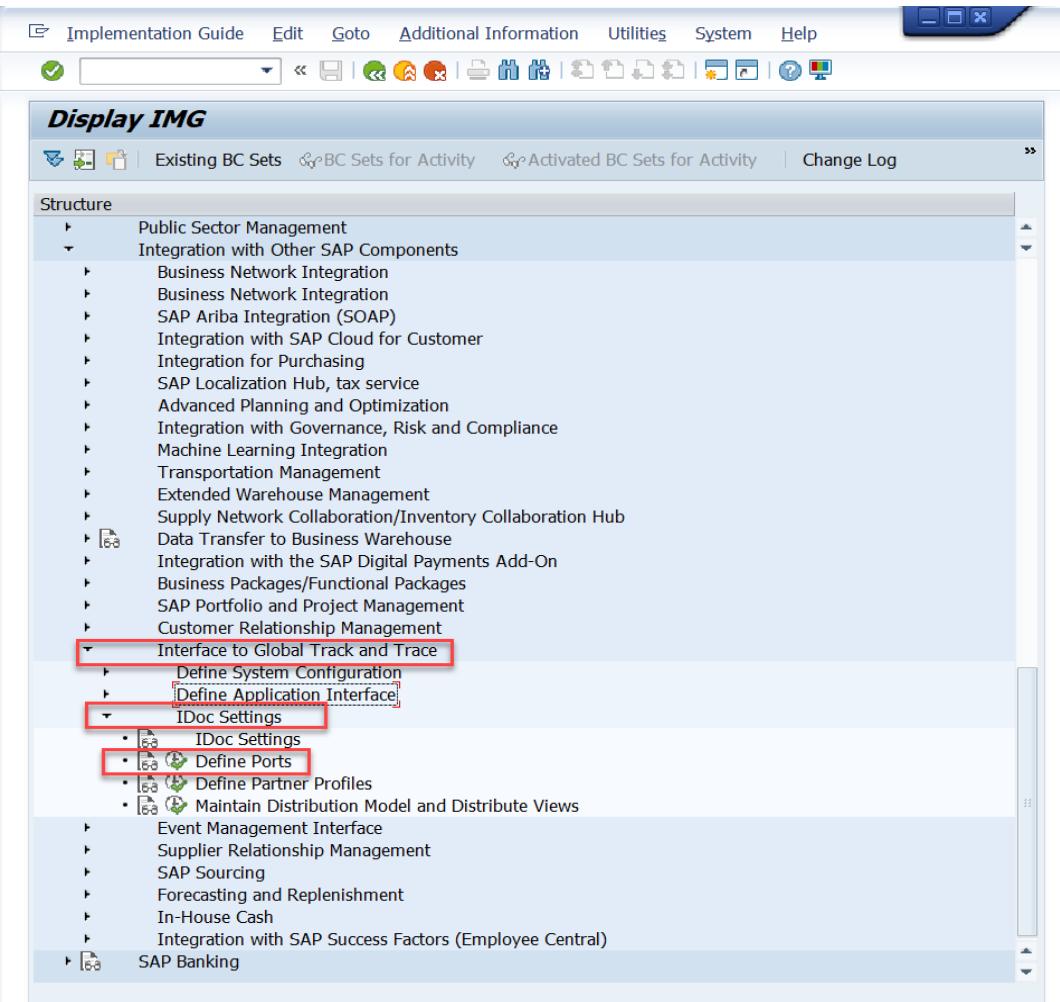


Log.System	Name
ZLSGTTINT	GTT V2 SAMPLE INT

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

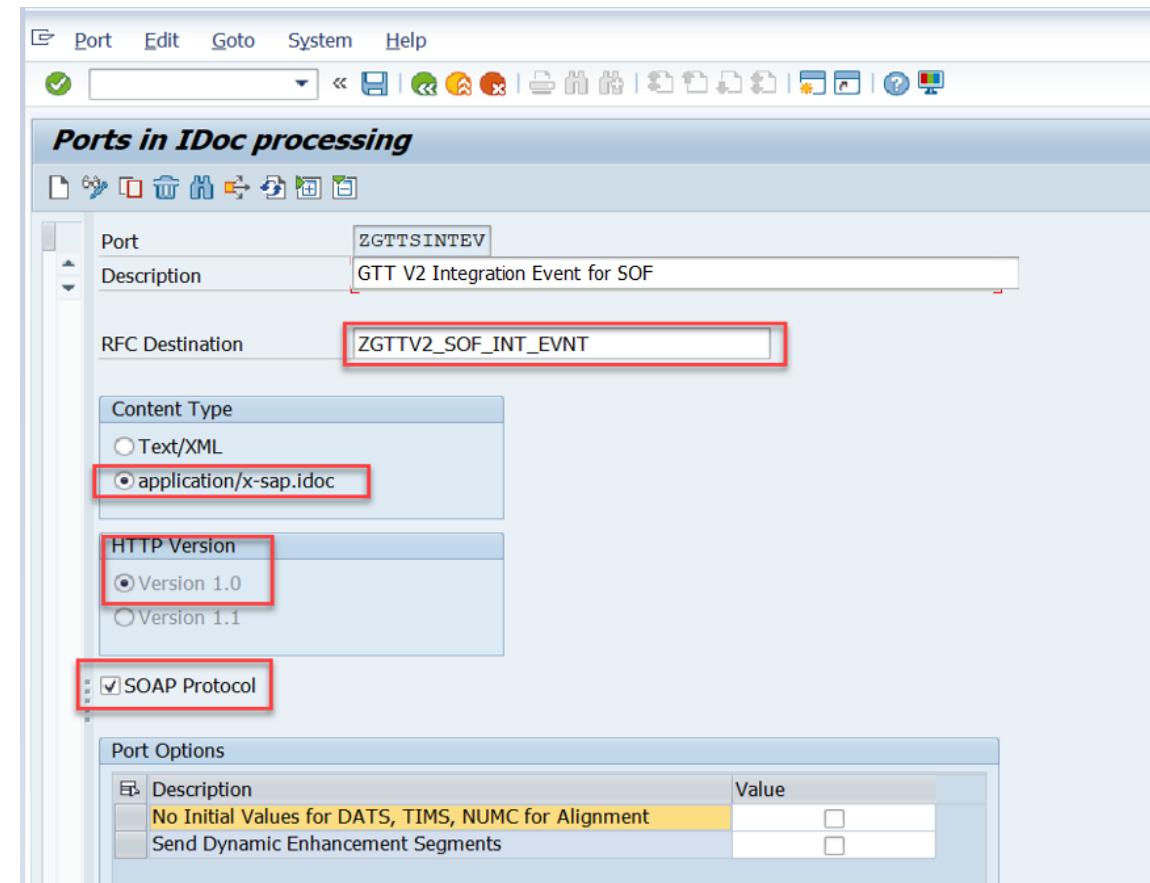
3-5: Choose **Content Type** as *application/x-sap.idoc*

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.

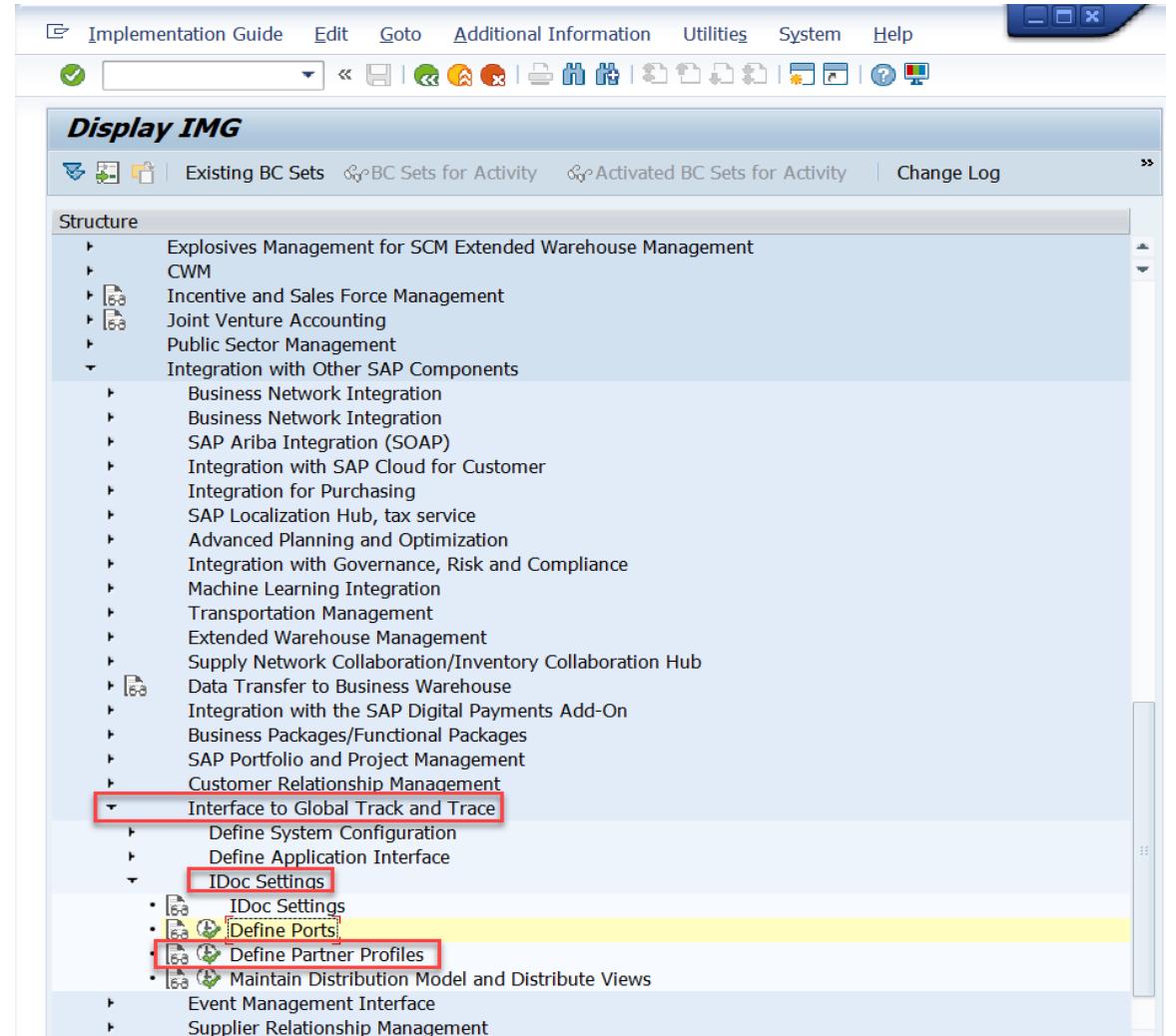


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSINTEV	GTT V2 Integration Event for SOF	ZGTTV2_SOF_INT_EVNT	application/x-sap.idoc	Version 1.0	Checked
ZGTTSINTTP	GTT V2 Integration Tracked Process for SOF	ZGTTV2_SOF_INT_TP	application/x-sap.idoc	Version 1.0	Checked

STEP 4: Define Partner Profiles

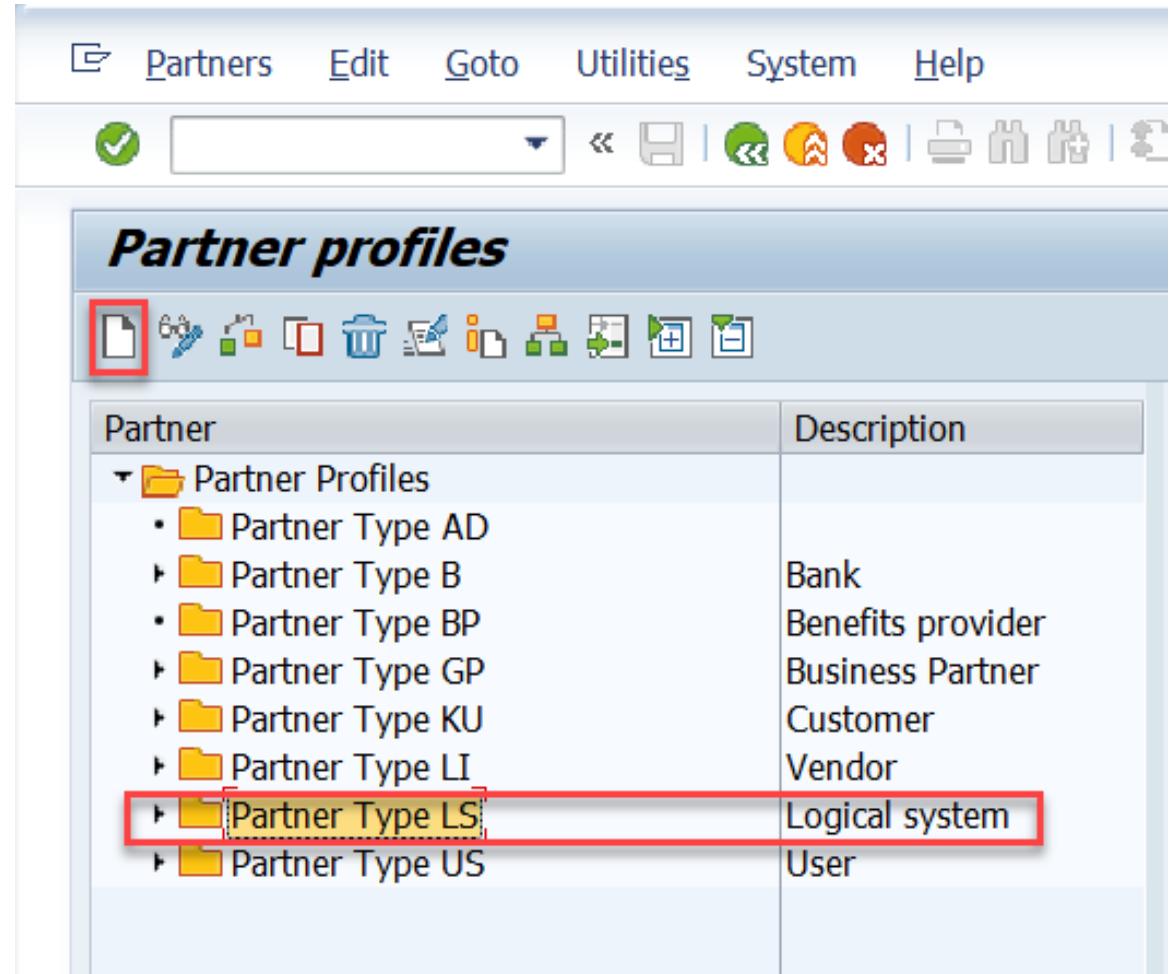
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

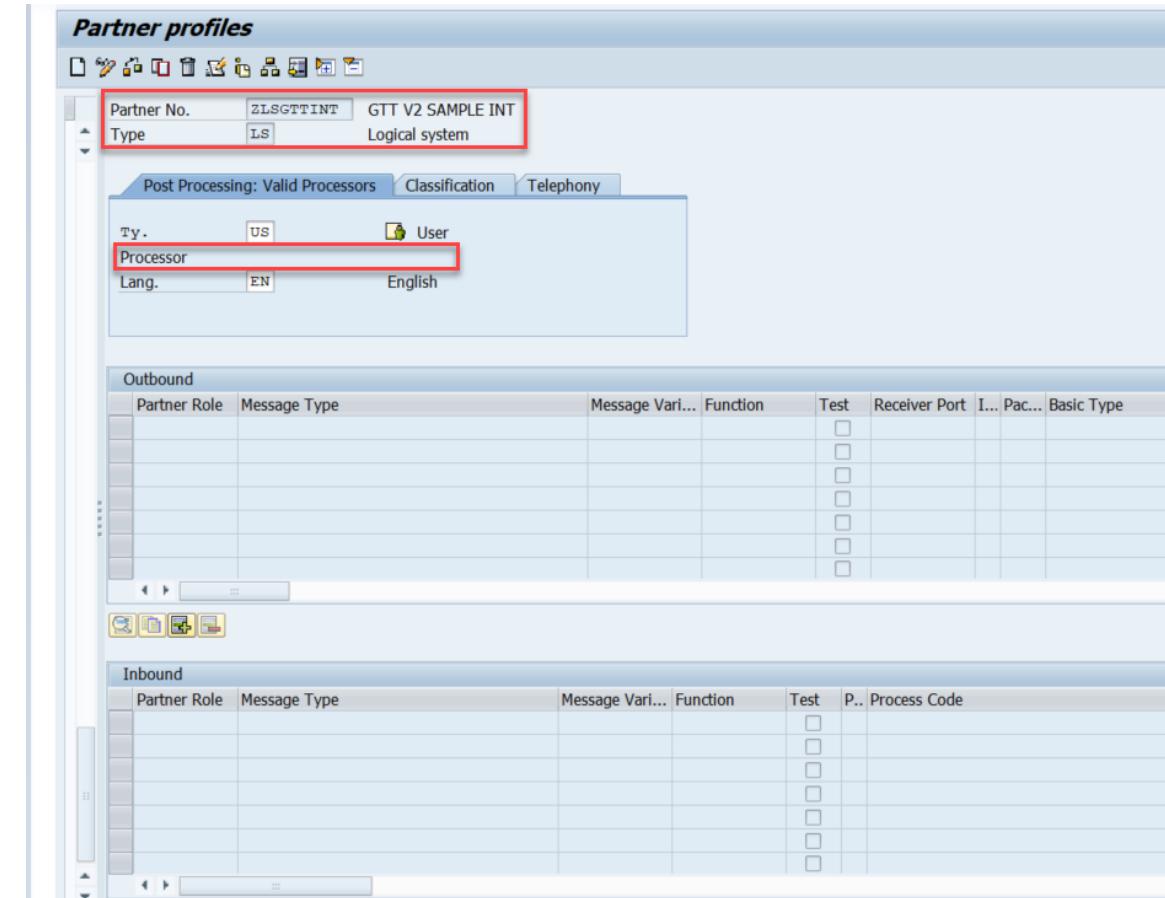
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

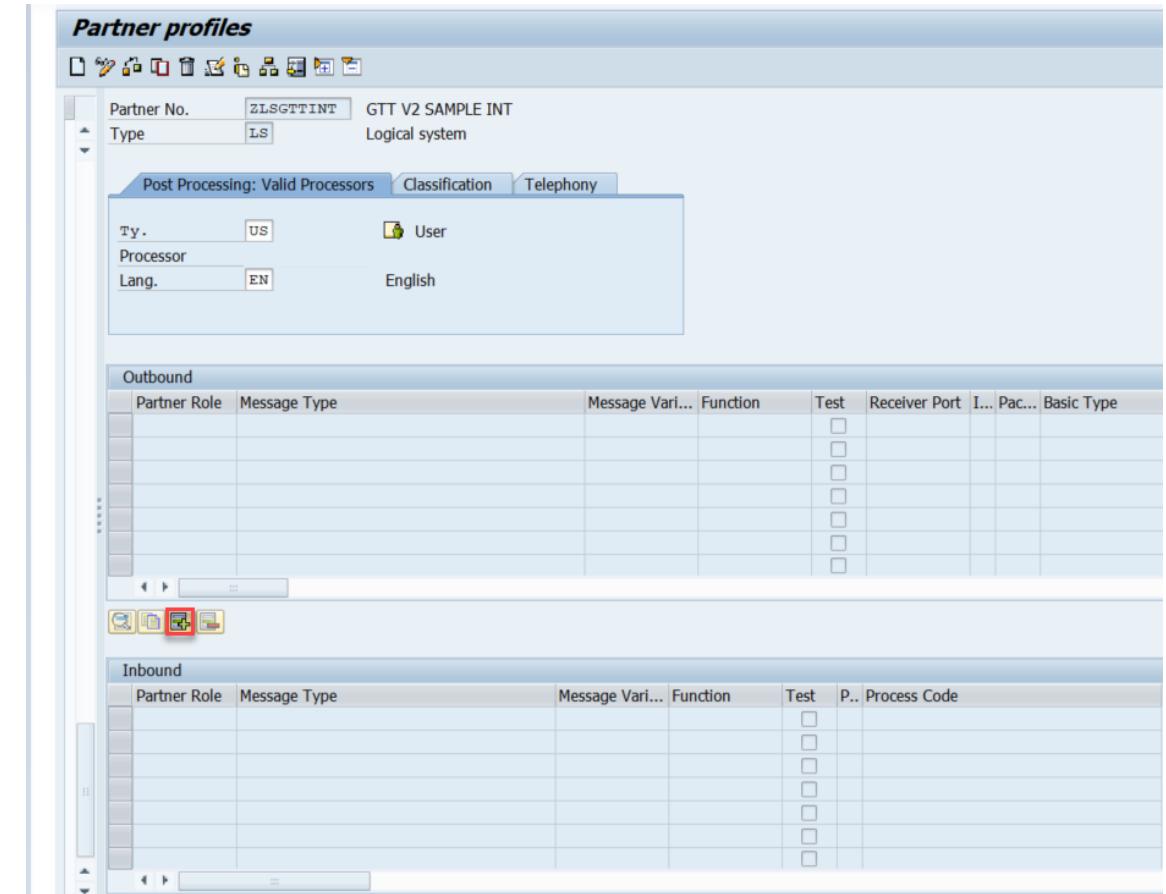
4-4: Fill in the **Partner No.** that you created in STEP 2

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



STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

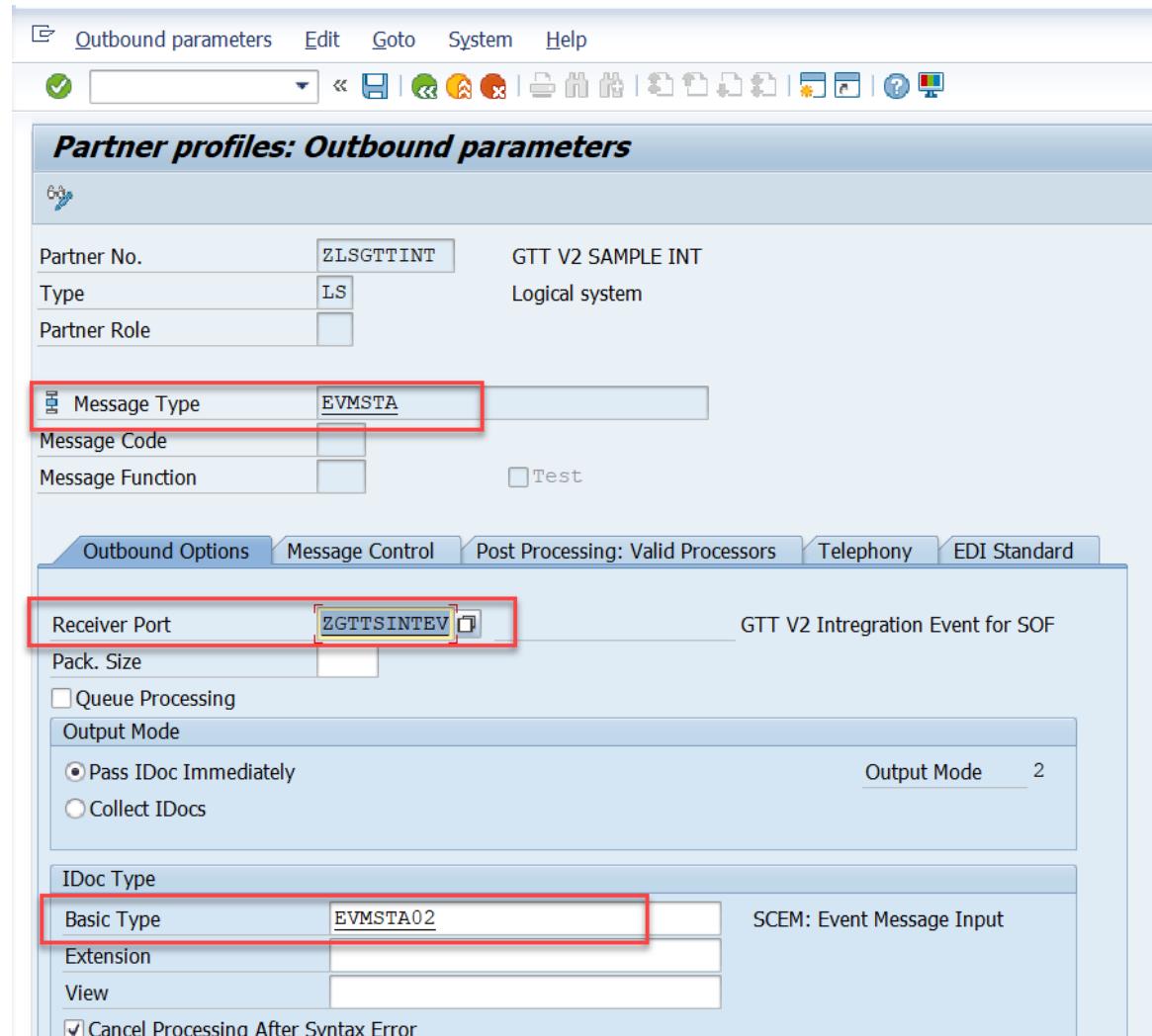
For the tracked Process:

