



SAP Business Network Global Track and Trace

Track Sales Orders - Deep Dive with SAP ERP Integration

SAP Business Network for Logistics
March 2021

PUBLIC

Objectives

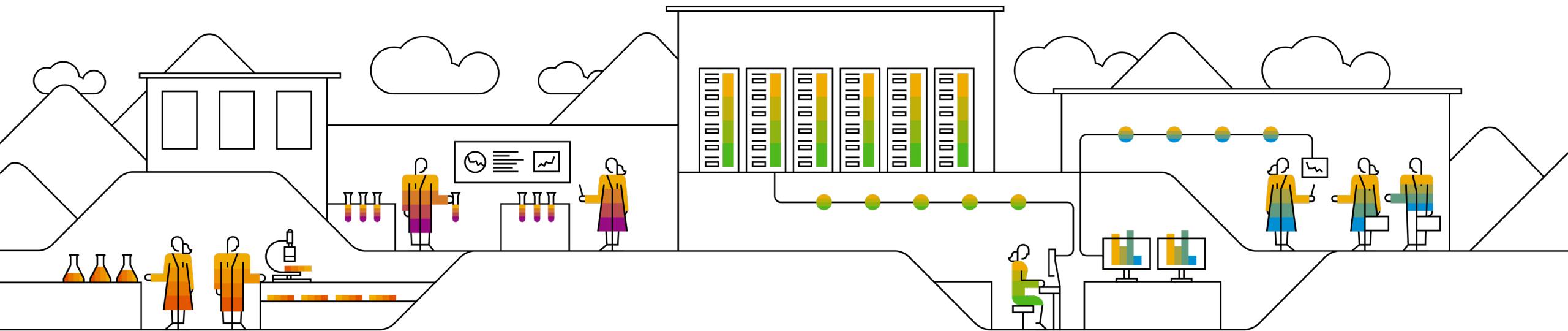


After completing this learning module, you will be able to:

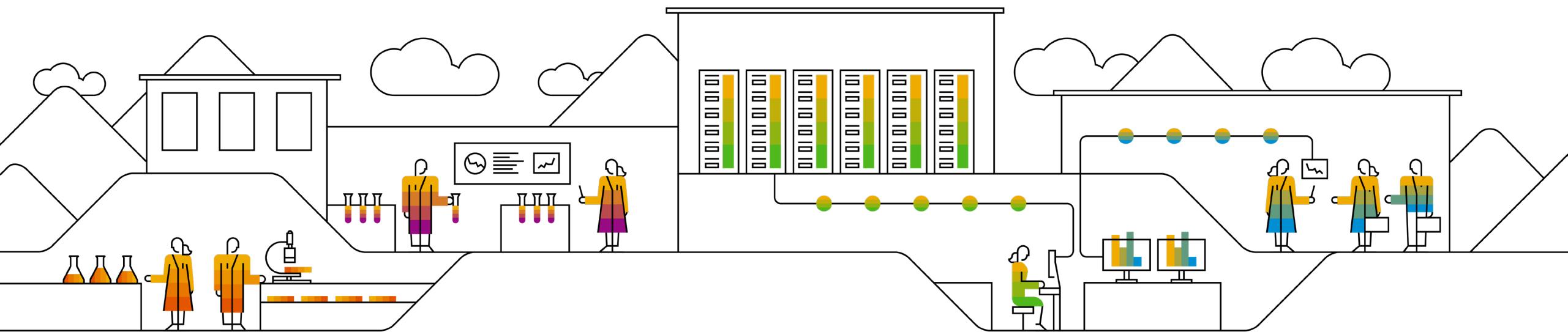
- Learn what prerequisite is necessary for SAP Business Network Global Track and Trace
- Learn how to maintain IDOC configurations in ERP for integration
- Learn how to maintain extractors in ERP for integration
- Learn how to download and implement sample ABAP codes from Github
- Learn how to customize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

Agenda

- A Prerequisites
- B Configuration and Implementation - Basic
 - B1 IDOC Configuration
 - B2 Extractor Configuration
- C Download ABAP Code from GitHub
- D Configuration and Coding Guide - Advanced



A) Prerequisites



STEP 1: Check the SAP Product Version

1-1: Make sure that you have met the requirements for the product version mentioned in the [Prerequisites](#) chapter of How to Send Documents from SAP ERP to SAP Business Network Global Track and Trace. You can find this guide at <http://help.sap.com/gtt>.