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	ZGTTTSINTTP	EHPOST01
ZLSGTTINT	LS	Yes	EVMSTA	ZGTTTSINTEV	EVMSTA02

Partner profiles: Outbound parameters

Partner No. ZLSGTTINT GTT V2 SAMPLE INT
Type LS Logical system
Partner Role

Message Type AOPOST AOPOST

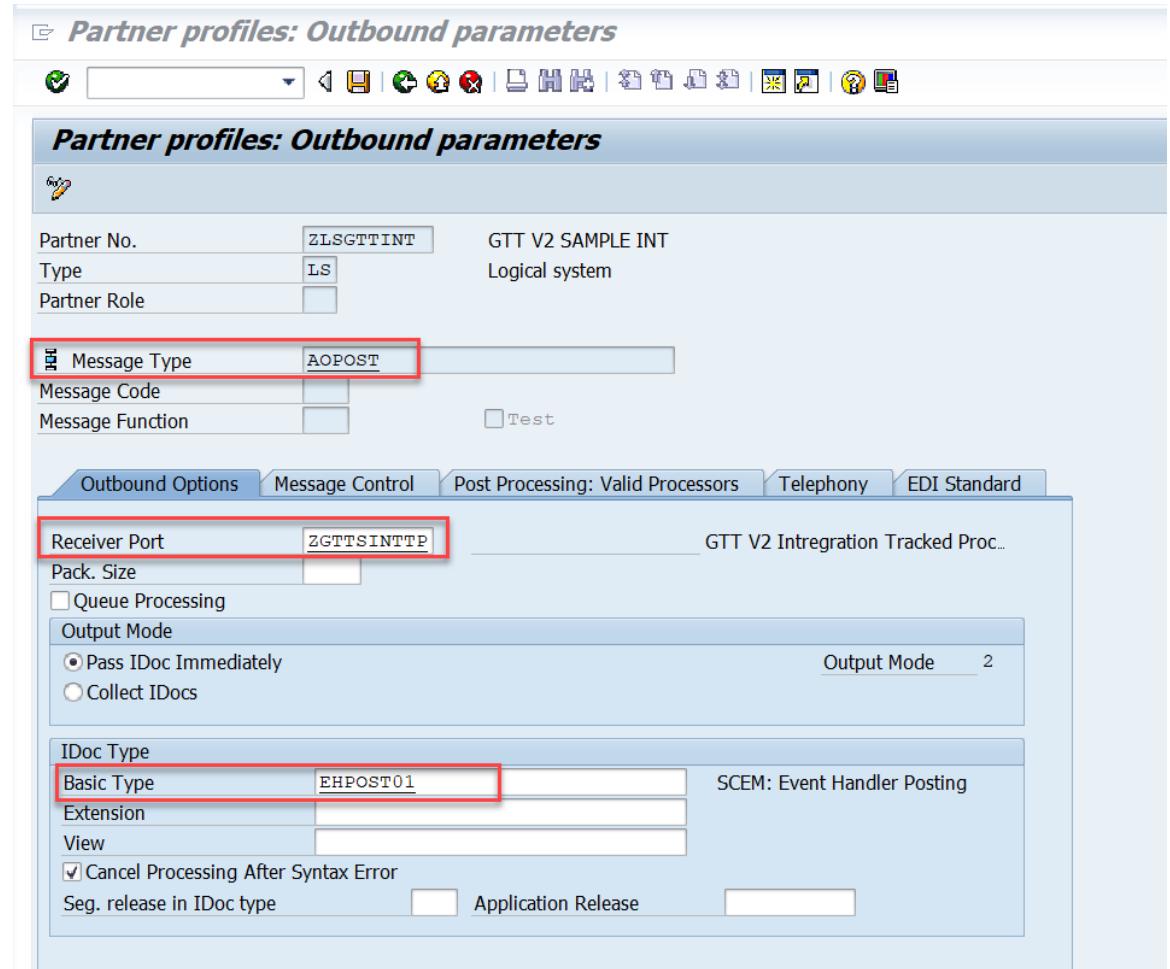
Message Code
Message Function Test

Outbound Options **Message Control** Post Processing: Valid Processors Telephony EDI Standard

Receiver Port ZGTTTSINTTP ZGTTTSINTTP
Pack. Size
 Queue Processing
Output Mode
 Pass IDoc Immediately Output Mode 2
 Collect IDocs

IDoc Type
Basic Type EHPOST01 EHPOST01
Extension
View
 Cancel Processing After Syntax Error
Seg. release in IDoc type Application Release

GTT V2 Intregation Tracked Proc... SCEM: Event Handler Posting



B) Configuration and Implementation

- Basic

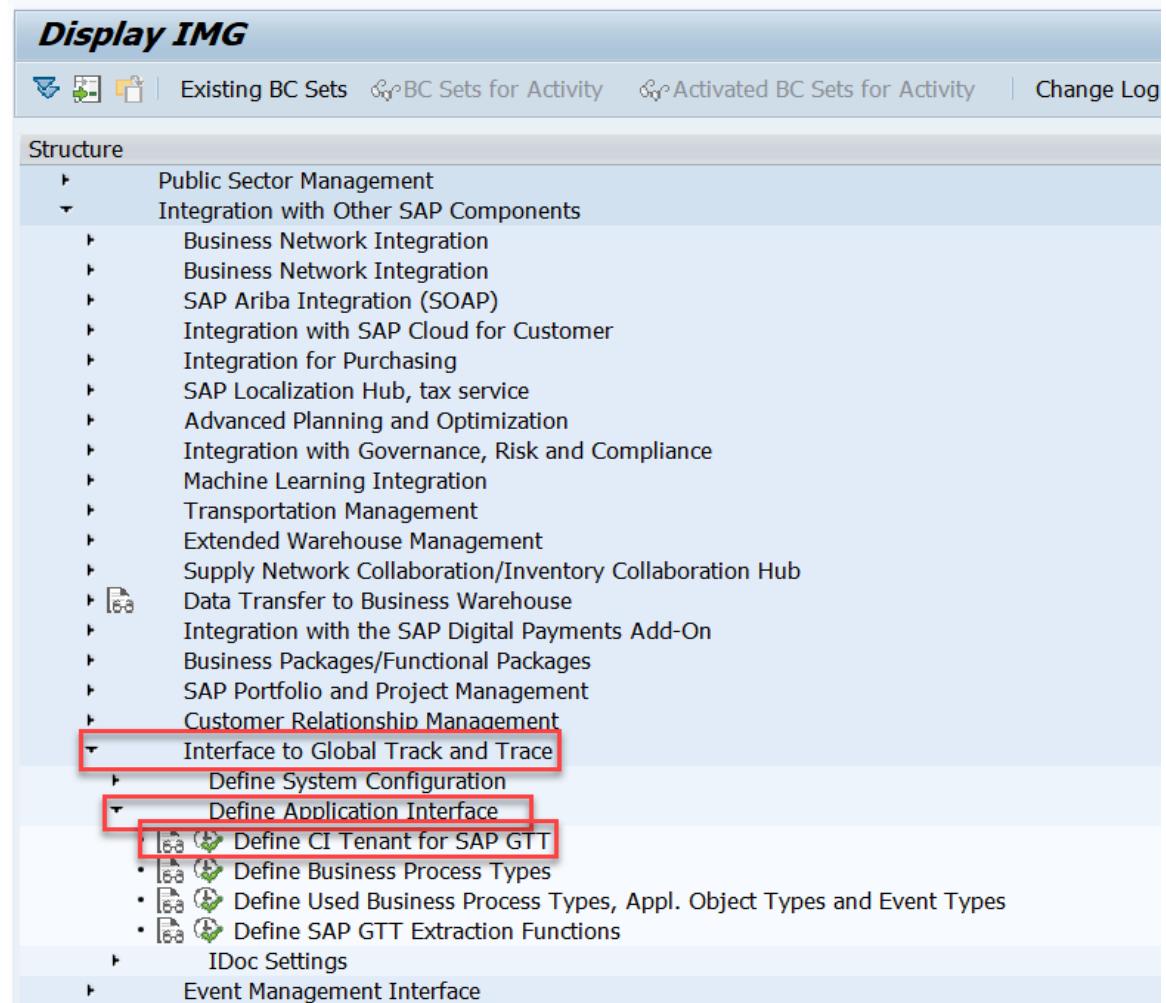
B2. Extractor Configuration



STEP 5: Define CI Tenant for GTT

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

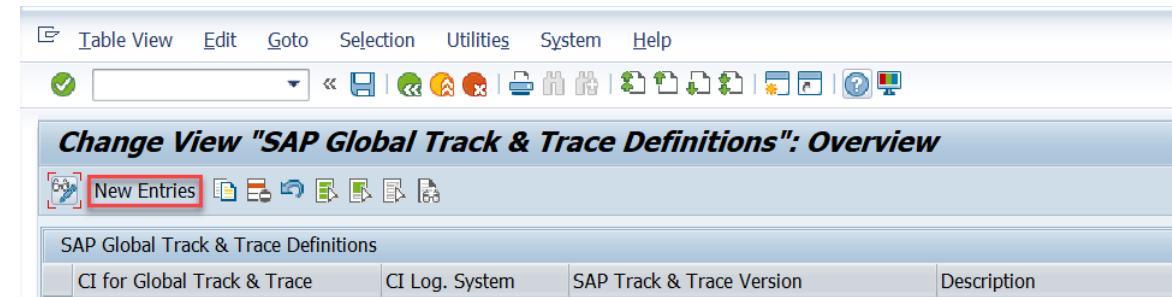
5-2: Choose activity
Define CI Tenant for SAP GTT



STEP 5: Define CI Tenant for GTT

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

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



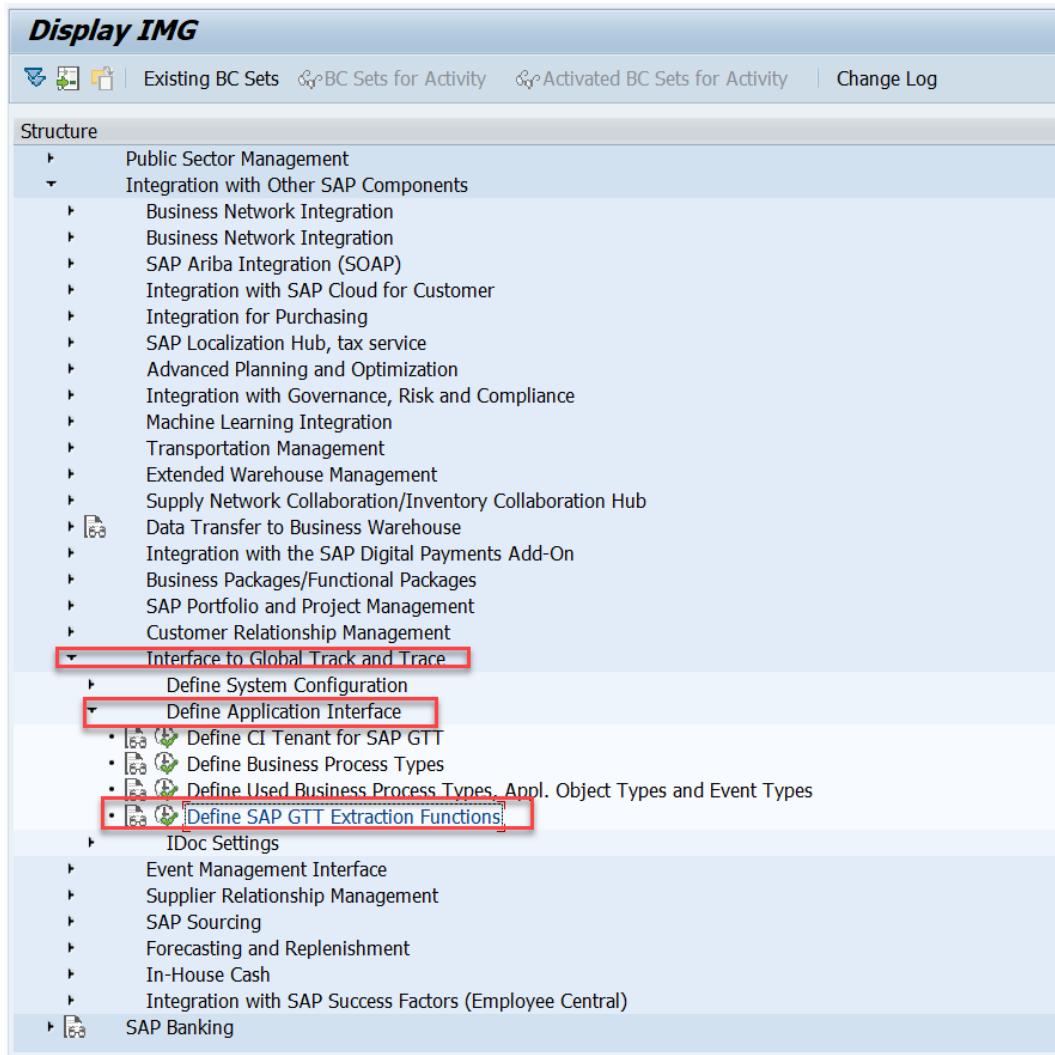
The screenshot shows the SAP GUI interface with the title bar "Display View 'SAP Global Track & Trace Definitions': Overview". Below the title bar is a toolbar with various icons. The main area displays a table titled "SAP Global Track & Trace Definitions" with four columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. A row in the table is highlighted with a yellow background. The first column of the highlighted row contains the value "ZGTTSOFINST", which is also highlighted with a red box.

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

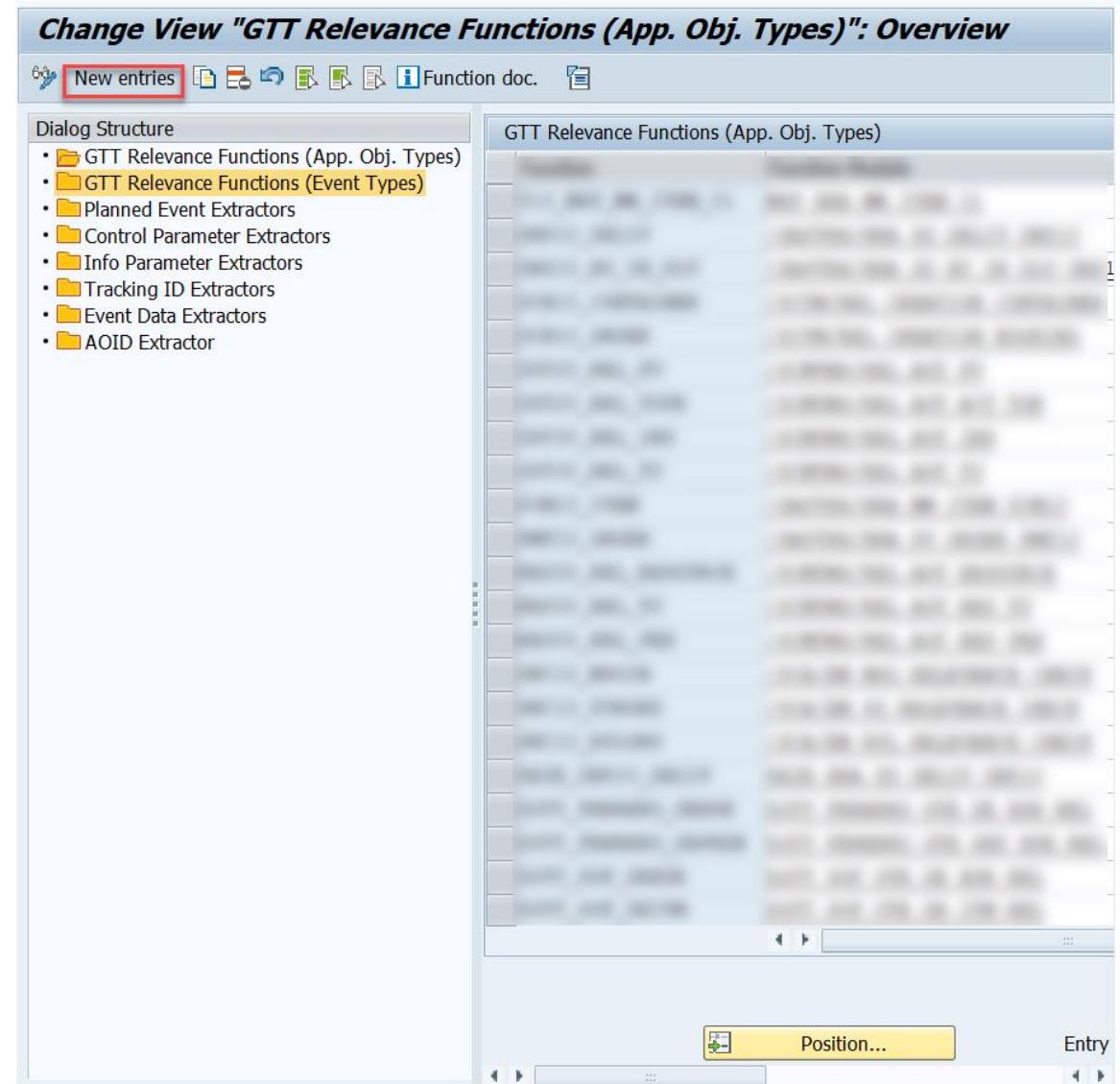
6-1: In **Display IMG** page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

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**



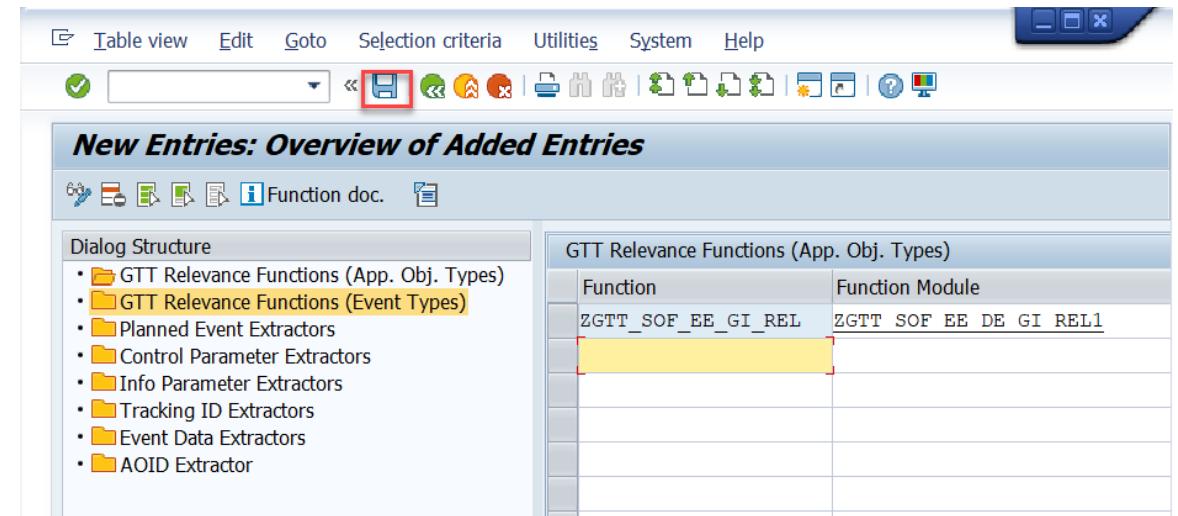
STEP 6: Define GTT Extraction Functions

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

GTT Relevance Functions (App. Obj. Types)	
Function	Function Module
ZGTT_SOF_EE_GI_REL	ZGTT_SOF_EE_DE_GI_REL1

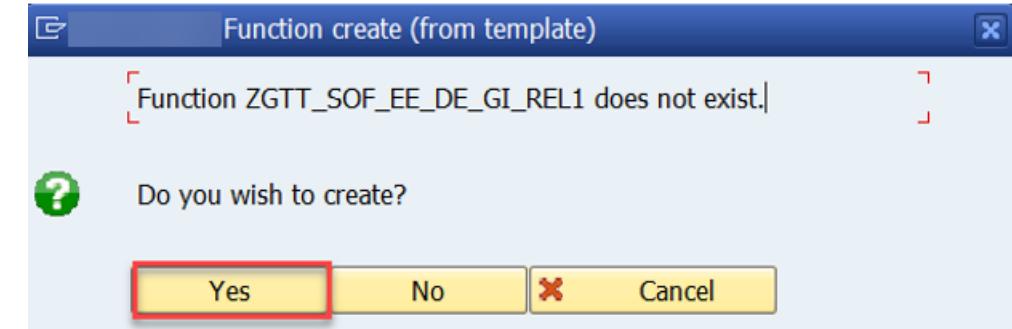
STEP 6: Define GTT Extraction Functions

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.

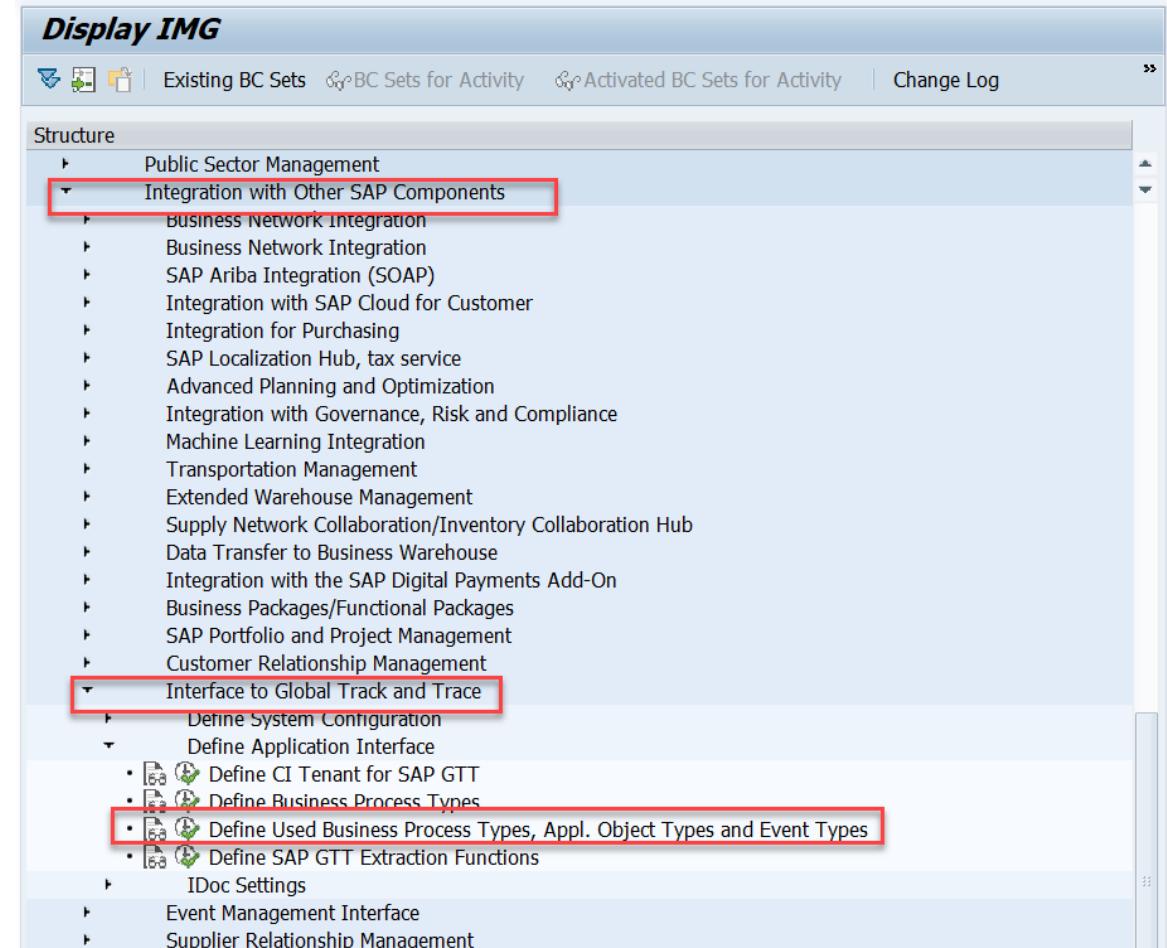
The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOEE_DE_GI_REL1". The "Function Module" field contains "ZGTT_SOEE_DE_GI_REL1" and is marked as "inactive". The "Source Code" tab is selected, displaying the ABAP code for the function module. The code defines a function named ZGTT_SOEE_DE_GI_REL1 with various parameters and table references. The "Object Name" tree on the left shows the function group ZGTT_SOEE and its sub-modules, with ZGTT_SOEE_DE_GI_REL1 highlighted. A red box highlights the function group dropdown in the Repository Browser and the function name in the object tree.

```
FUNCTION ZGTT_SOEE_DE_GI_REL1.  
  * Local Interface:  
  * REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM  
  * REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/AOTYPES  
  * REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER  
  * REFERENCE(I_APPTYPE_TAB) TYPE TRXAS_APPTYPE_TABS_WA  
  * REFERENCE(I_APP_OBJECT) TYPE TRXAS_APPOBJ_CTAB_WA  
  * EXPORTING  
  *   VALUE(E_RESULT) LIKE SY-BINPT  
  * TABLES  
  *   C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL  
  * EXCEPTIONS  
  *   PARAMETER_ERROR  
  *   RELEVANCE_DETERM_ERROR  
  *   STOP_PROCESSING  
  *--  
  * Top Include  
  * TYPE-POOLS:trxas.  
--  
ENDFUNCTION.
```

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

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

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



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

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

In the following:

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

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

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

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

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

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 CI For GTT V2 Integration system Sales Order Sa

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):

- General Data:** Bus. Proc. Type: ESC_SHIPMT, Event Type: ZGTT_SOF_CHIN_INT, Text: Check In Event for GTT SOF Integration System.
- 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.

Bottom Screenshot (Business Process Type: ESC_DELIV):

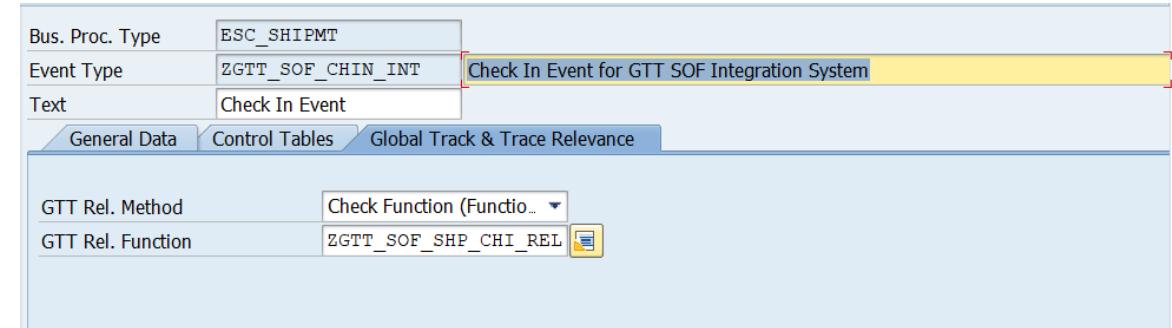
- General Data:** Bus. Proc. Type: ESC_DELIV, Event Type: ZGTT_SOF_PICKING_INT, Text: Picking Event for GTT SOF Integration System.
- Data Source for Events:** Main Obj. Table: DELIVERY_ITEM_NEW (highlighted with a red box), Master Table: DELIVERY_HEADER_NEW (highlighted with a red box). A callout box 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)A second field reference section is also shown below.

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

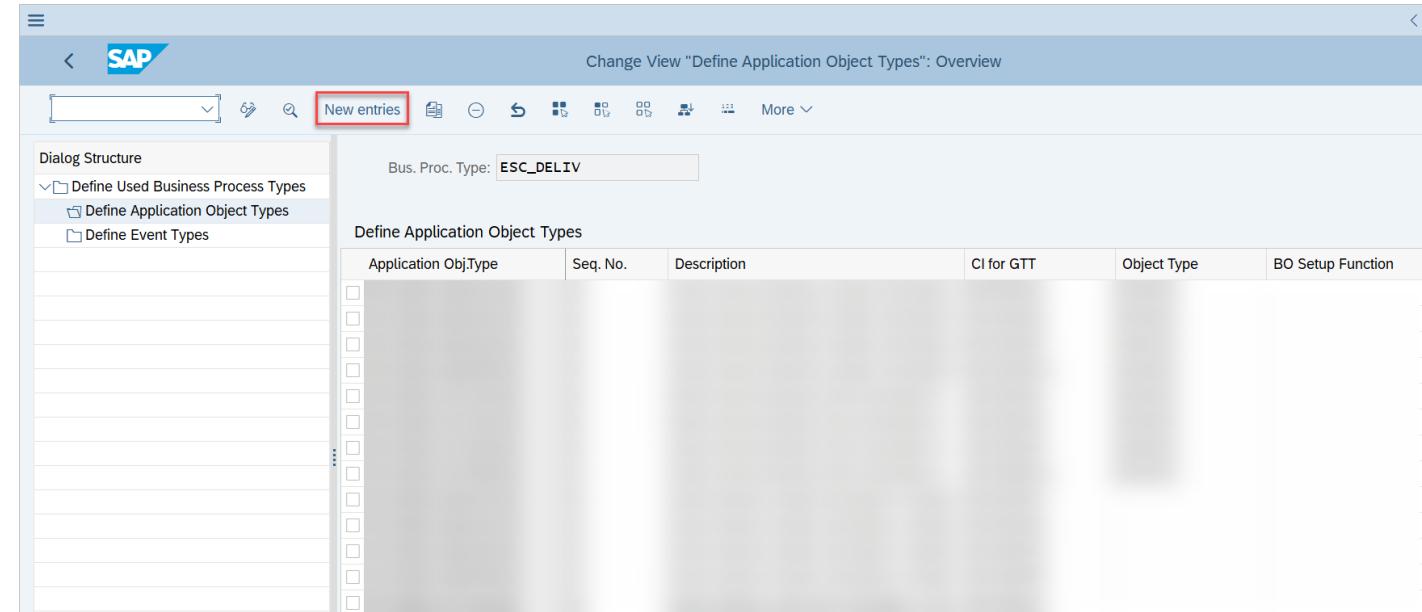
7-12: Double click **Define Application Object Types**

The screenshot shows the SAP Fiori interface for defining used business process types. The title bar reads "Change View 'Define Used Business Process Types': Overview". The top navigation bar includes "New Entries", "Copy As...", "Delete", "Undo Change", "Select All", "Select Block", "Deselect All", "Configuration Help", and "More...". On the left, a "Dialog Structure" sidebar lists "Define Used Business Process Type", "Define Application Object Type" (which is selected and highlighted with a red box), and "Define Event Types". The main area is a table titled "Define Used Business Process Types" with columns: Bus. Proc. Type, Update Mode, BPT Process Mode, and Description. The table lists various business process types, each with a checkbox in the first column. One row, "ESC_SHIPMT", has a checked checkbox and is also highlighted with a red box. The "Description" column provides a brief explanation for each entry.

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 navigation bar includes tabs for General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The General Data tab is active. The configuration fields include:

- 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
- Seq. No.: 20
- CI for GTT: ZGTTSOFINST (highlighted with a red box)
- Object Type: (empty input field)
- BO Setup Fnct.: (empty input field)
- Behavior section:
 - GTT Relevant (highlighted with a red box)
 - Stop AO Determ.
 - 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 interfaces. Both screenshots show a top navigation bar with tabs: General Data, Control Tables (which is selected), Object Identification, Global Track & Trace Relevance, and Parameter Setup.

Top Screenshot (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 input field]
- Control Tables tab is selected.
- Data Source for Created and Updated Objects:
 - Main Obj. Table: SHIPMENT_HEADER_NEW (highlighted with a red box)
 - Master Table: [empty input field]
 - 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
 - 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

Bottom Screenshot (Business Process Type: ESC_DELIV):

- General Data tab is selected.
- 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
 - Uplink Mode: R (highlighted with a red box)
 - Uplink Target Fld: VBELN
 - Uplink Const: [empty input field]

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

Determine AOID By Function

AOID Function:

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

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

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 at the bottom includes tabs for 'General Data', 'Control Tables', 'Object Identification', 'Global Track & Trace Relevance' (which is highlighted in blue), and 'Parameter Setup'. Under the 'Global Track & Trace Relevance' tab, there are two configuration fields: '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 enclosed in a red rectangular box.

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' (highlighted with a red box). Other fields include 'Tr.ID Tab. Type', 'Tr. ID Code Set', and 'Trk.ID Function' (ZGTT_TID_DE_ITEM). To the right, there is a 'Tracking ID Fld:' input field. Under 'Parameter Setup', three function names are listed: 'Ctrl Data Function' (ZGTT_OTE_DE_ITEM), 'Info Data Function' (empty), and 'Planned Event Function' (ZGTT_EE_DE_ITM), all highlighted with red boxes.

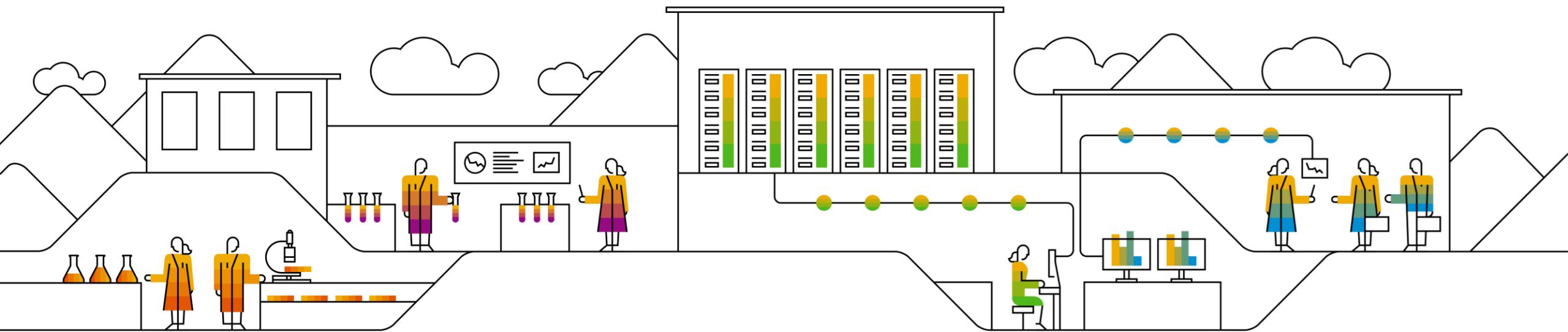
C) Download ABAP Code from GitHub

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

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

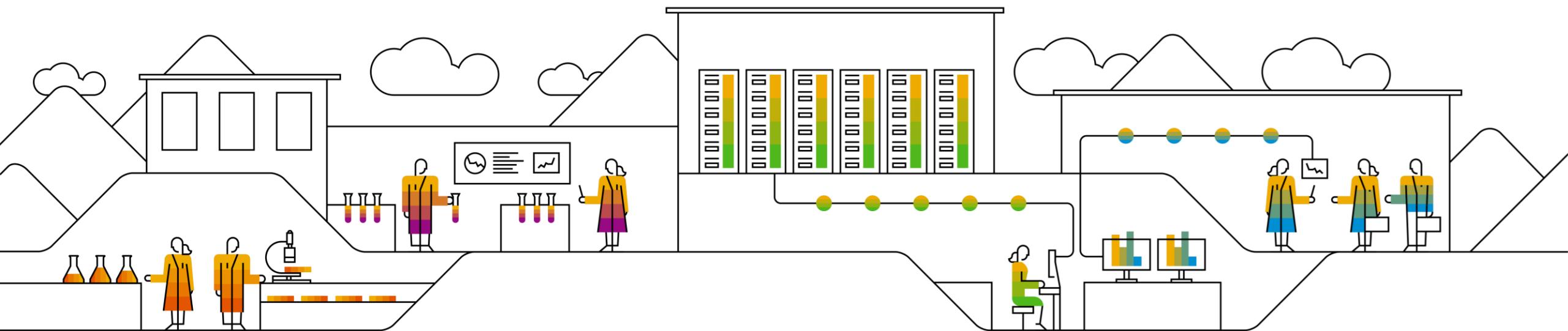
C3. Download Another ABAP Code from GitHub (Only for TPOF)

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**.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar contains links for "Getting Started", "Setup", "Online Projects", "Offline Projects", and "Reference". The main content area starts with a "Summary" section, followed by a note about abapGit existing in two flavours: "standalone" and "developer". The "Installation" section is highlighted with a red border. It contains a link to "Improve this page" and a "Prerequisites" section stating "abapGit requires SAP BASIS version 702 or higher". The "Install standalone version" section is also highlighted with a red border and contains the following steps:

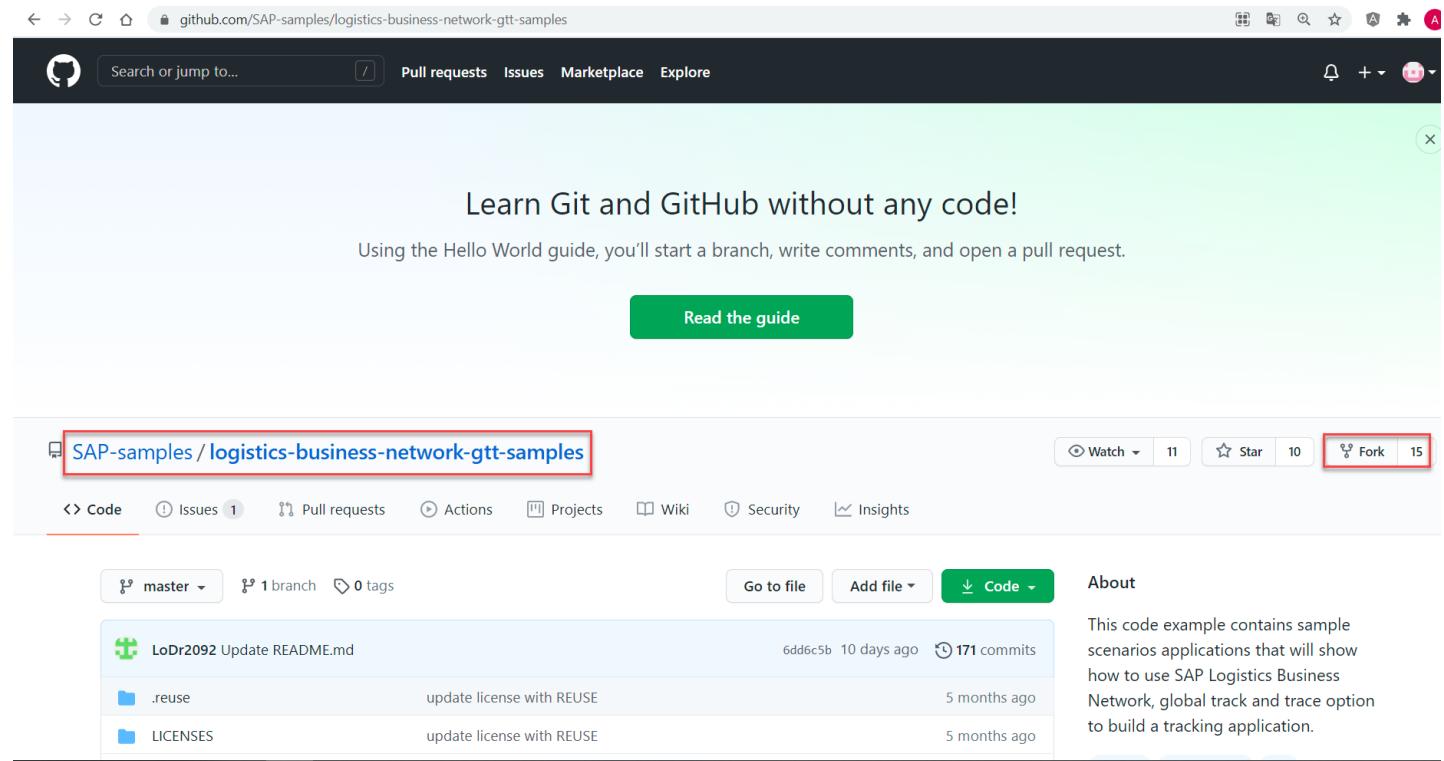
1. Download the ABAP code(right click -> save-as) to a file.
2. Via SE38 or SE80 , create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version.
3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload
4. Activate

Below these steps, it says "Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. \$ZABAPGIT)." and "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’

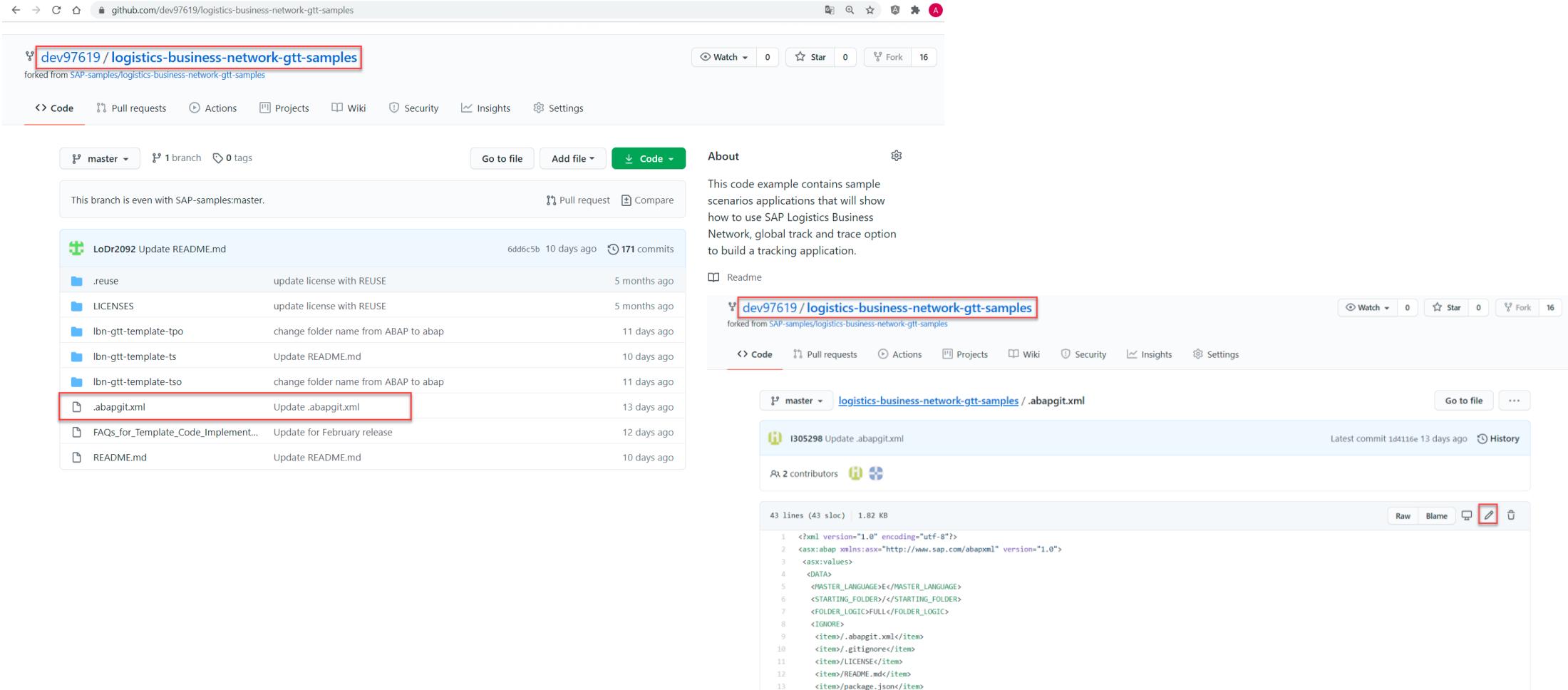
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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 history lists several changes, including one by LoDr2092 that updates the README.md file. The commit 'Update .abapgit.xml' is highlighted with a red box. 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, and building a tracking application. It includes links to 'Readme', 'Releases', and 'Packages'.

File	Description	Time
LoDr2092 Update README.md	update license with REUSE	6dd6c5b 10 days ago
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
.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

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 repository page shows a list of commits, with one commit highlighted: `I305298 Update .abapgit.xml`. This commit was made by `dev97619` 13 days ago. The commit message is `Update .abapgit.xml`. The file `.abapgit.xml` is shown in the code editor with the following XML content:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER>./</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 commit dialog for the '.abapgit.xml' file in the 'logistics-business-network-gtt-samples' repository. The code editor on the left shows the XML configuration file with a specific line highlighted and boxed in red. The commit dialog on the right contains fields for the commit message ('Update .abapgit.xml'), an optional extended description, and two radio button options for committing: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request.' A large green 'Commit changes' button is at the bottom of the dialog, which is also boxed in red.

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

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

<> Edit file Preview changes

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

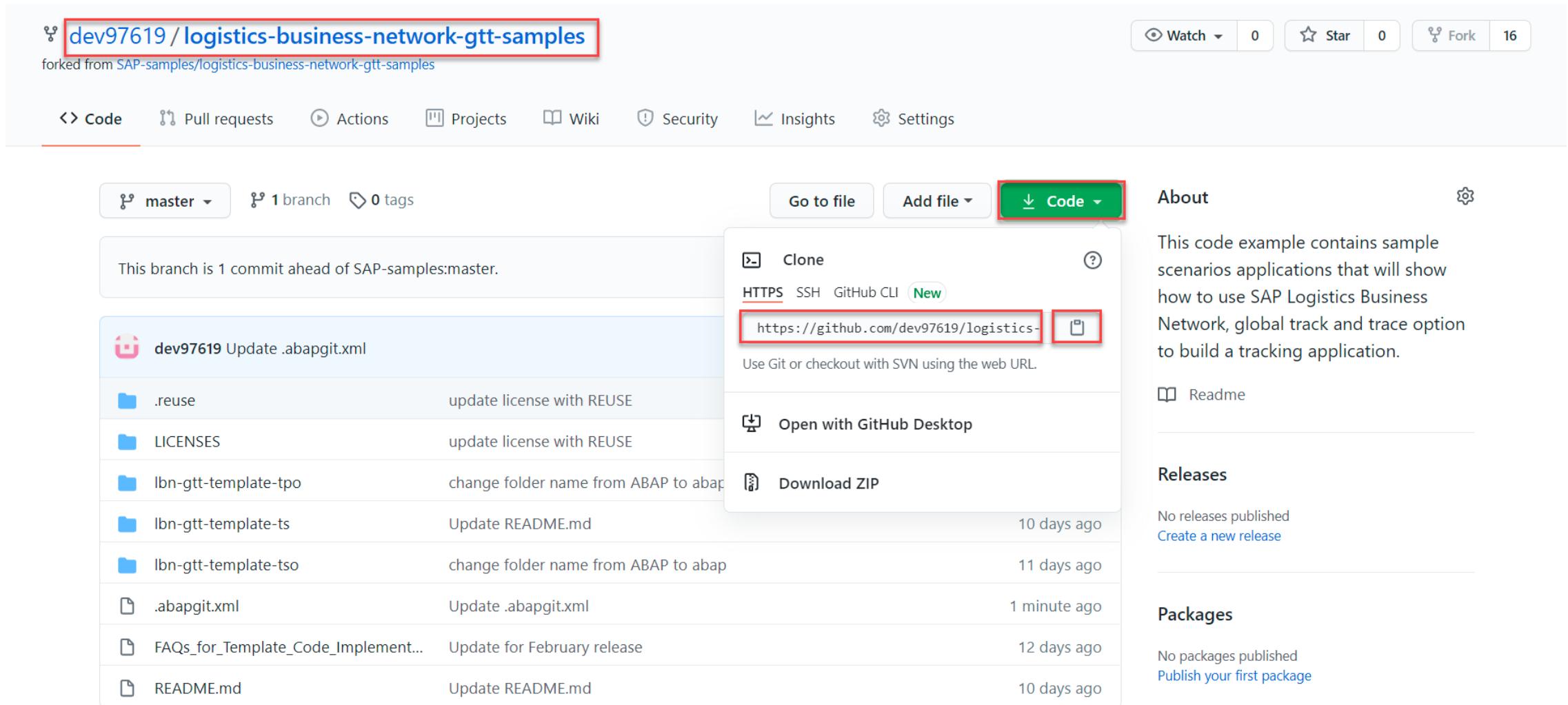
Commit directly to the master branch.

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

Commit changes Cancel

STEP 3: 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. In the top right, there are 'Watch' (0), 'Star' (0), and 'Fork' (16) buttons. Below the tabs, it shows 'master' branch, 1 branch, and 0 tags. A message indicates the branch is 1 commit ahead of SAP-samples:master. On the left, a list of commits is shown:

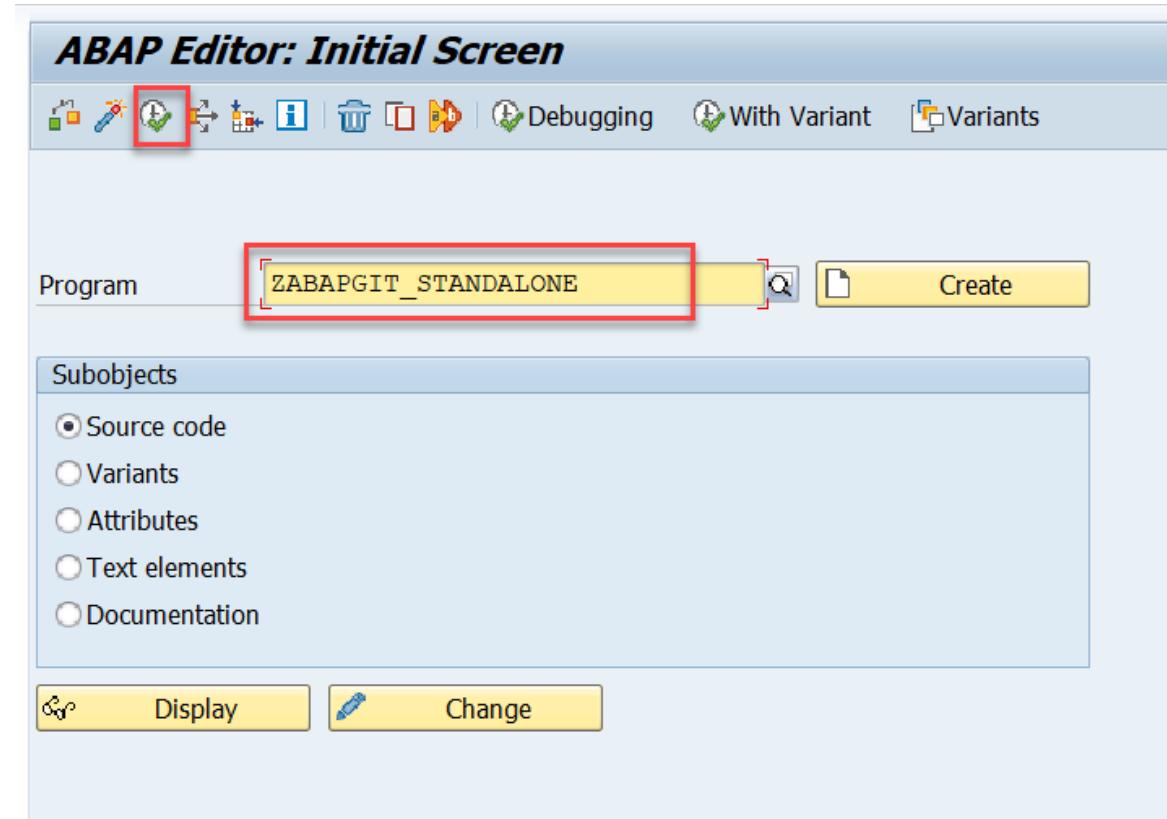
- dev97619 Update .abapgit.xml
- .reuse update license with REUSE
- LICENSES update license with REUSE
- Ibn-gtt-template-tpo change folder name from ABAP to abap
- Ibn-gtt-template-ts Update README.md
- Ibn-gtt-template-tso change folder name from ABAP to abap
- .abapgit.xml Update .abapgit.xml
- FAQs_for_Template_Code_Implement...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

On the right, the 'Code' dropdown menu is open, showing options: 'Clone' (with a red box around the URL), 'Add file', 'Go to file', and 'Download ZIP'. The URL 'https://github.com/dev97619/logistics...' is highlighted with a red box. The 'About' section describes the repository as containing sample scenarios applications for SAP Logistics Business Network, global track and trace option to build a tracking application. It also links to 'Readme', 'Releases' (no releases published), and 'Packages' (no packages published).