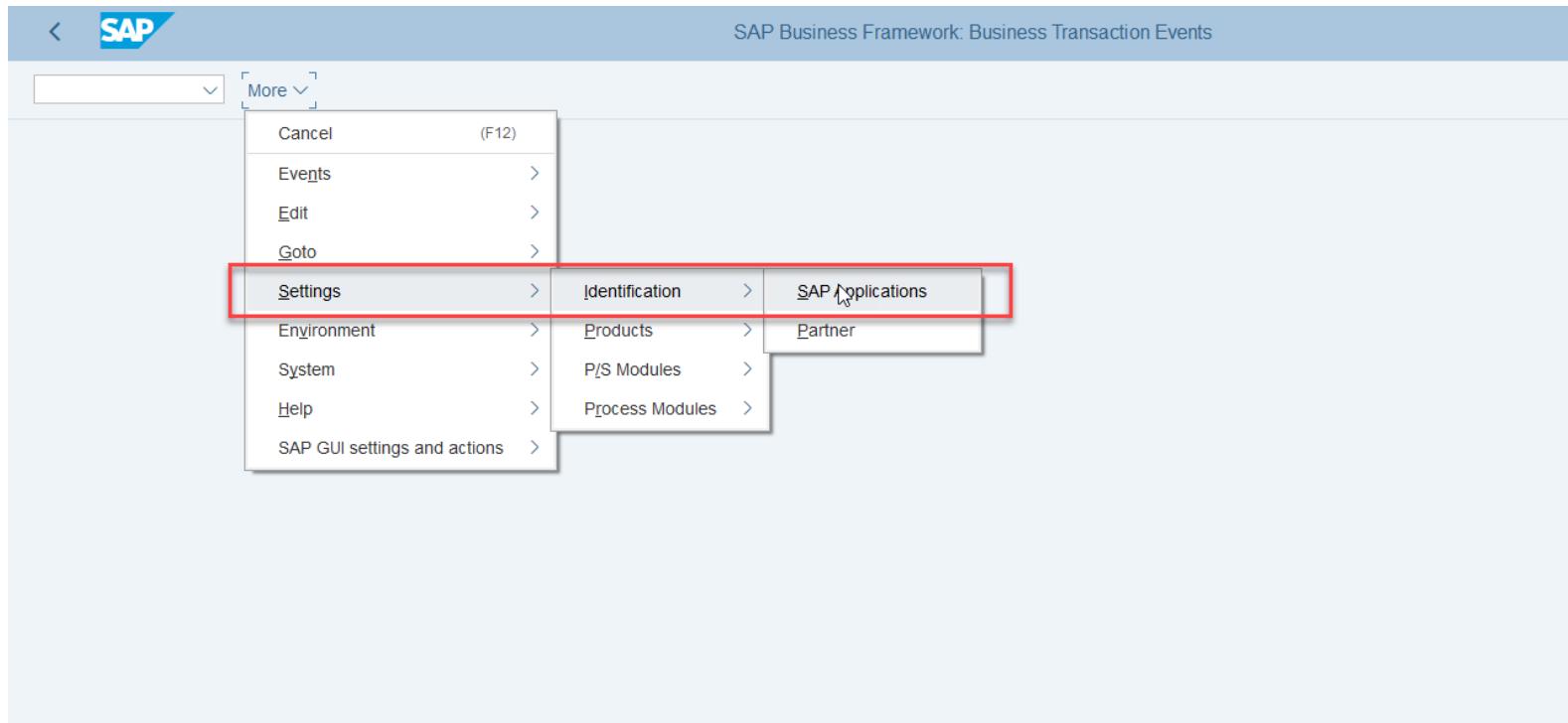
1-2 : The ABAP codes on Github to support sample apps for SAP Business Network Global Track and Trace shall be implemented in SAP S/4HANA 1909 SP03 on premise or higher. Please note that the codes are not validated in its lower version or other ECC series of products, so you might need to do further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

STEP 2: Log on the Development Client to Configure BTE

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

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

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



STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**

The screenshot shows a SAP application window titled "Change View 'BTE Application Indicator'. Overview". The window displays a list of applications and their status. The application "PI-EM" is selected, indicated by a red box around its row. The "Text" column for PI-EM shows "SAP Event Manager Integration" with a checked checkbox. Other applications listed include PM, PM-BW, PM-EQM, PM-PAM, PMA-PC, PMAT, PMIPUR, PMPUSH, PP-BD, PP-DD, PP-MRP, PRICAT, PS-REP, PSRV, QBEXT, QBEXTP, QILPO, RDSVFI, and RDSVMD. The "Display" button is highlighted in blue at the top right of the screen.