STEP 4: Download ABAP Code from GitHub

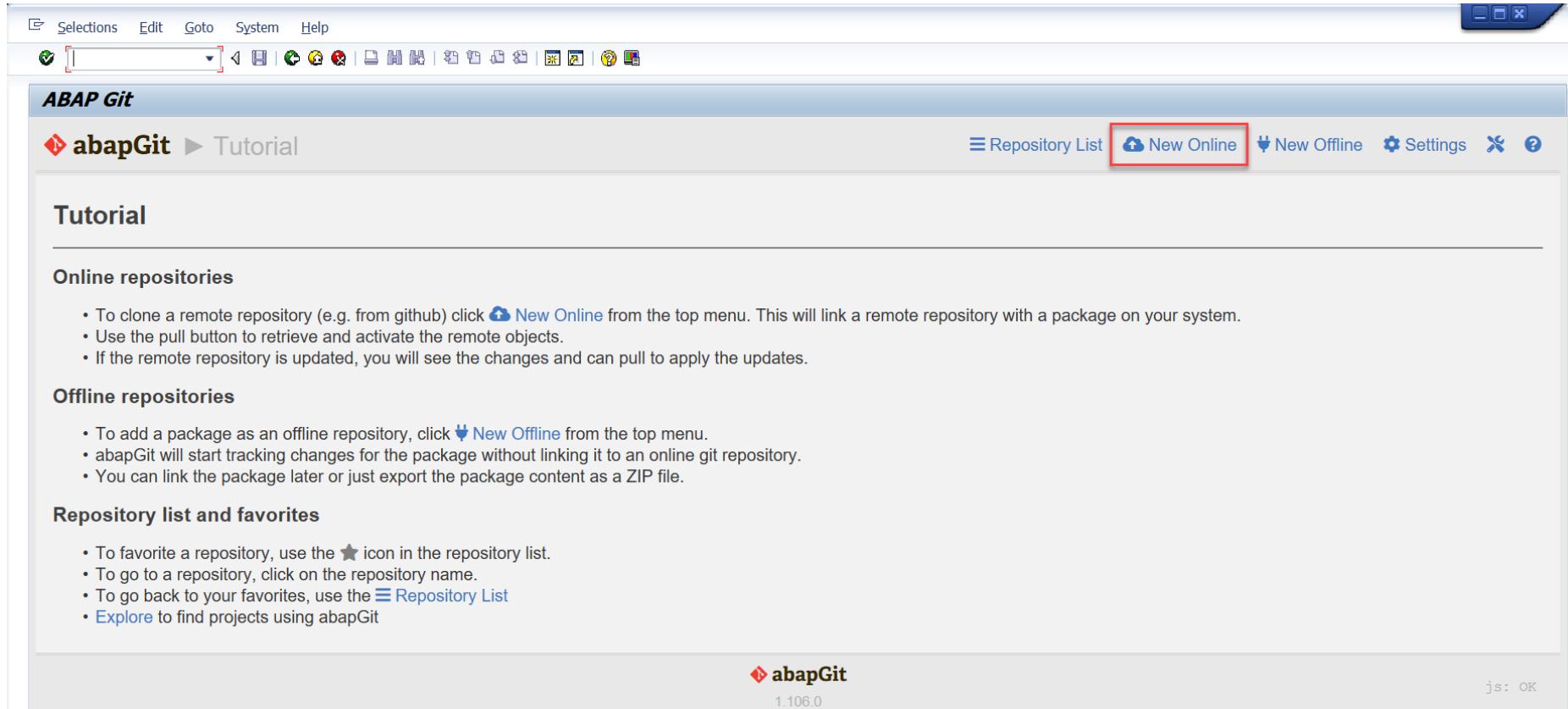
4-1: Enter T-code *SE38* and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

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

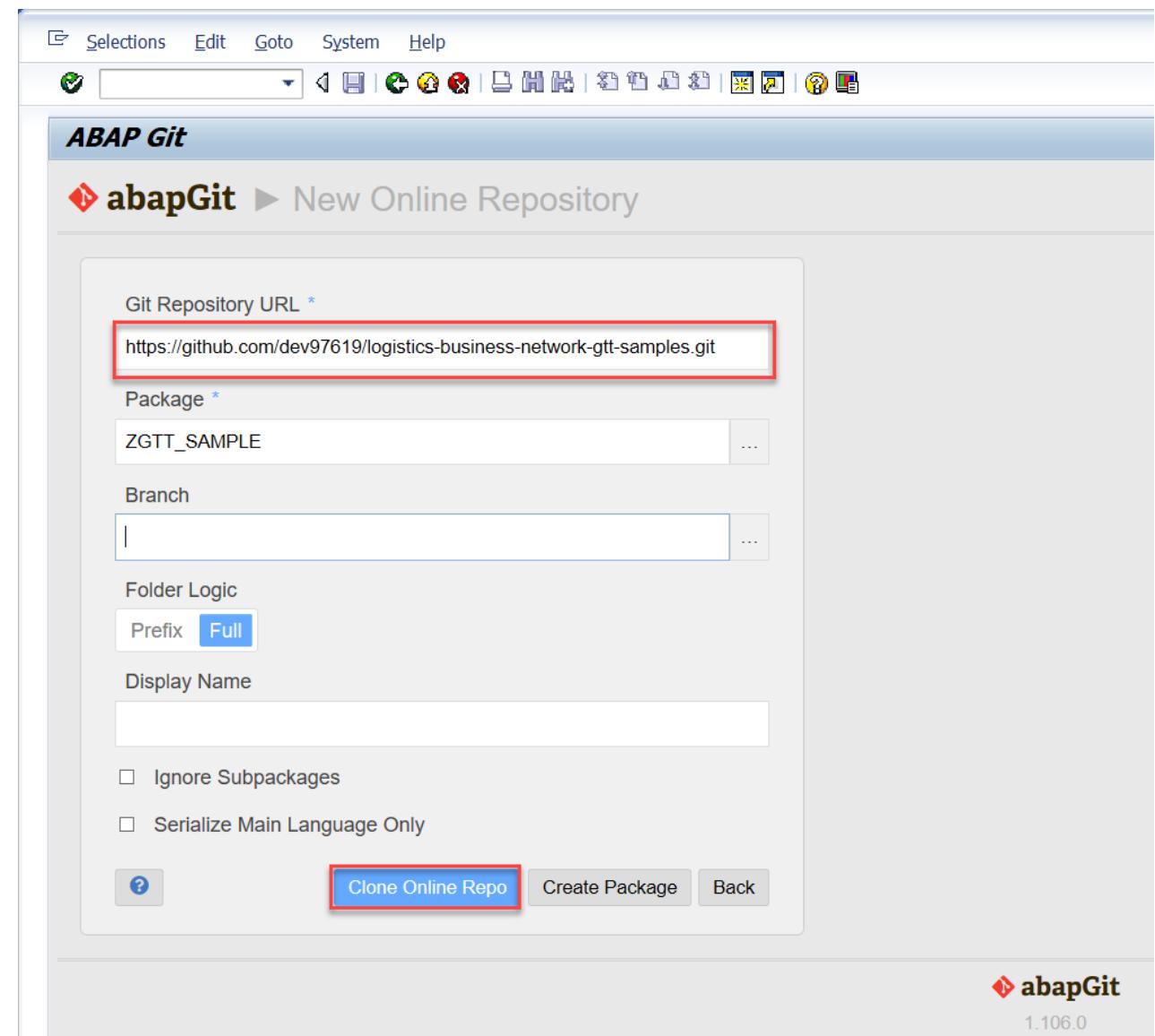
Caution:

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

4-5: Fill in the **Package** where you want to create the new ABAP code. If the package does not exist yet, click **Create package** to create it.

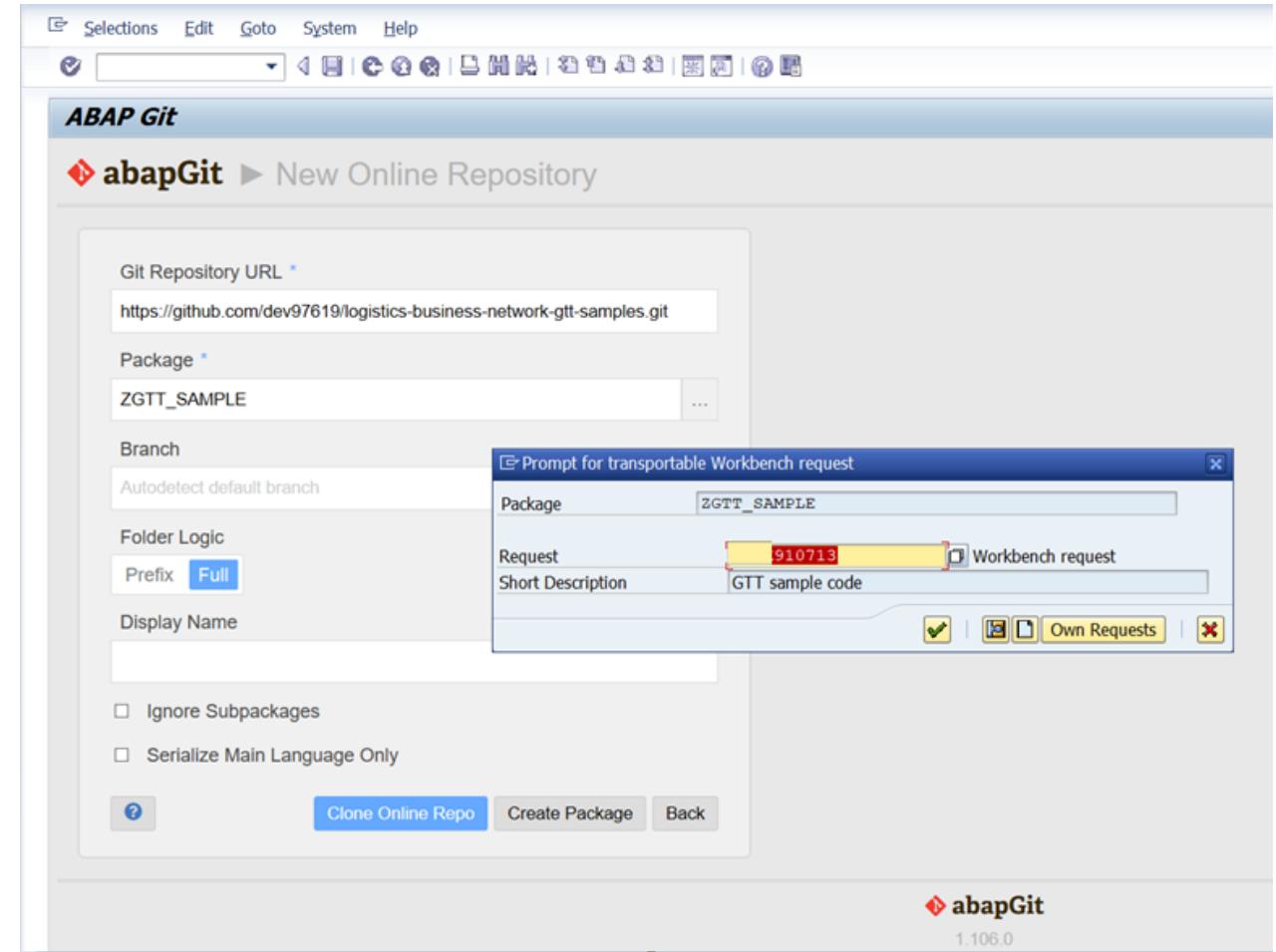
4-6: Set *Full* for **Folder Logic**

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



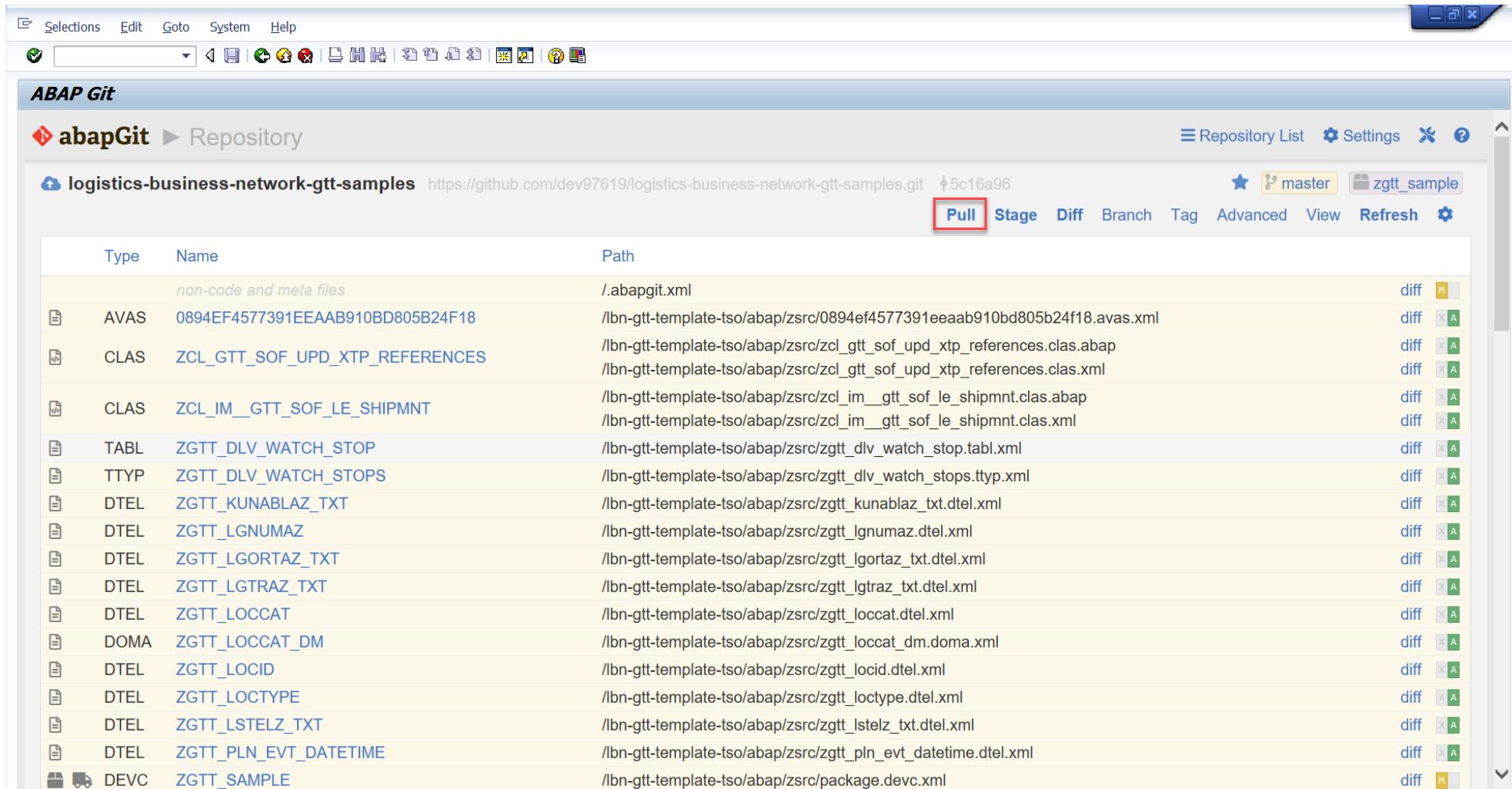
STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

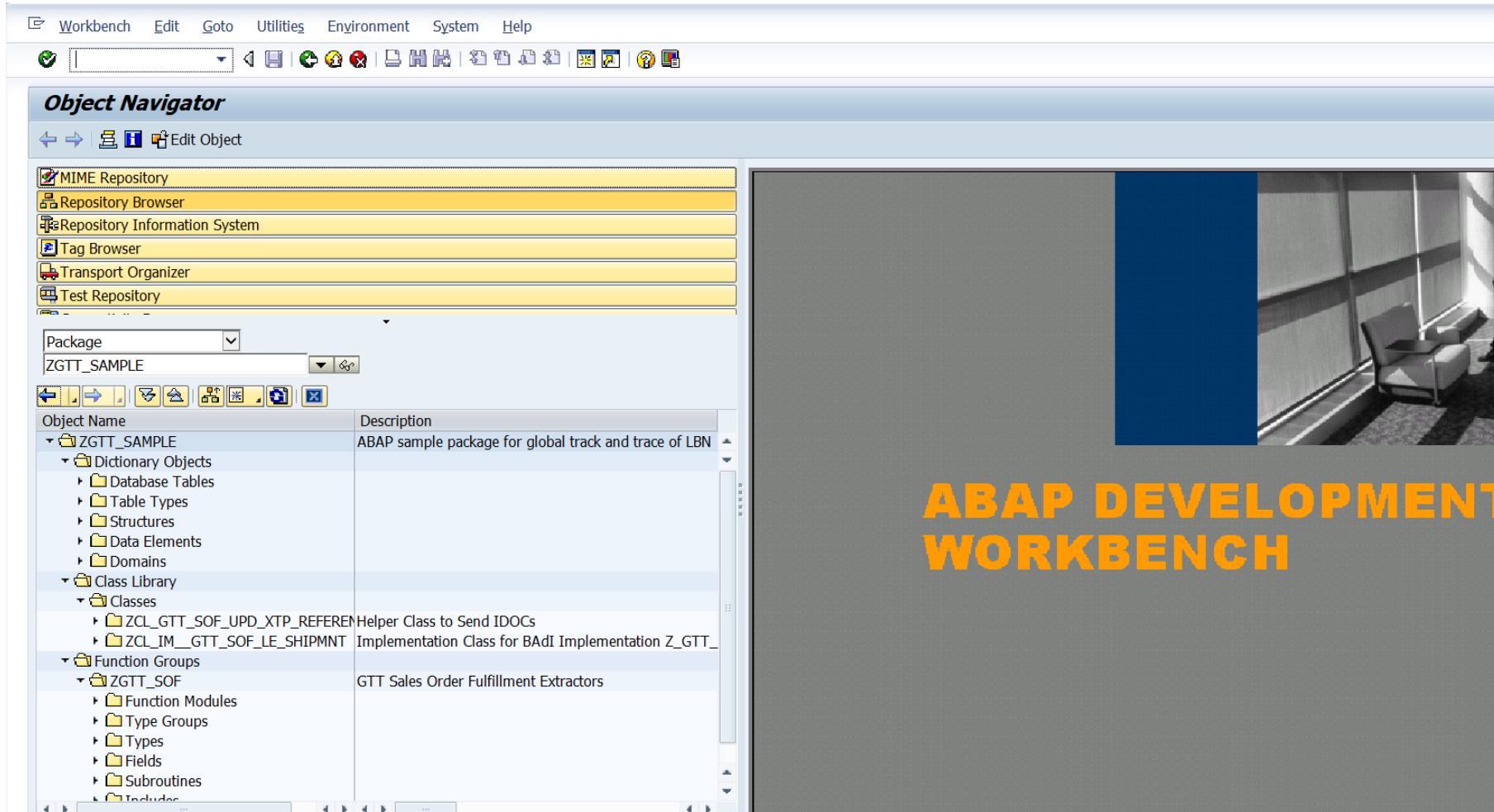


The screenshot shows the ABAP Git interface within an SAP application. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. The main menu bar has icons for 'File', 'Edit', 'Goto', 'System', and 'Help'. The toolbar below the menu bar includes icons for 'New', 'Open', 'Save', 'Print', 'Copy', 'Paste', 'Find', 'Replace', 'Select All', 'Find Next', 'Find Previous', 'Find in Path', 'Find in File', 'Find in Project', 'Find in Database', 'Find in Help', and 'Find in Help'. The title 'ABAP Git' is displayed above the repository list. The repository 'abapGit' is selected, and the path 'Repository' is shown. The repository 'logistics-business-network-gtt-samples' is listed with the URL 'https://github.com/dev97619/logistics-business-network-gtt-samples.git' and the commit hash '5c16a96'. The status bar indicates 'master' and 'zgtt_sample'. The toolbar at the top right includes 'Repository List', 'Settings', 'X', '?', 'Pull' (which is highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a gear icon. The main content area displays a table of files with columns 'Type', 'Name', and 'Path'. Each row shows a file's type (e.g., AVAS, CLAS, TABL, DTEL, DOMA, DTEP, DEV), name, and path. To the right of each row are 'diff' and 'A' buttons. A vertical scrollbar is visible on the right side of the table.

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

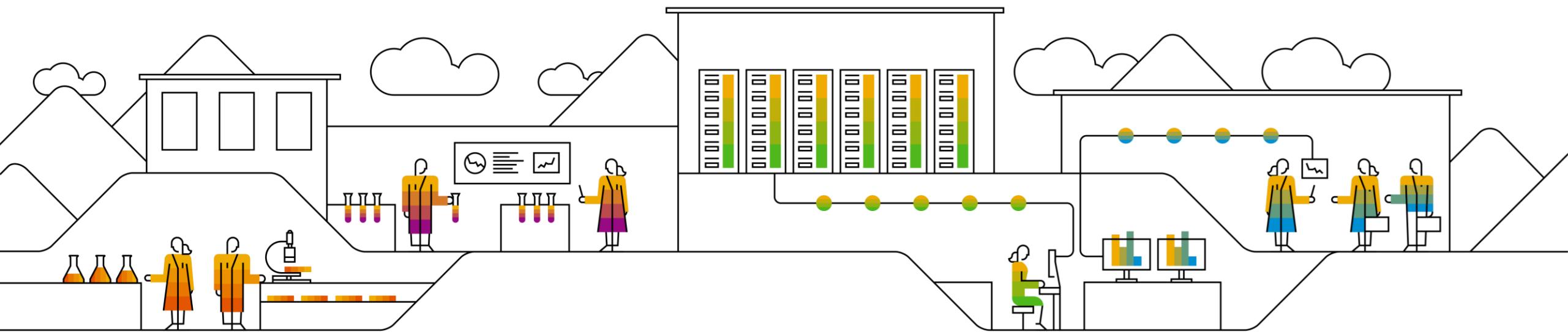
STEP 4: Download ABAP Code from GitHub

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)

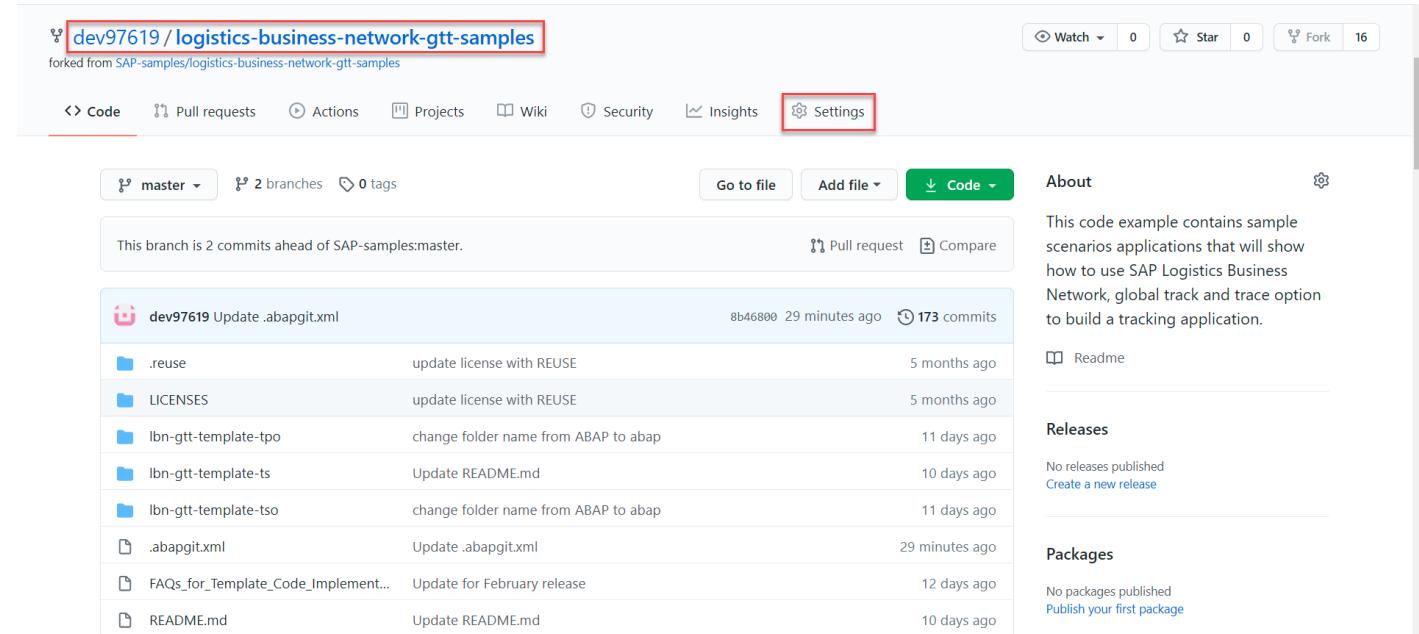


STEP 1: Delete the User's Account Repository

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.

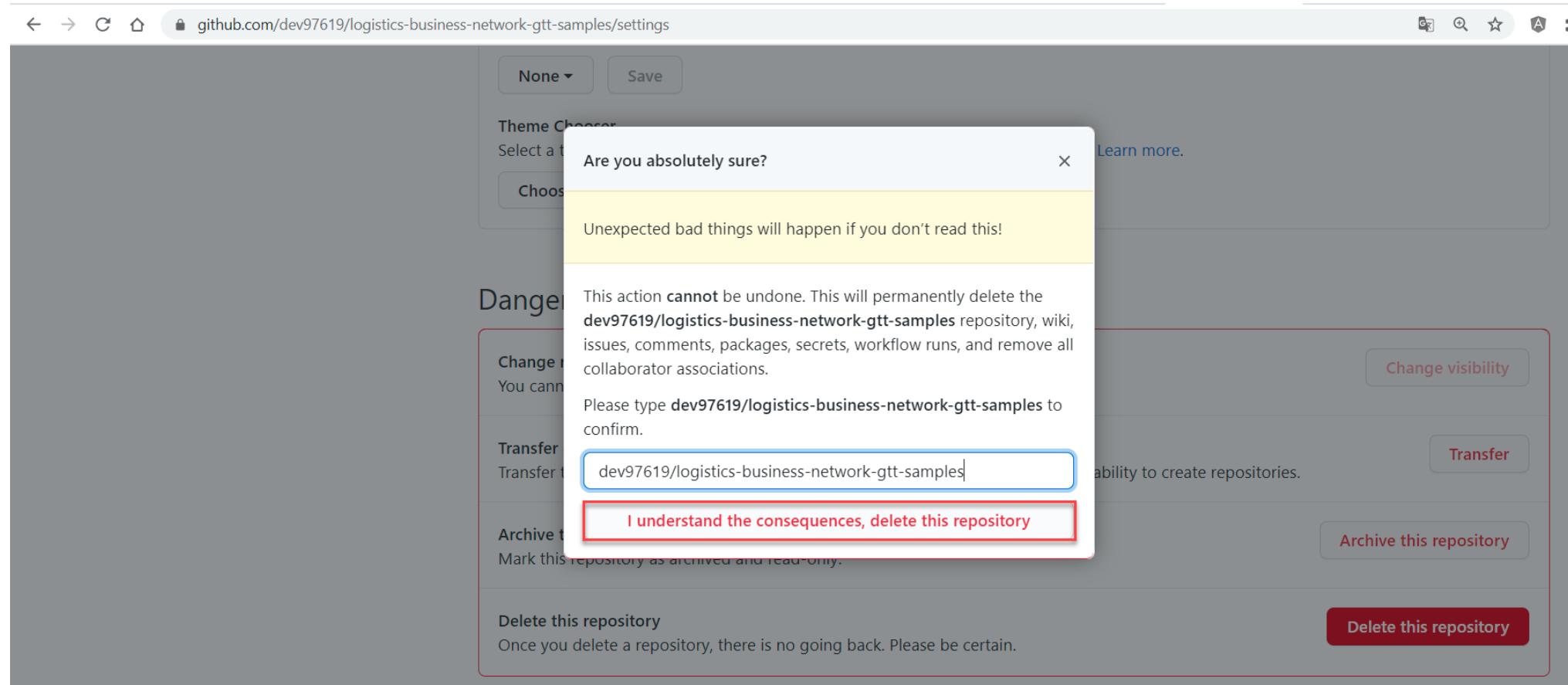
The screenshot shows a GitHub repository settings page for 'github.com/dev97619/logistics-business-network-gtt-samples/settings'. At the top, there is a 'Theme Chooser' section with a 'None' dropdown, a 'Save' button, and a 'Choose a theme' button. Below this is a 'Danger Zone' section with four options:

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

The 'Delete this repository' button is highlighted with a red border.

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

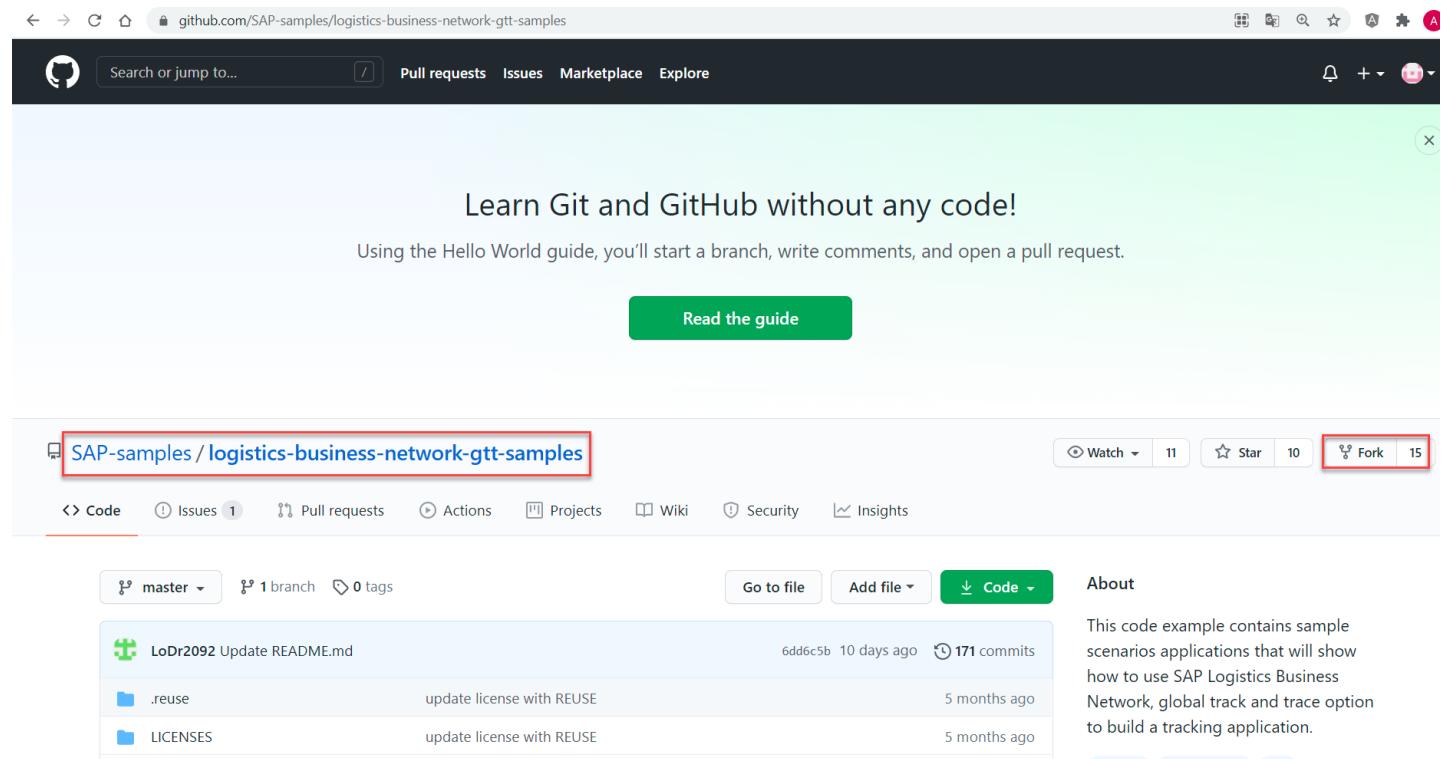
1-5: The user's account repository is deleted.

The screenshot shows a GitHub-like application interface. At the top, there is a dark header bar with the GitHub logo, a search bar, and navigation links for "Pull requests", "Issues", "Marketplace", and "Explore". To the right of the header are icons for notifications, a plus sign, and a profile picture. Below the header, a light blue banner displays a success message: "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." An "X" icon is at the end of this banner. The main content area has a white background. On the left, there is a sidebar with sections for "Create your first project" (with "Create repository" and "Import repository" buttons), "Working with a team?", and a "Create an organization" button. A large, semi-transparent green overlay is centered over the main content. It features the text "Learn Git and GitHub without any code!" in bold, followed by a descriptive paragraph: "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below this text are two buttons: a green one labeled "Read the guide" and a white one labeled "Start a project".

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

Pull request Compare

File	Description	Time
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’

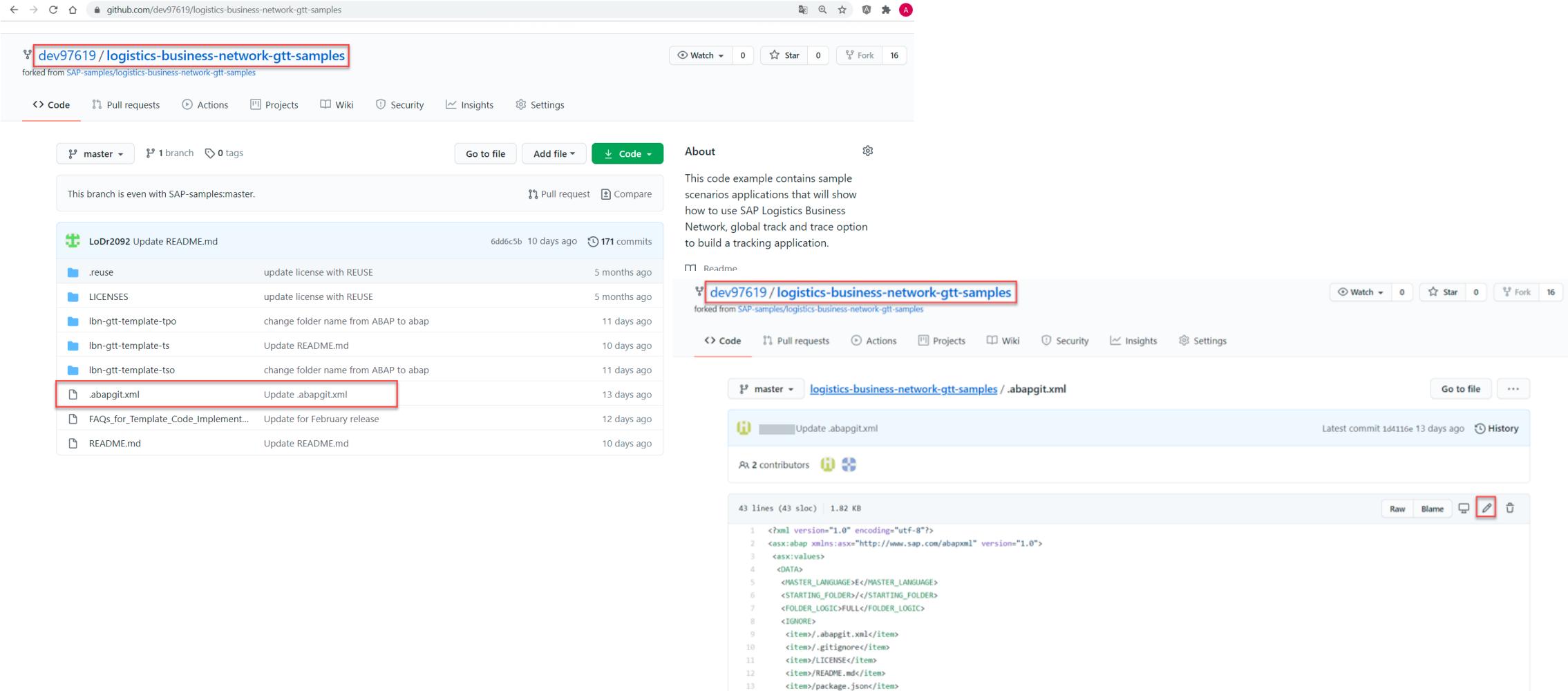
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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 a 'Readme' link. Below that is a 'Releases' section stating 'No releases published' and a 'Create a new release' link. Finally, there is a 'Packages' section stating 'No packages published' and a 'Publish your first package' link.

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
.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

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

3-2: Click  button to edit the file.



The screenshot shows two GitHub repository pages. The top page is for the forked repository `dev97619 / logistics-business-network-gtt-samples`. The bottom page is for the original repository `logistics-business-network-gtt-samples`.

Top Repository (Forked):

- Branch:** master (1 branch, 0 tags)
- Commits:**
 - 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)
 - lbn-gtt-template-tpo change folder name from ABAP to abap (11 days ago)
 - lbn-gtt-template-ts Update README.md (10 days ago)
 - lbn-gtt-template-tso change folder name from ABAP to abap (11 days ago)
 - .abapgit.xml** Update .abapgit.xml (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.

Bottom Repository (Original):

- Branch:** master (logistics-business-network-gtt-samples / .abapgit.xml)
- Commit:** Update .abapgit.xml (Latest commit 1d4116e, 13 days ago) 
- Contributors:** 2 contributors 
- File Content:**

```
43 lines (43 sloc) 1.82 KB
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3 <asx:values>
4 <DATA>
5 <MASTER_LANGUAGE>€</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 'logistics-business-network-gtt-samples'. The repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. In the code editor, the '.abapgit.xml' file is open, showing XML configuration. Line 6 contains the path '<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>'. A red box highlights this line. To the right, a 'Commit changes' dialog is displayed. It includes fields for a commit message ('Update .abapgit.xml'), an optional description, and two radio button options: one selected for committing directly to the 'master' branch, and another for creating a new branch. A green 'Commit changes' button is at the bottom left of the dialog, also highlighted with a red box.

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

-o- Commit directly to the master branch.

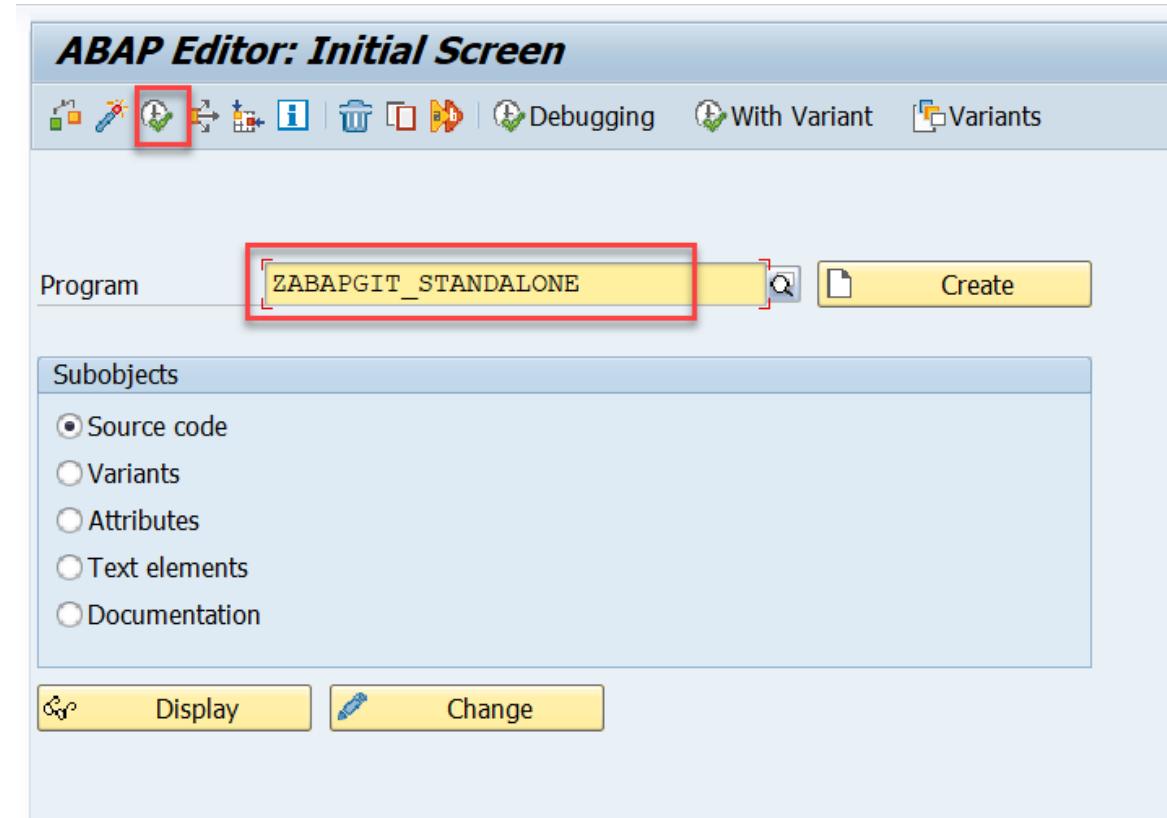
! Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes Cancel

STEP 4: Update ABAP Code from GitHub

4-1: Enter T-code *SE38* and fill in the report name *ZABAPGIT_STANDALONE*.

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



STEP 4: Update ABAP Code from GitHub

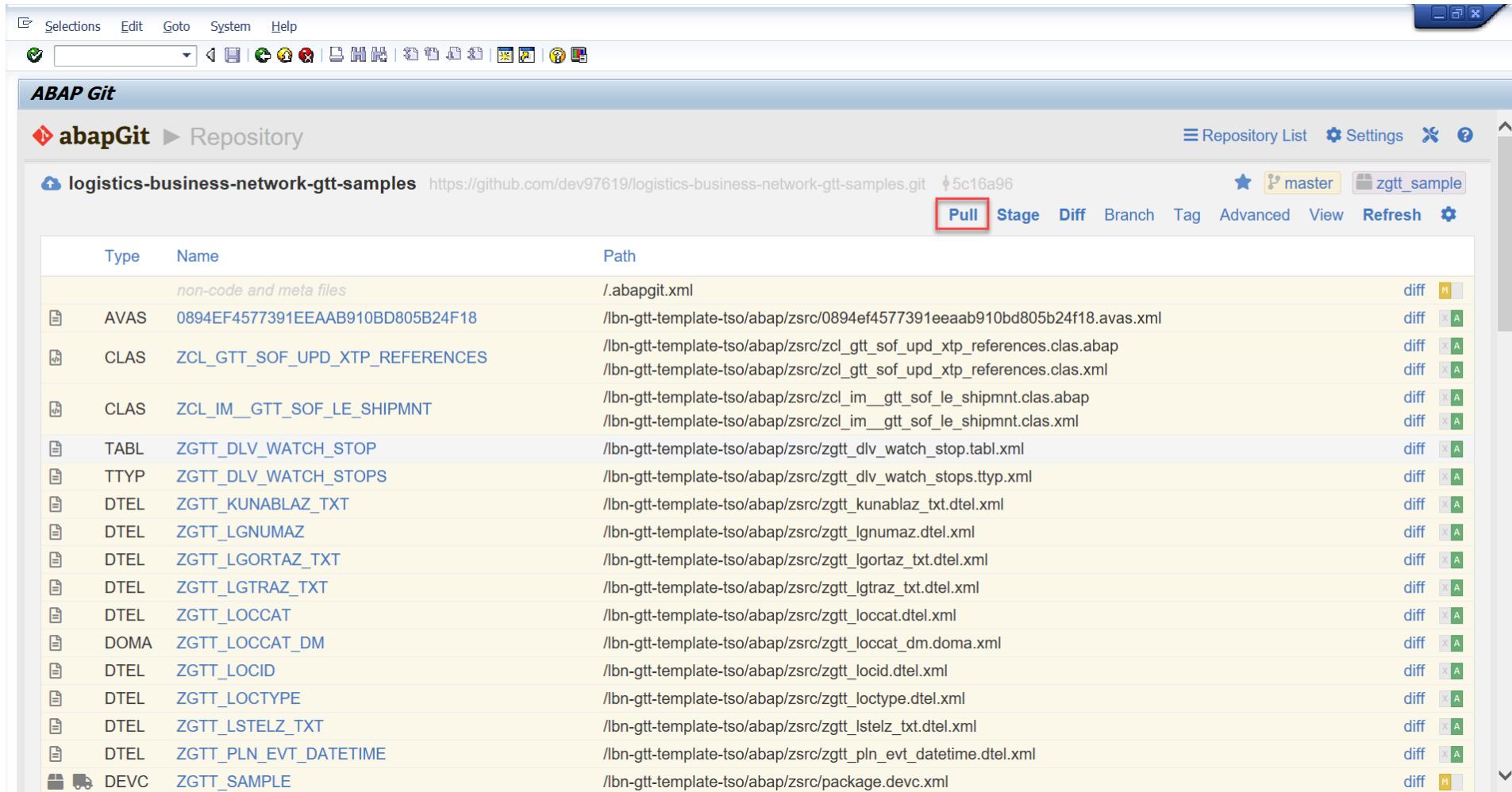
4-3: Check if the URL is changed or not after your recreation of repository copy. Access the TSOF Repository by clicking  button.



The screenshot shows the SAP ABAP Git interface. The title bar includes standard SAP menu items like Selections, Edit, Goto, System, and Help. Below the menu is a toolbar with various icons. The main area is titled "ABAP Git" and shows a "Repository List". On the left, there's a sidebar with a star icon and a cloud icon, followed by the repository name "logistics-business-network-gtt-samples". To the right of the repository name is its URL: "github.com/dev97619/logistics-business-network-gtt-samples.git". Underneath the URL are the package name "zgtt_sample" and the branch name "master". To the right of the branch name are four buttons: "Check", "Stage", "Patch", and "Settings", with the "Settings" button highlighted by a red box. At the bottom right of the list area is a blue "More" button with a red box around it. The footer of the interface displays the "abapGit" logo and version "1.106.0", along with a status message "js: OK".

STEP 4: Update ABAP Code from GitHub

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

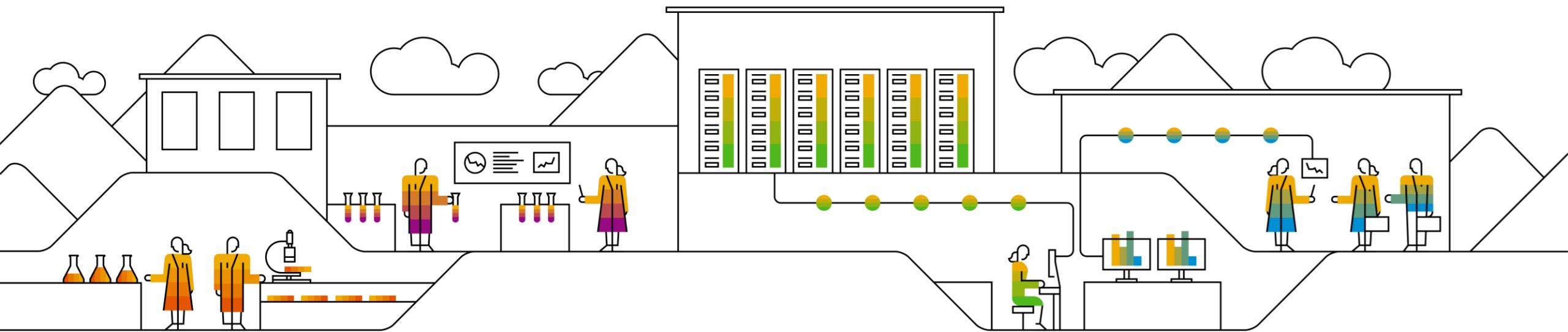


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git". Underneath, the repository path "abapGit > Repository" is shown, along with the URL "logistics-business-network-gtt-samples" and the commit hash "5c16a96". To the right of the URL, there are buttons for "master" and "zgtt_sample". A navigation bar below the URL includes "Pull", "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The main area is a table with columns "Type", "Name", and "Path". The "Pull" button is highlighted with a red box. The table lists several files and their paths, such as "AVAS", "ZCL_GTT_SOUPD_XTP_REFERENCES", "ZCL_IM_GTT_SOUPD_LE_SHIPMNT", etc., each with a "diff" link and a status indicator (M, A, or C).

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
CLAS	ZCL_IM_GTT_SOUPD_LE_SHIPMNT	//lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	A
TABL	ZGTT_DLV_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	A
TTYP	ZGTT_DLV_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_kunablaTxt.dtel.xml	A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml	A
DTEL	ZGTT_LGORAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgoraz_txt.dtel.xml	A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	A
DTEL	ZGTT_LOCCAT	//lbn-gtt-template-tso/abap/zsrc/zggt_locat.dtel.xml	A
DOMA	ZGTT_LOCCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zggt_locat_dm.doma.xml	A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.datetime.dtel.xml	A
DEV	ZGTT_SAMPLE	//lbn-gtt-template-tso/abap/zsrc/package.devcl.xml	M

C) Download ABAP Code from GitHub

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



STEP 1: Fork Sample Code Repository

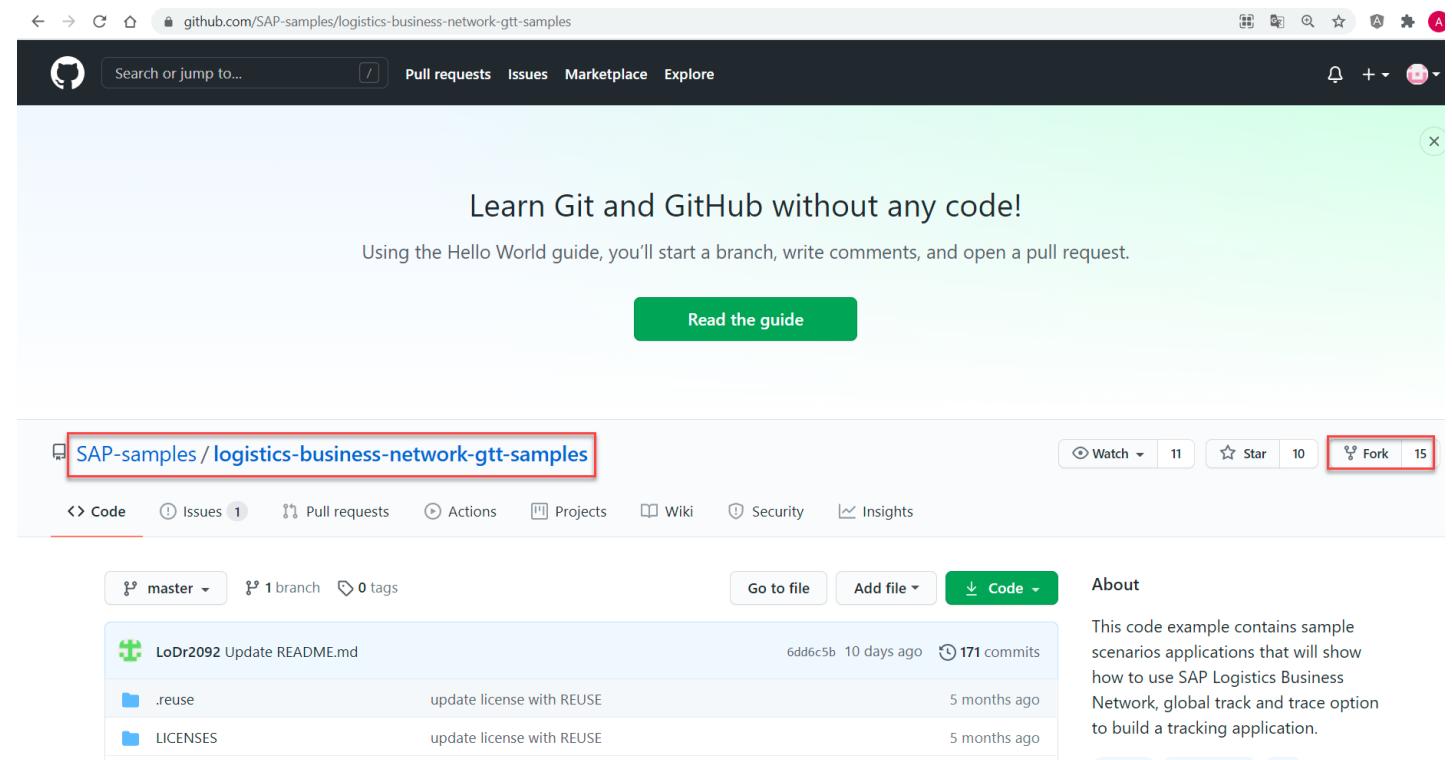
Prerequisite:

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’

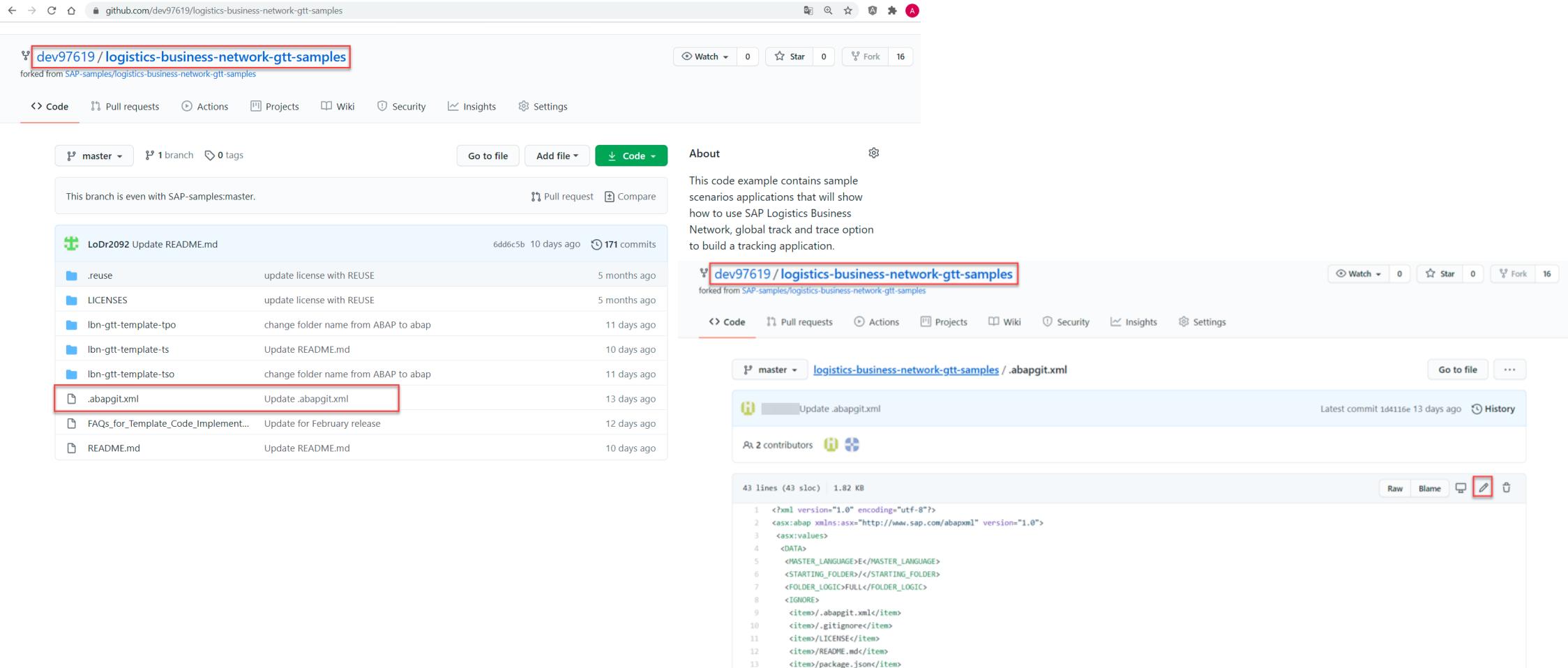
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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

Commit	Message	Date
LoDr2092 Update README.md	6dd6c5b 10 days ago	171 commits
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
.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

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

2-2: Click  button to edit the file.



The screenshot shows two views of a GitHub repository. The top view is the repository page for `dev97619 / logistics-business-network-gtt-samples`. The bottom view is a detailed look at the `.abapgit.xml` file within the repository.

Repository Page:

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

.abapgit.xml File View:

- File name: `Update.abapgit.xml`
- Contributors: At 2 contributors
- Commit: 1d4116e 13 days ago
- Code editor interface with a red box around the edit icon.

```
<?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.mdc</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 `logistics-business-network-gtt-samples` repository. The code editor on the left displays the XML configuration, with the line `<STARTING_FOLDER>/</STARTING_FOLDER>` highlighted and enclosed in a red box. The commit message on the right is set to "Update .abapgit.xml". The "Commit changes" button at the bottom is highlighted with a red border.

```
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...

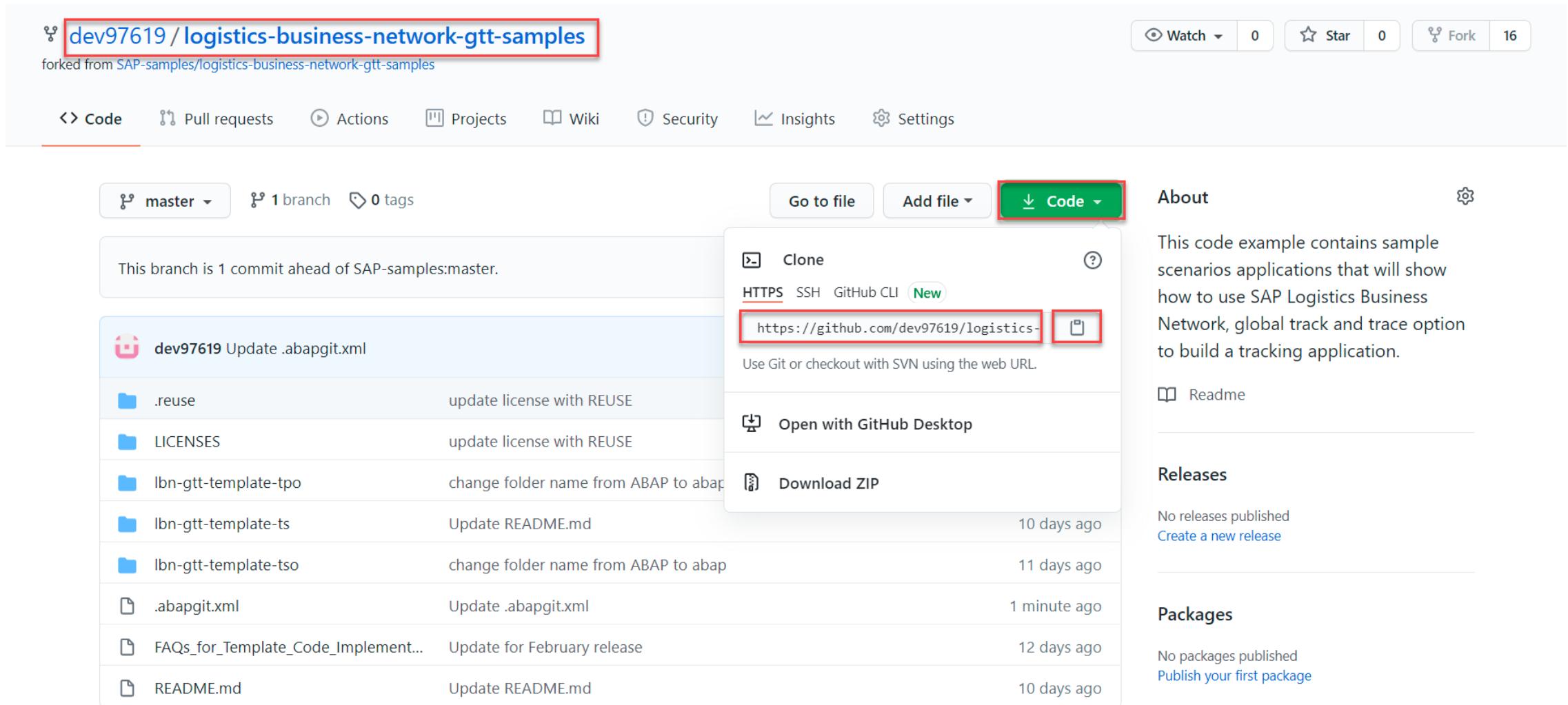
-o- Commit directly to the `master` branch.

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

Commit changes **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. On the right, there's a 'Code' dropdown menu with a 'Clone' option. The 'Clone' section displays the repository URL: <https://github.com/dev97619/logistics-business-network-gtt-samples>. A red box highlights both the URL and the copy icon. 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.

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

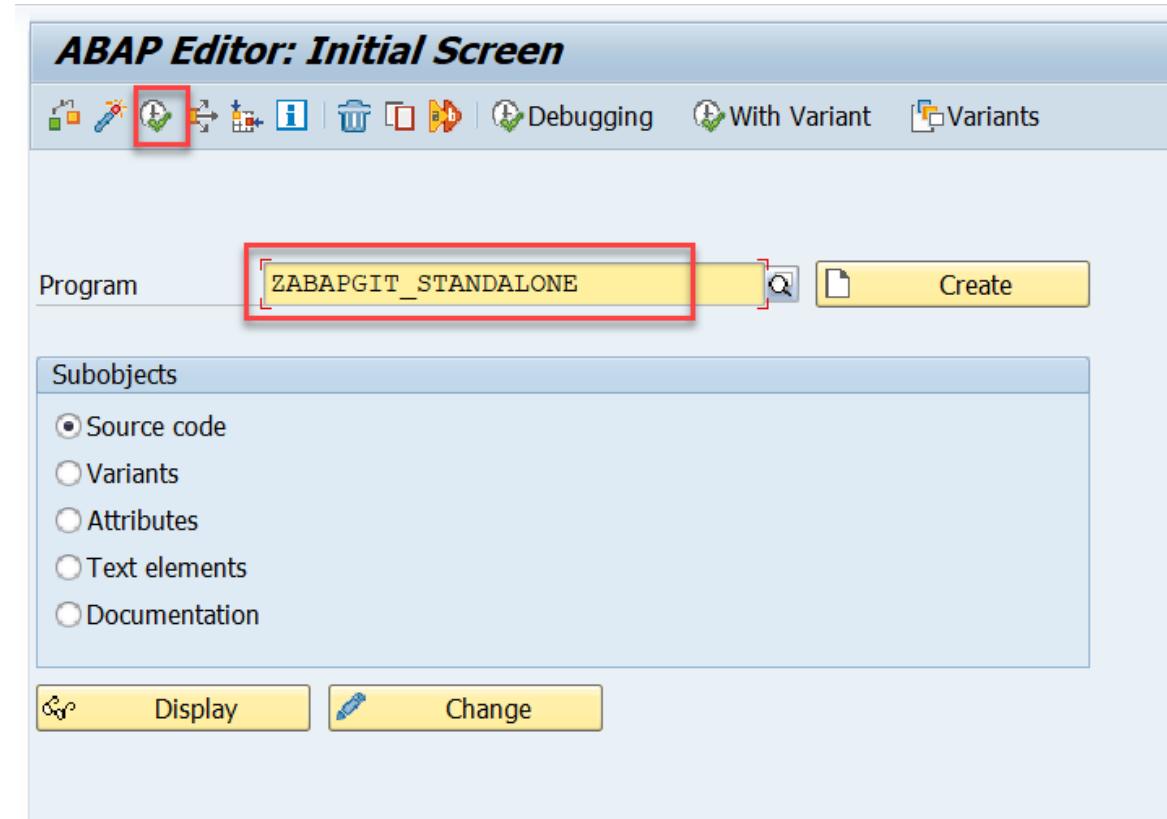
Releases

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:** Title: abapGit ► Repository List. Buttons: New Online, New Offline, Settings, ?.
- Filter:** Filter field, 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 ABAPGit interface for managing repositories. The repository listed is "logistics-business-network-gtt-samples" from GitHub. The "Advanced" menu is open, and the "Remove" option is highlighted with a red box.

ABAP Git

abapGit ► Repository

logistics-business-network-gtt-samples <https://github.com/dev97619/logistics-business-network-gtt-samples.git> 6dd6c5b

Pull Stage Diff Branch Tag Advanced View Refresh

Type Name Path

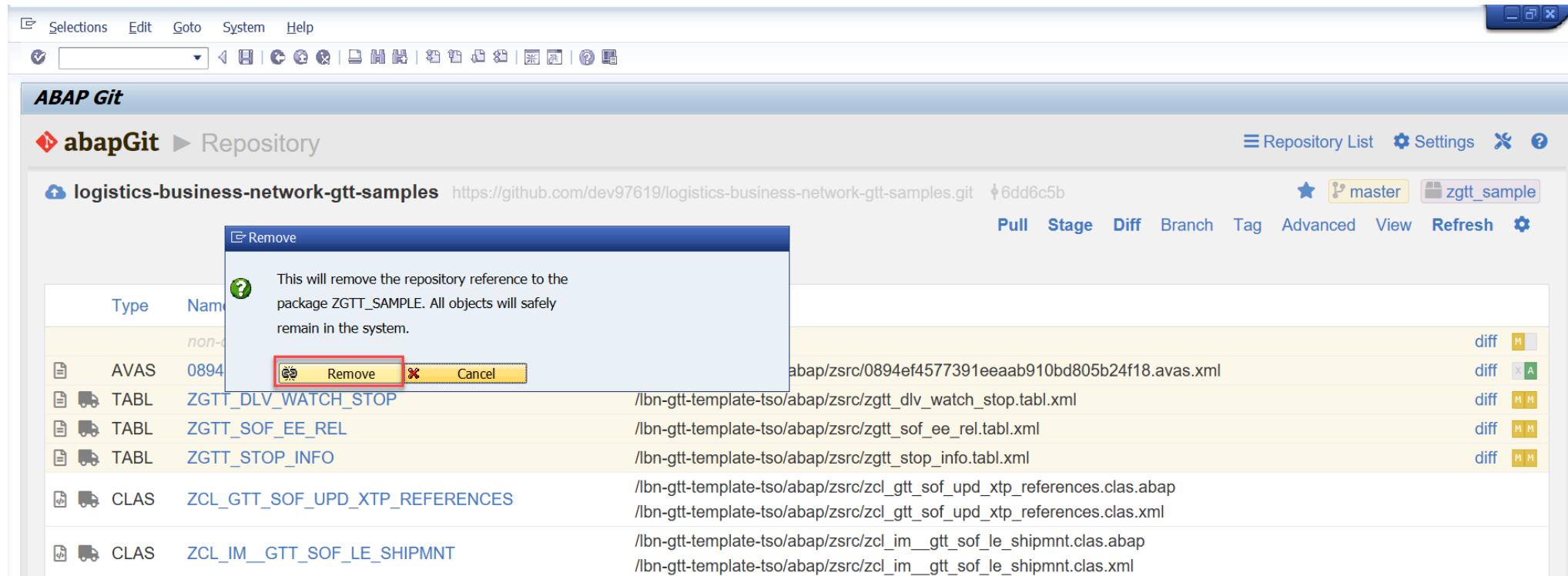
non-code and meta files		
AVAS	0894EF4577391EEAAB910BD805B24F18	./abapgit.xml
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml
TABL	ZGTT_SOF_EE_REL	/lbn-gtt-template-tso/abap/zsrc/zggt_sof_ee_rel.tabl.xml
TABL	ZGTT_STOP_INFO	/lbn-gtt-template-tso/abap/zsrc/zggt_stop_info.tabl.xml
CLAS	ZCL_GTT_SOF_UPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen
CLAS	ZCL_IM_GTT_SOF_LE_SHIPMNT	/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt
TTYP	ZGTT_DLV_WATCH_STOPS	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xr
DTEL	ZGTT_KUNABLAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_kunablaz_txt.dtel.xml
DTEL	ZGTT_LGNUMAZ	/lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml
DTEL	ZGTT_LGORTAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml

Advanced

- 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**
- Uninstall

STEP 3: Remove TSOF Repository in ABAPGit

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

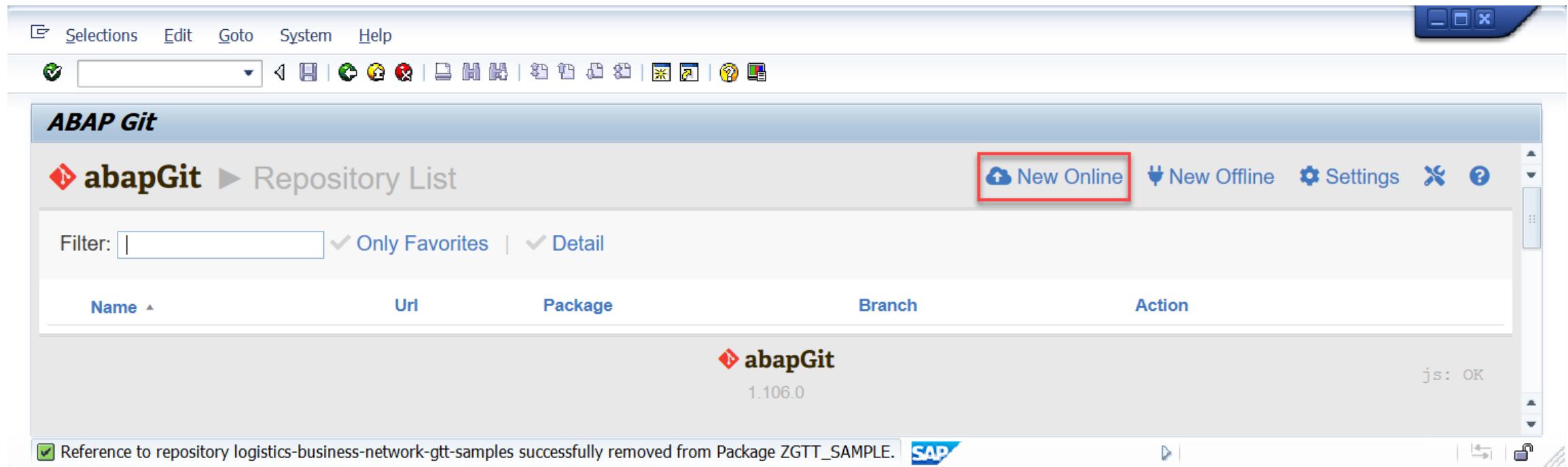


3-6: After repository removal you will see the following message:



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>

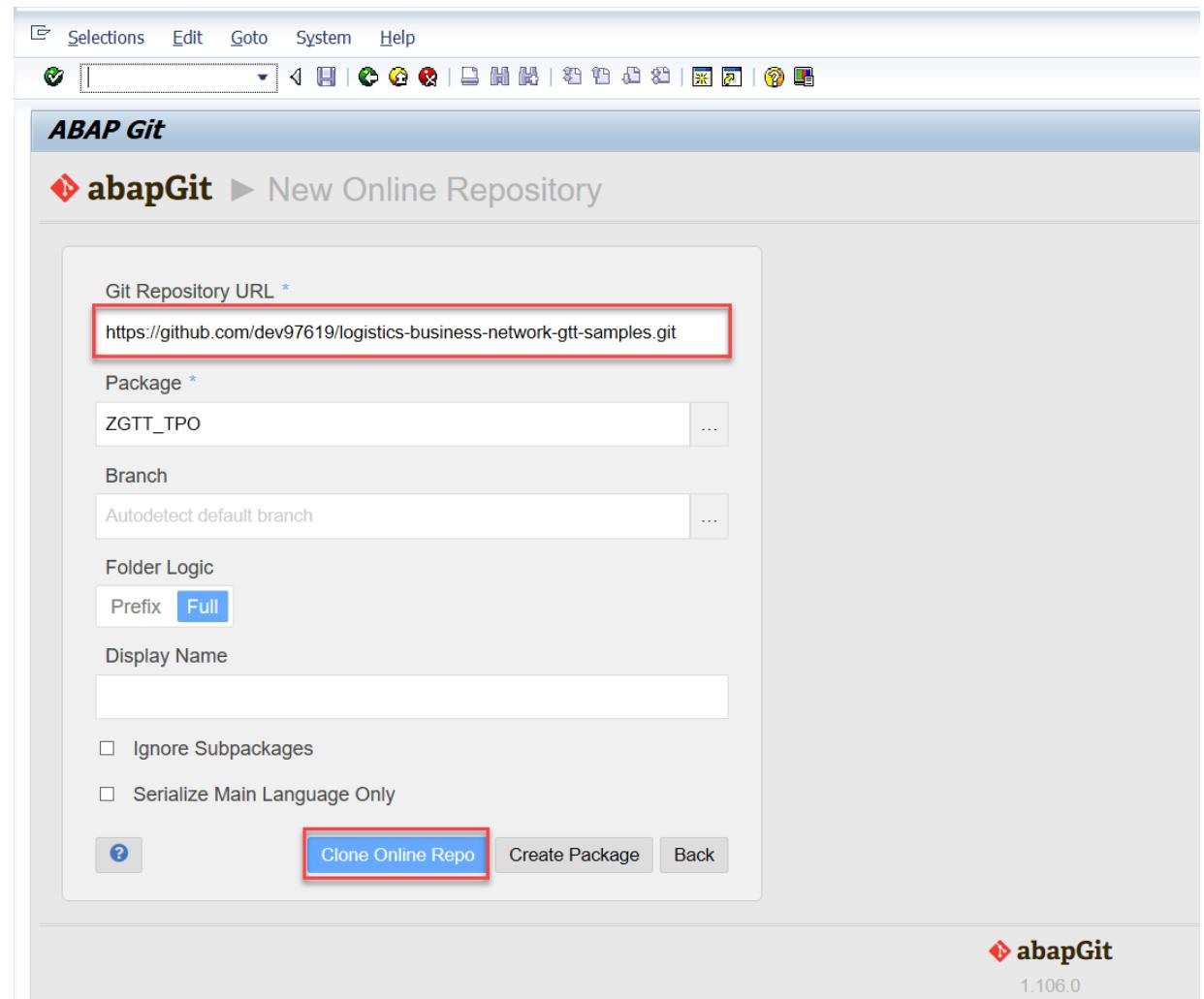
Caution:

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.

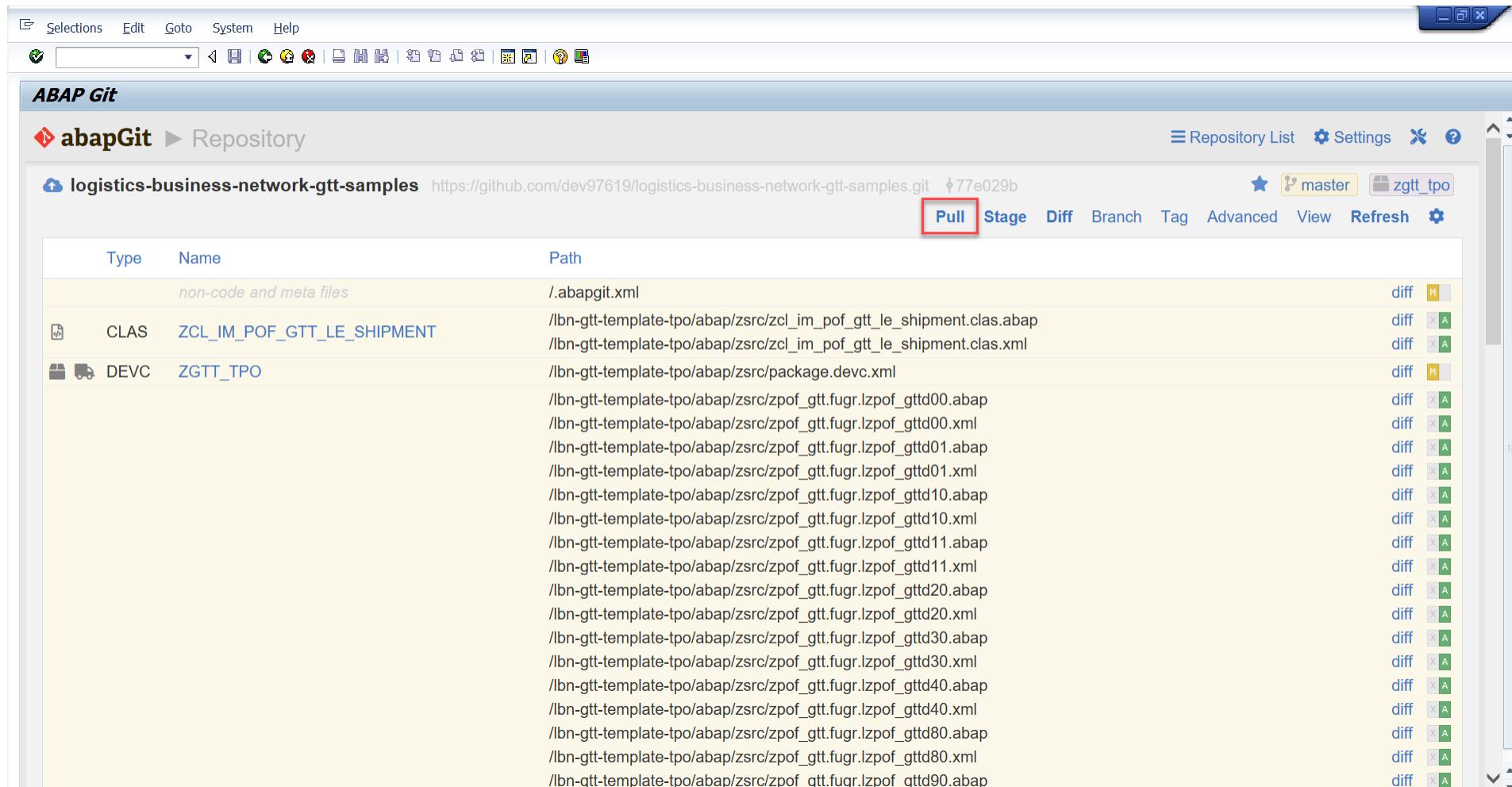
4-4: Set *Full* for **Folder Logic**

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 details: 'logistics-business-network-gtt-samples' with the URL 'https://github.com/dev97619/logistics-business-network-gtt-samples.git' and the commit hash '77e029b'. To the right of the URL are icons for a star, a master branch, and a tag named 'zgtt_tpo'. A navigation bar below the repository details 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 and directories under two main entries: 'non-code and meta files' and 'ZCL_IM_POF_GTT_LE_SHIPMENT'. The 'ZCL_IM_POF_GTT_LE_SHIPMENT' entry contains many sub-files related to ABAP classes and XML files for GTT template processing. On the far right of the table, there are 'diff' buttons for each file, some of which have small colored status indicators (yellow, green, or grey).

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**.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar contains links for "Getting Started", "Setup", "Online Projects", "Offline Projects", and "Reference". The main content area starts with a "Summary" section, followed by a note about the two flavours: "standalone version or developer version". The "Installation" section is highlighted with a red border. It includes a link to "Improve this page", a "Prerequisites" section (noted as SAP BASIS version 702 or higher), and a detailed "Install standalone version" section with four numbered steps:

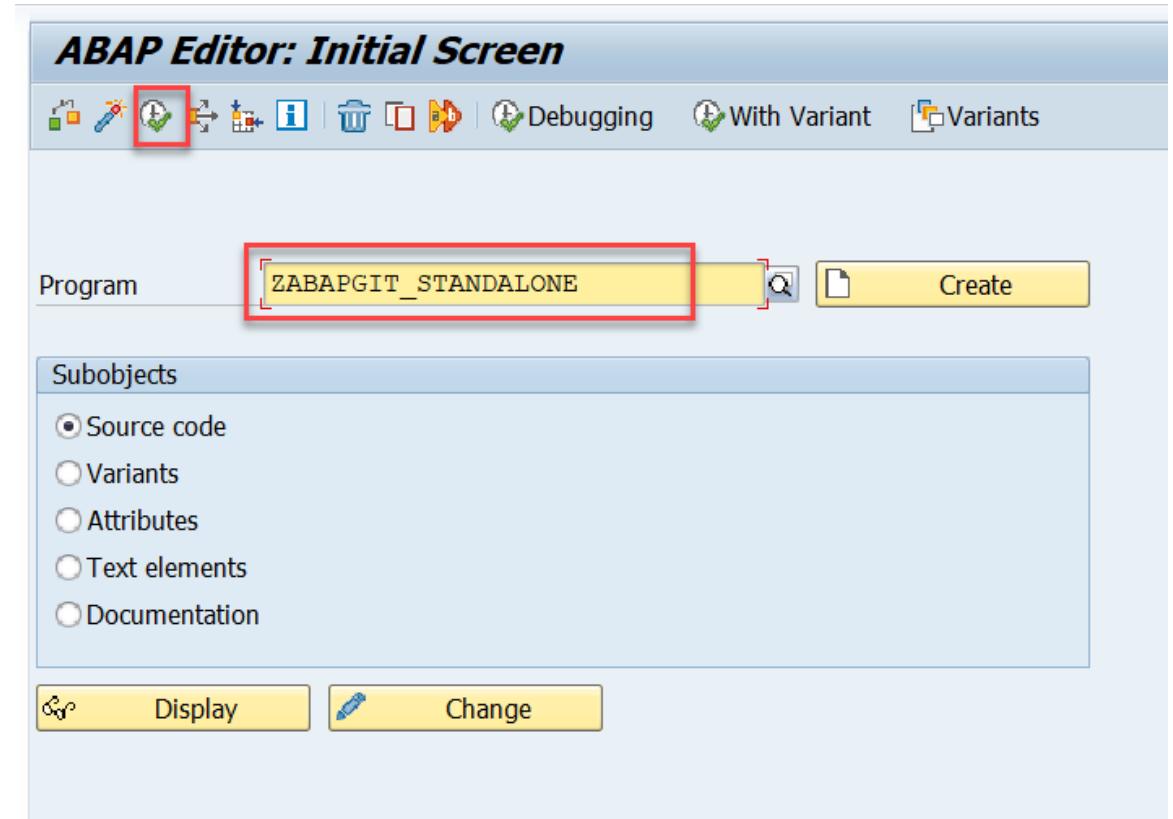
1. Download the ABAP code(right click -> save-as) to a file.
2. Via SE38 or SE80 , create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version.
3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload
4. Activate

Below the steps, it notes: "Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. \$ZABAPGIT)." A final note says: "Now you can use abapGit by executing the report in transaction SE38 ."

STEP 2: Download ABAP Code

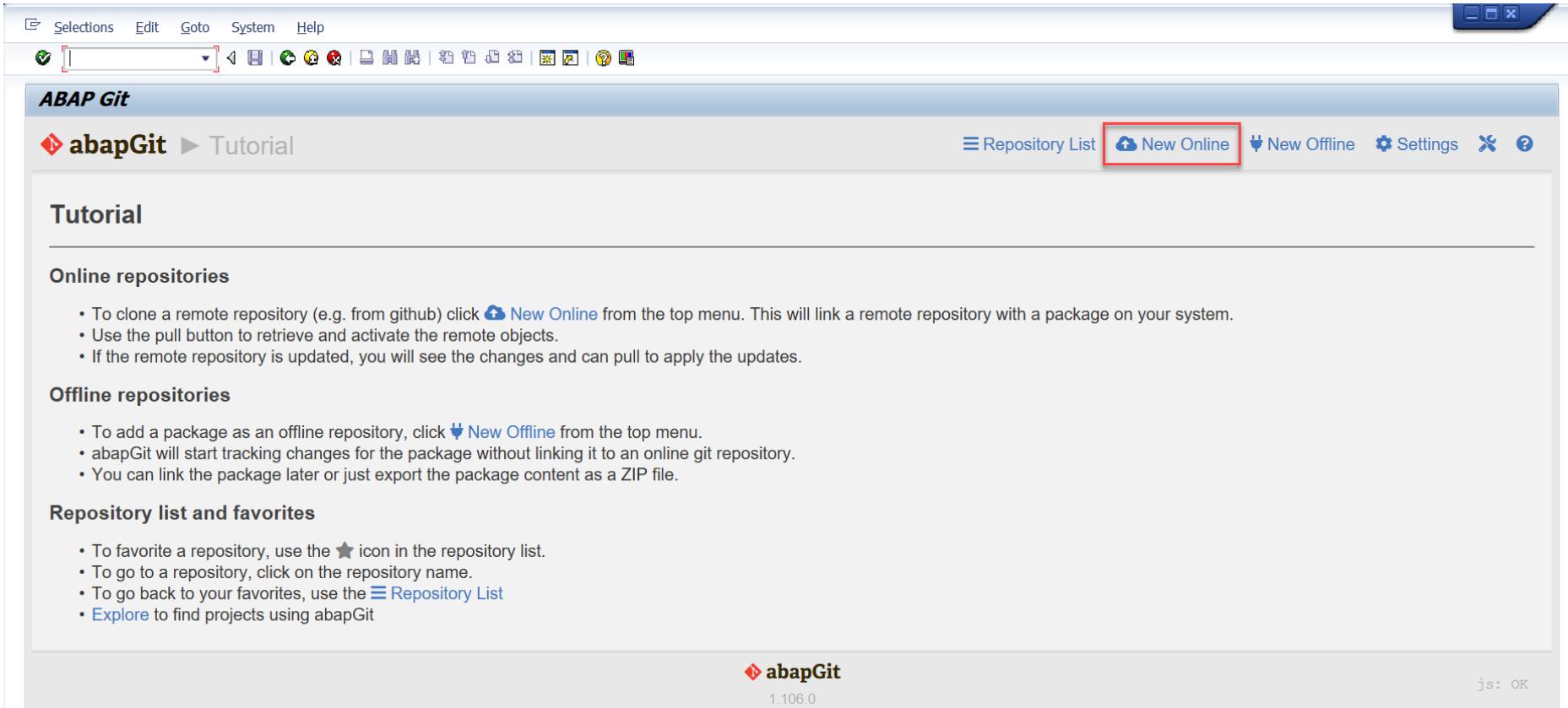
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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 SAP ABAP Git interface. At the top, there's a menu bar with 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the menu is a toolbar with various icons. The main title is 'ABAP Git' and the sub-section is 'abapGit ► Tutorial'. On the right side of the toolbar, 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' and contains sections for 'Online repositories' and 'Offline repositories'. Under 'Online repositories', there's a list of instructions: '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.', and 'If the remote repository is updated, you will see the changes and can pull to apply the updates.' Under 'Offline repositories', there's a list of instructions: '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.', and 'You can link the package later or just export the package content as a ZIP file.' At the bottom of the interface, there's a footer with the 'abapGit' logo and version '1.106.0', and the text 'js: OK'.

STEP 2: Download ABAP Code

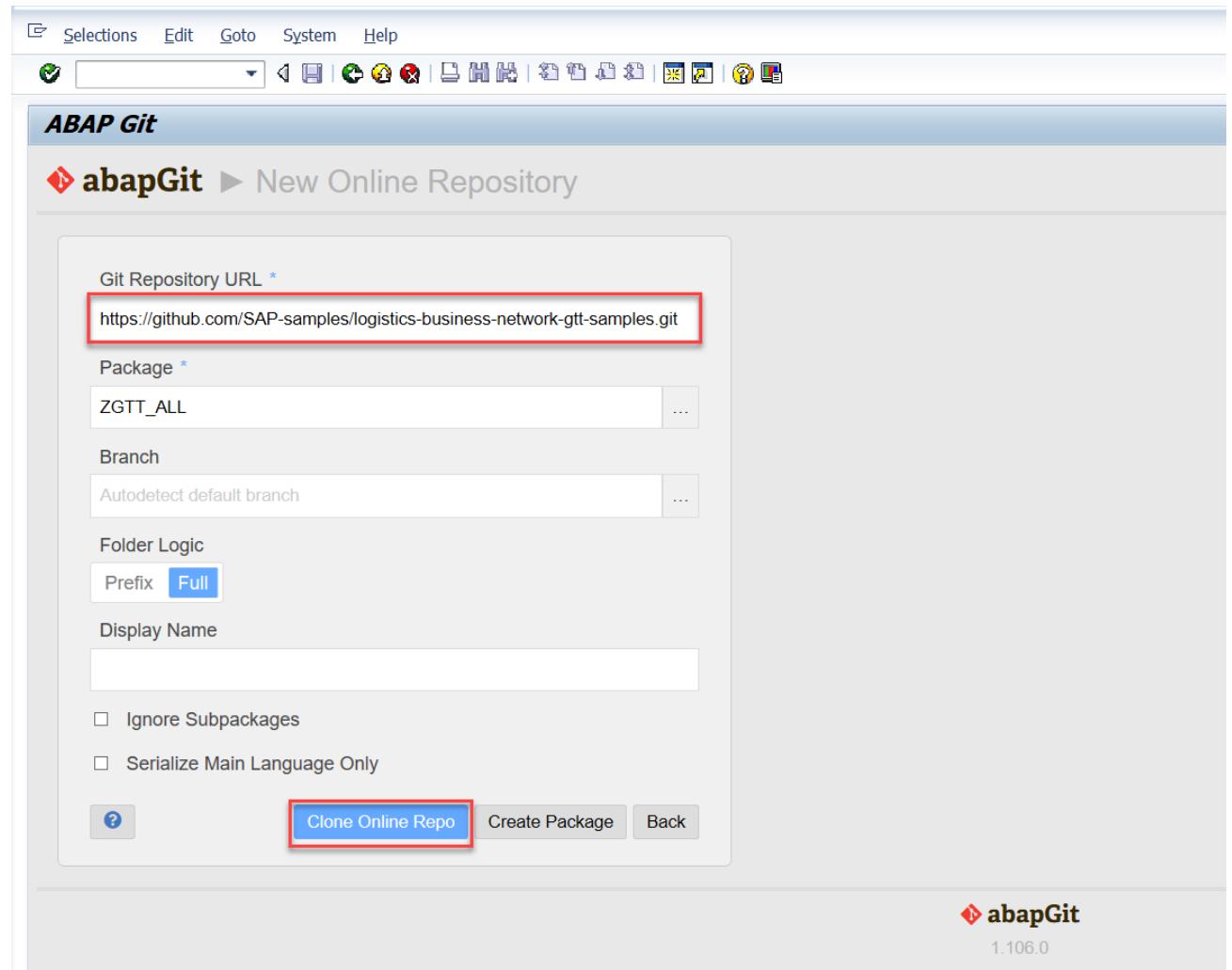
2-4: Fill in the **Git Repository URL**:

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

2-5: Fill in the **Package** where you want to create the new ABAP code. If the package does not exist yet, click **Create package** to create it.

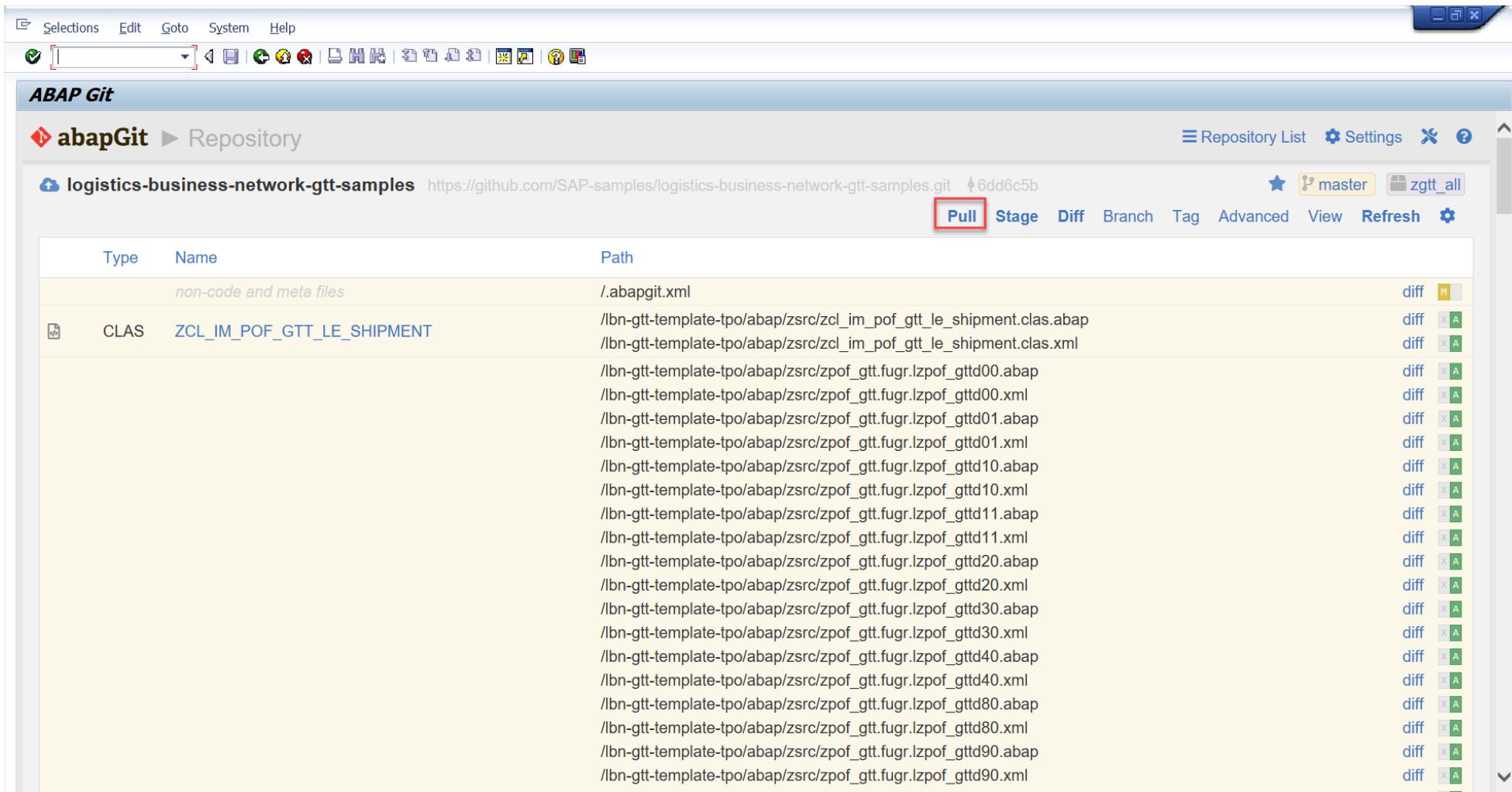
2-6: Set *Full* for **Folder Logic**

2-7: Click **Clone Online Repo** to download the code.



STEP 2: Download ABAP Code

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

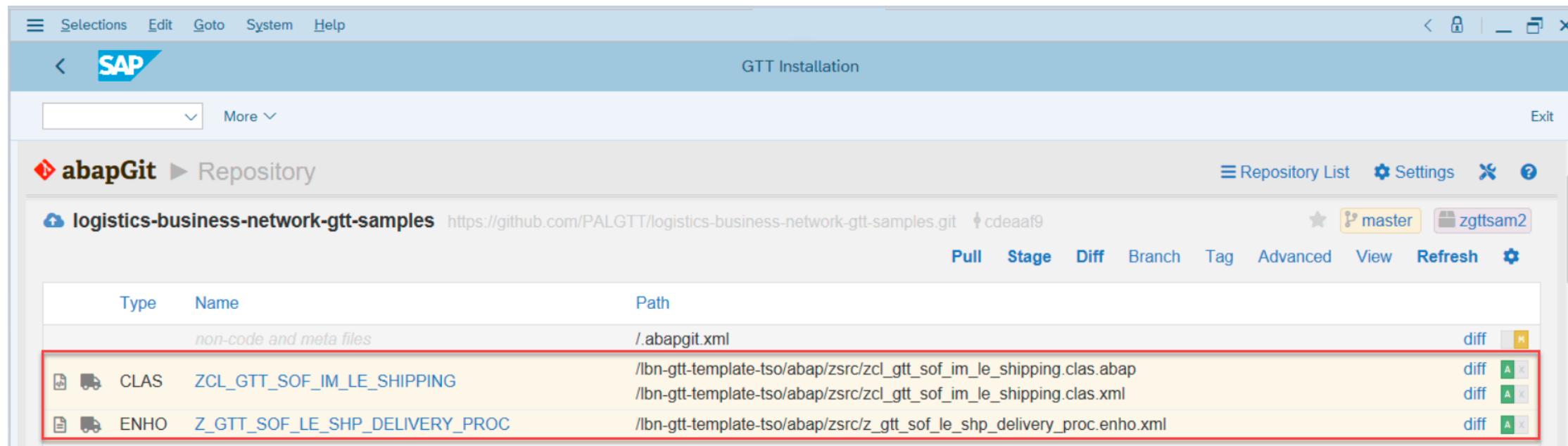


The screenshot shows the ABAP Git interface within an SAP application. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the path 'abapGit > Repository'. A repository card for 'logistics-business-network-gtt-samples' is displayed, including its URL and a commit hash. The 'Pull' button is highlighted with a red box. Below the card is a table listing files and their paths. The table has columns for 'Type', 'Name', and 'Path'. The 'Type' column shows 'non-code and meta files' and '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/'. The 'Path' column shows the full file paths. To the right of the table are 'diff' buttons with status indicators (M, A, or C).

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 with a GitHub repository named 'logistics-business-network-gtt-samples'. The repository URL is <https://github.com/PALGTT/logistics-business-network-gtt-samples.git>. The commit hash is cdeaaaf9. The master branch is selected. A red box highlights the 'diff' column for two specific files: 'ZCL_GTT_SOFTWARE_SHIPPING' and 'Z_GTT_SOFTWARE_SHIPPING_DELIVERY_PROC'. Both rows show a 'diff' icon with a green 'A' and a red 'X'.

Type	Name	Path	diff
	non-code and meta files	/abapgit.xml	[diff icon]
CLAS	ZCL_GTT_SOFTWARE_SHIPPING	/Ibn-gtt-template-tso/abap/zsrc/zcl_gtt_software_im_le_shipping.clas.abap /Ibn-gtt-template-tso/abap/zsrc/zcl_gtt_software_im_le_shipping.clas.xml	[diff icon] [diff icon]
ENHO	Z_GTT_SOFTWARE_SHIPPING_DELIVERY_PROC	/Ibn-gtt-template-tso/abap/zsrc/z_gtt_software_im_le_shipping_delivery_proc.enho.xml	[diff icon]

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 tabs for "Check", "Delete implementation", "Copy implementation", "Rename implementation", "Application help", and "More". Below the tabs, there's a section for "Edit Implementation" with a radio button for "New BAdI" selected. The "Enhancement implementation" field contains "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 tabs for "Properties", "History", "Technical Details", and "Implementation Elements". The "Implementation Elements" tab is active. It shows a table with one row for "Z_GTT_SOF_IM_LE_SHIPPING". The "Implementation" column shows "Implementation Class" and "Implementing Class". On the right side of the bottom window, there are sections for "BAdI Implementation" (set to "Z_GTT_SOF_IM_LE_SHIPPING"), "Description" (set to "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list"), and "Runtime Behavior". Under "Runtime Behavior", the checkbox "Implementation is active" is unchecked, and the note "The implementation will not be called" is displayed. A red box highlights this unchecked checkbox.

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 GTT Version 2.

The image shows two screenshots of the SAP GTT (Global Trace and Trace) interface. The left screenshot displays the 'Define Application Object Types' screen under 'Define Used Business Process'. It shows a business process type 'ESC_SORDER' and an application object type 'ZGTT_SO_INT_HD' (highlighted with a red box). The right screenshot shows the 'Manage Models' application for the 'sof' namespace, specifically the 'IDOC Integration' tab. It lists a tracked process 'SalesOrder' and its mapping to an ERP object type 'Others'. The application object type 'ZGTT_SO_INT_HD' is also highlighted with a red box in this section.

Left Screenshot: Define Application Object Types - Details

- Bus. Proc. Type: ESC_SORDER
- Appl. Obj. Type: ZGTT_SO_INT_HD (highlighted)
- Text: Sales Order Header

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)
- Tracked Process / Events (2):
 - SalesOrderEvent: E1EHPAO
 - Completion: 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 GTT Version 2.

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

The image displays two screenshots illustrating the configuration of Tracking ID Types.

SAP AOT Screenshot: The 'Display View "Define Application Object Types": Details' screen shows the 'Bus. Proc. Type' as 'ESC_SORDER', 'Appl. Obj. Type' as 'ZGTT_SO_INT_HD', and 'Text' as 'Sales Order Header'. Under 'Parameter Setup', the 'TrkID Method' is set to 'D Determine from Field', 'Tr.ID Tab. Type' is '1 Main Object Table', and 'Tr. ID Code Set' is highlighted with a red box and contains the value 'SALES_ORDER'. The 'Tracking ID Fld' is 'VBELN'.

GTT Version 2 Screenshot: The 'Model Details' screen for 'SOF' (Active) shows the 'Tracked Process' tab selected. In the 'Items (6)' list, the 'SalesOrder' item is selected, and its 'Tracking Id Type' is highlighted with a red box and set to 'SALES_ORDER'. In the 'User Model Fields (16)' list, the 'SalesOrder' field is selected, and its 'Tracking Id Type' is also highlighted with a red box and set to 'SALES_ORDER'. A modal dialog titled 'Edit Tracked Process' is open, showing the 'Name' field as 'SalesOrder' and the 'Tracking Id Type' field as 'SALES_ORDER', both 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:

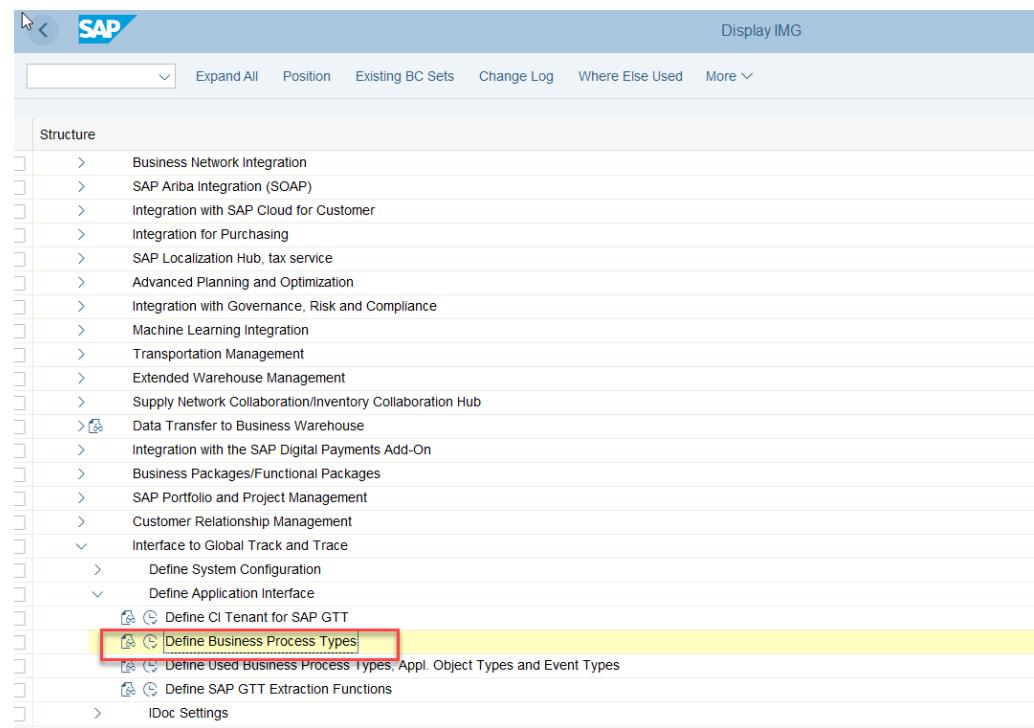
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_HD	Function for setup of control parameters of delivery header
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_ITM	Function for setup of control parameters of delivery item
Control 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 Check In 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

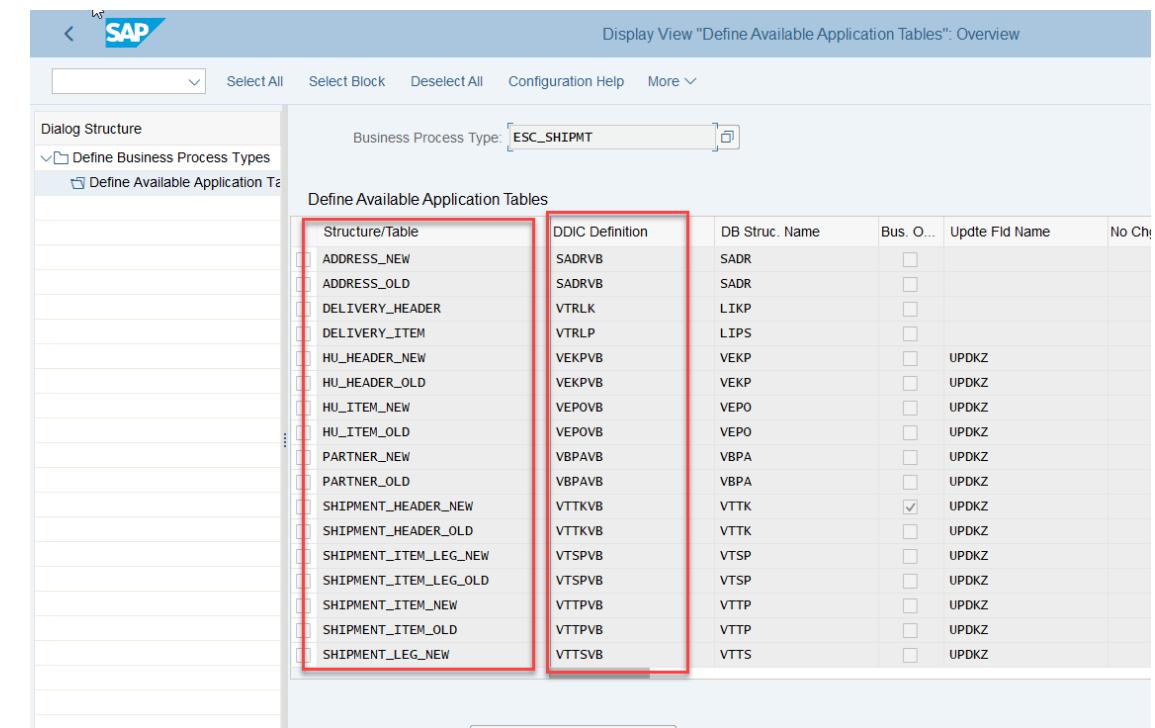
5-1: In Display IMG page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

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.



The screenshot shows the SAP Display IMG interface. The top navigation bar includes 'Expand All', 'Position', 'Existing BC Sets', 'Change Log', 'Where Else Used', and 'More'. The main area is titled 'Structure' and lists various integration components. Under 'Interface to Global Track and Trace', the 'Define Application Interface' option is expanded, revealing the 'Define Business Process Types' link, which is highlighted with a red box.



The screenshot shows the SAP Display View "Define Available Application Tables" overview. The top navigation bar includes 'Select All', 'Select Block', 'Deselect All', 'Configuration Help', and 'More'. The 'Business Process Type' field is set to 'ESC_SHIPMT'. The main area displays a table with two columns: 'Structure/Table' and 'DDIC Definition'. Both columns for the selected business process type 'ESC_SHIPMT' are highlighted with a red box. The table includes rows for various application tables like ADDRESS_NEW, ADDRESS_OLD, DELIVERY_HEADER, etc., and their corresponding DDIC definitions.

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	LIKP			
DELIVERY_ITEM	VTRLP	LIPS			
HU_HEADER_NEW	VEKPV	VEKP			UPDKZ
HU_HEADER_OLD	VEKPV	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 menu bar includes "Previous Object", "Display <> Change", "Other Object...", "Enhance", "Check", "Activate", "Test/Execute", "Where-Used List", "Display Object List", and "More". Below the menu is a toolbar with search, filter, and exit buttons. The main area displays the ABAP code for the function module ZGTT_SOF_OTE_SHP_HDR_REL. The code is annotated with red boxes highlighting specific sections:

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

The code performs several tasks: it checks if the main table is a Shipment (VTTK), creates a logitable reference, reads the main object table, and then checks the relevance of the AOT (Activity Object Type) in the OTE (Order Type Element). It also handles the case where the shipment is newly created by checking the delivery assignment.

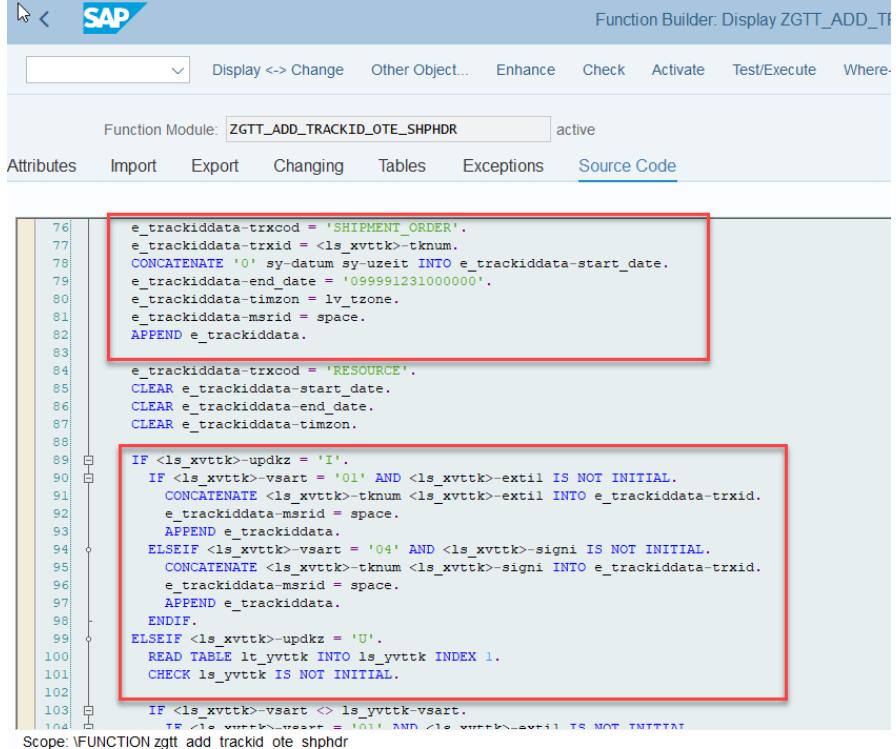
7: Coding Tips in the Tracking ID function modules

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. GTT v2 accepts delta transport for tracking IDs, which means that only the newly-created / changed / deleted tracking IDs shall be filled, while the ones without change need to be ignored in the logic.
5. The tracking ID for its own process type needs to be filled for each process update.
6. In case of tracking ID deletion, the field *ACTION* shall be filled with 'D'.

See sample code of function:

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". The "Source Code" tab is selected. The code is written in ABAP and handles the creation and updating of tracking IDs based on the value of the *updckz* field in the input table *ls_xvttk*. It uses concatenation and APPEND statements to build the *e_trackiddata* structure, and IF-ELSEIF blocks to handle different update scenarios. A red box highlights the main processing loop from line 76 to line 102.

```
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>-updckz = '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>-updckz = '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. GTT v2 asks for full transport for all the control parameters, which means that all the fields needs to be extracted in all cases, no matter whether their values have been changed.
4. To fill in the composition (table) fields defined in Manage Model app, use the parameter field *PARAMINDEX* to specify the line number. If the field is empty, GTT regards it as a simple flat field.
5. **To clear a composition, fill the key field using invalid values, for which key attribute has been checked in Manage Model app. It's not recommended to fill a code list type field to clear a composition even if it's a key field.**
6. The field with fixed name 'ACTUAL_BUSINESS_DATETIME' and 'ACTUAL_BUSINESS_TIMEZONE' are mandatory fields to be transported for event handling sequencing in GTT Version 2.
7. The fields with fixed names 'ACTUAL_TECHNICAL_TIMEZONE' and 'ACTUAL_TECHNICAL_DATETIME' are optional and recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in GTT via TP request, which leads to an error)
8. In Manage Models app, click tab *IDOC Integration* to map the parameter names and model field names.
9. For DATE or DATETIME fields, when the source value is initial like '00000000' '0000000000000000', then please ensure to only enable *PARAMNAME* and *PARAMINDEX* in the extractor code, not enable *VALUE* for IDOC sending.
10. For amount field which has reference currency, ensure to call BAPI 'BAPI_CURRENCY_CONV_TO_EXTERNAL' using the reference currency to make the amount tracked correctly by GTT Version 2. The BAPI will output the conversion result in 4 decimals as fixed, which needs additional rounding in the extractor if the corresponding field defined in the tracking model is less than 4 decimals.
11. In the shipment extractor, add the prefix LBN# into the fields SERVICE AGENT LBN ID for integration with Visibility Providers.

See sample code of function: *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 'Sales Order Fulfillment' description is present. On the right, there are 'Edit' and 'Draft View' buttons.

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'.

Below the navigation, there are two input fields: 'Tracked Process:' containing 'Shipment' and 'Integration Switch:' with a blue 'ON' button.

A section titled 'Tracked Process Mapping' shows 'ERP Object Type: Others' and 'Application Object Type: ZGTT_SHP_INT_HD'.

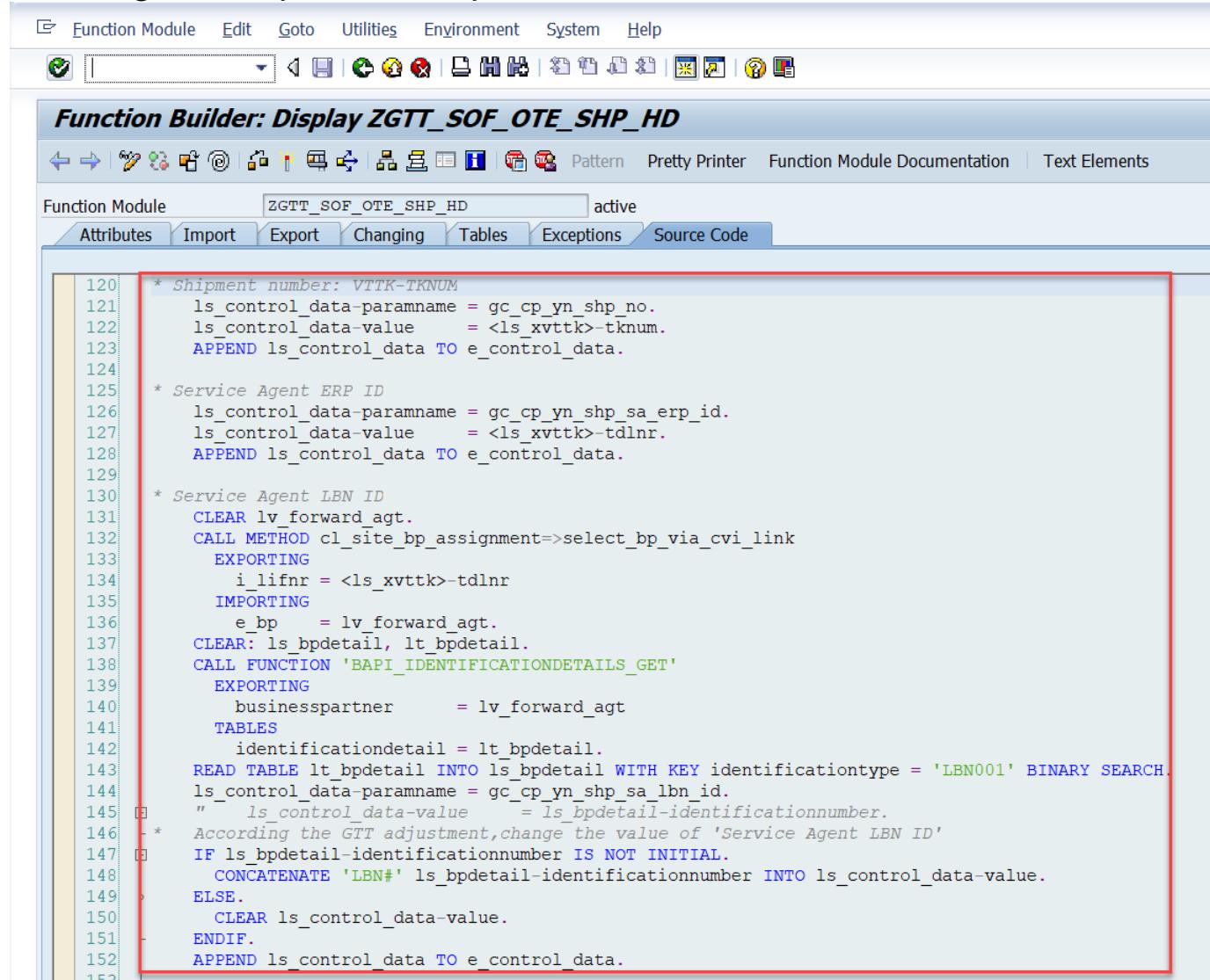
The 'Tracked Process / Events (26)' table lists various tracked processes and their corresponding IDOC segments and event codes. Some rows are collapsed under sections like 'Tracked Process' and 'Event Types'.

The 'User Model Fields' table, which maps application object fields to IDOC segments and fields, is highlighted with a red border. It contains the following data:

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPBP	YN_SHP_NO
serviceAgentLbnId	E1EHPBP	YN_SHP_SA_LBN_ID
transportationMode	E1EHPBP	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPBP	YN_SHP_CONTAIN_DGOODS
forwardingAgentTrackingId	E1EHPBP	YN_SHP_FA_TRACKING_ID
> stops		
shippingType	E1EHPBP	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 name "ZGTT_SOF_OTE_SHP_HD" is selected in the top bar. The "Source Code" tab is active. The code is displayed in a text editor with line numbers on the left. A red box highlights the main logic starting from line 120.

```
120 * Shipment number: VTTK-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, GTT 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 GTT 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 GTT Version 2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function: *ZGTT_SOF_EE_SHP_HD*

Name	IDOC	Event Code
Tracked Process		
ShipmentEvent	E1EHPAO	
Event Types		
LoadingStart	E1EVMHDR02	LOAD_BEGIN
POD	E1EVMHDR02	POD
Departure	E1EVMHDR02	DEPARTURE
Arrival	E1EVMHDR02	ARRIV_DEST
LoadingEnd	E1EVMHDR02	LOAD_END
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 GTT via TP request, which leads to an error)
5. If the event has reference table information, fill in the table *CT_TRACKREFERENCES*.
6. The field *CT_TRACKINGHEADER-SRCCOD*, *SRCID*, *SRCTX* is used for event reason transport.
7. In *Manage Models* app, click tab *IDOC Integration* to map the user-defined parameter names and model field names.

See sample code of function: *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 'sof' model, which is active. The 'IDOC Integration' tab is selected. The interface includes sections for Tracked Process Mapping, Tracked Process / Events (4), and User Model Fields.

Tracked Process Mapping:

- Tracked Process: DeliveryItem
- Integration Switch: ON

Tracked Process / Events (4):

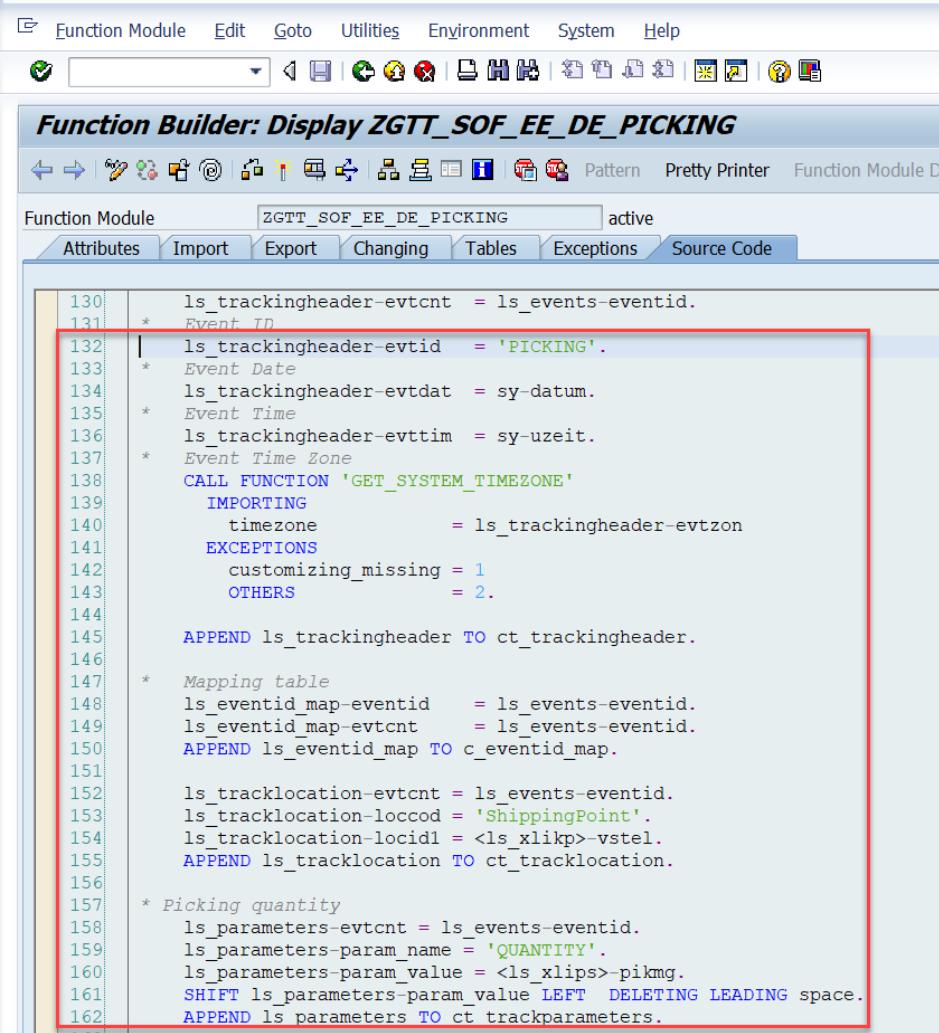
Name	IDOC	Event Code
Tracked Process		
DeliveryItemEvent	E1EHPAO	
Event Types		
Picking	E1EVMPAR	PICKING
Packing	E1EVMPAR	PACKING
DeliveryItemPOD	E1EVMPAR	DLV POD

User Model Fields:

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 "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 GTT, the preceding delivery and item process, and their planned events needs to be updated and transported to GTT.

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 implementation *Z_GTT_SOF_LE_SHIPMNT*. The top navigation bar includes links for Next Object, Display <-> Change, Other Object, Check, Display object list, Display navigation window, Application help, and Definition Document. The main area displays the following fields:

Implementation Name:	<input type="text" value="Z_GTT_SOF_LE_SHIPMNT"/>	<input checked="" type="checkbox"/> Active
Implementation Short Text:	GTT - Enhancement to update the impacted delivery orders	
Definition Name:	<input type="text" value="BADI_LE_SHIPMENT"/>	
Runtime Behavior:	<input type="text" value="Implementation will be called"/>	

Below these fields, under the **Interface** tab, are the following entries:

Interface Name:	<input type="text" value="IF_EX_BADI_LE_SHIPMENT"/>
Name of Implementing Class:	<input type="text" value="ZCL_IM_GTT_SOF_LE_SHIPMNT"/>

A table lists the methods and their implementation types:

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

At the bottom, there is a field for the Default Implementation Class:

Default Implementation Class:	<input type="text"/>
-------------------------------	----------------------

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 GTT prior to the order IDOC, it will lead to processing failure.

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

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

13: Solution of IDOC Sequencing Issue

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

2. Create CI tenant.

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

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 screenshots of SAP GUI screens. The top screenshot shows the 'SAP Global Track & Trace Definitions' table with one entry: CI for Global Track & Trace (ZGTTSOFIN2), CI Log. System (ZLSGTTINT), SAP Track & Trace Version (GTT2.0 Logistics Business N...), and Description (CI For GTT V2 Integration system Sales Order Sample APP). The bottom screenshot shows the 'RFC Destination ZGTTV2_SOF_INT2' configuration screen. It includes fields for RFC Destination (ZGTTV2_SOF_INT2), Connection Type (HTTP Connection to External Server), and three Description fields (Description 1: RFC for Tracked Process of Sales Order Sample Application, Description 2: blank, Description 3: blank). At the bottom, there are tabs for Administration, Technical Settings, Logon & Security, and Special Options. The 'Target System Settings' section shows Host and Port (443) fields, and 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



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.