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

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



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

1-1: Log on to the business client

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

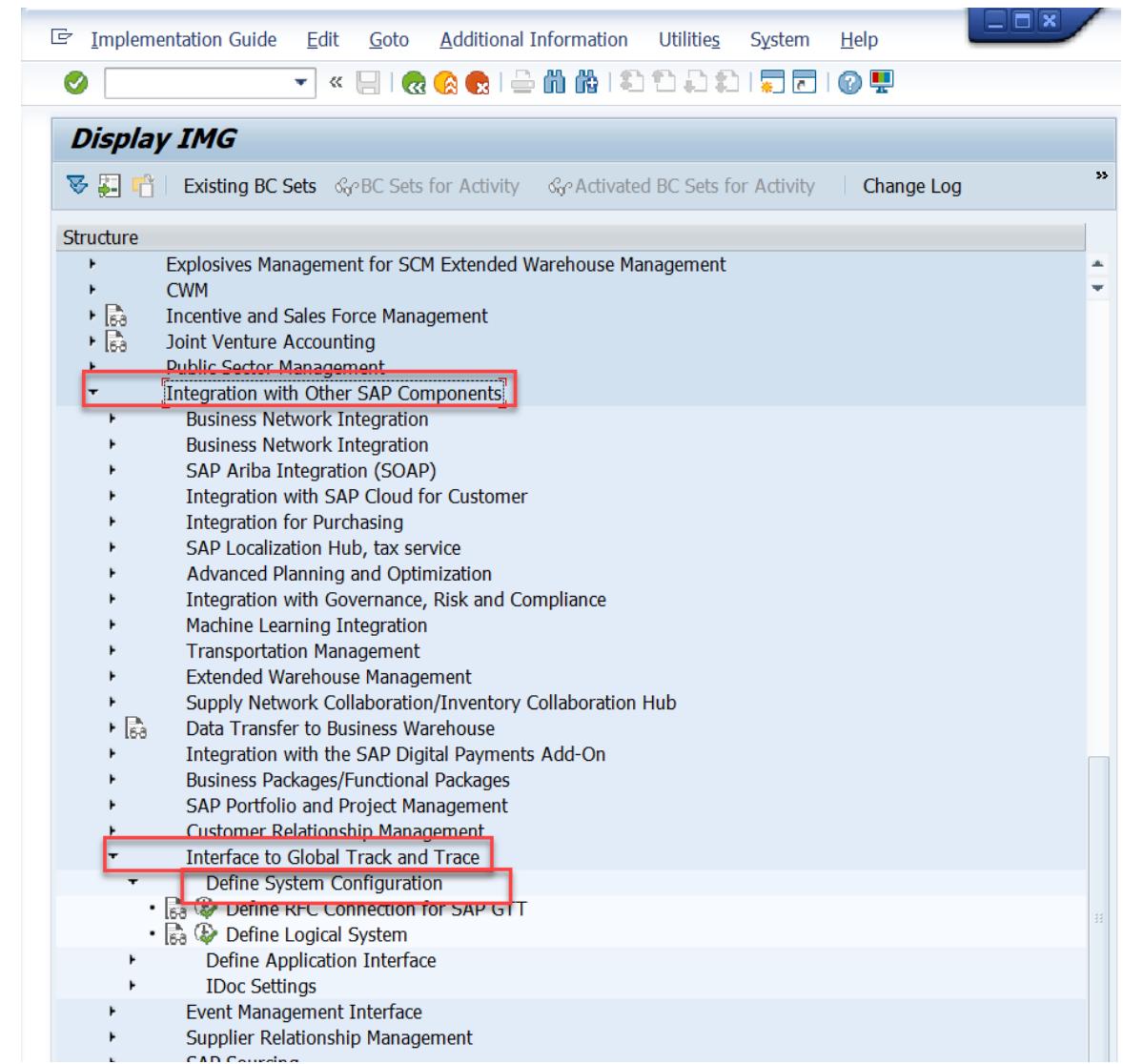
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

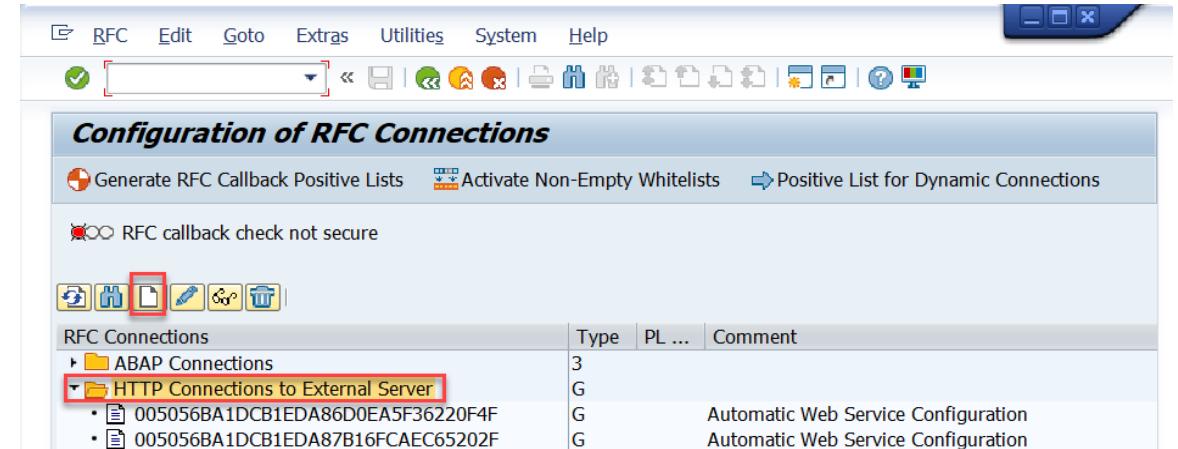
Define RFC Connection for SAP GTT



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

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

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



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

1-7: Enter a description

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

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

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

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

Port: 443

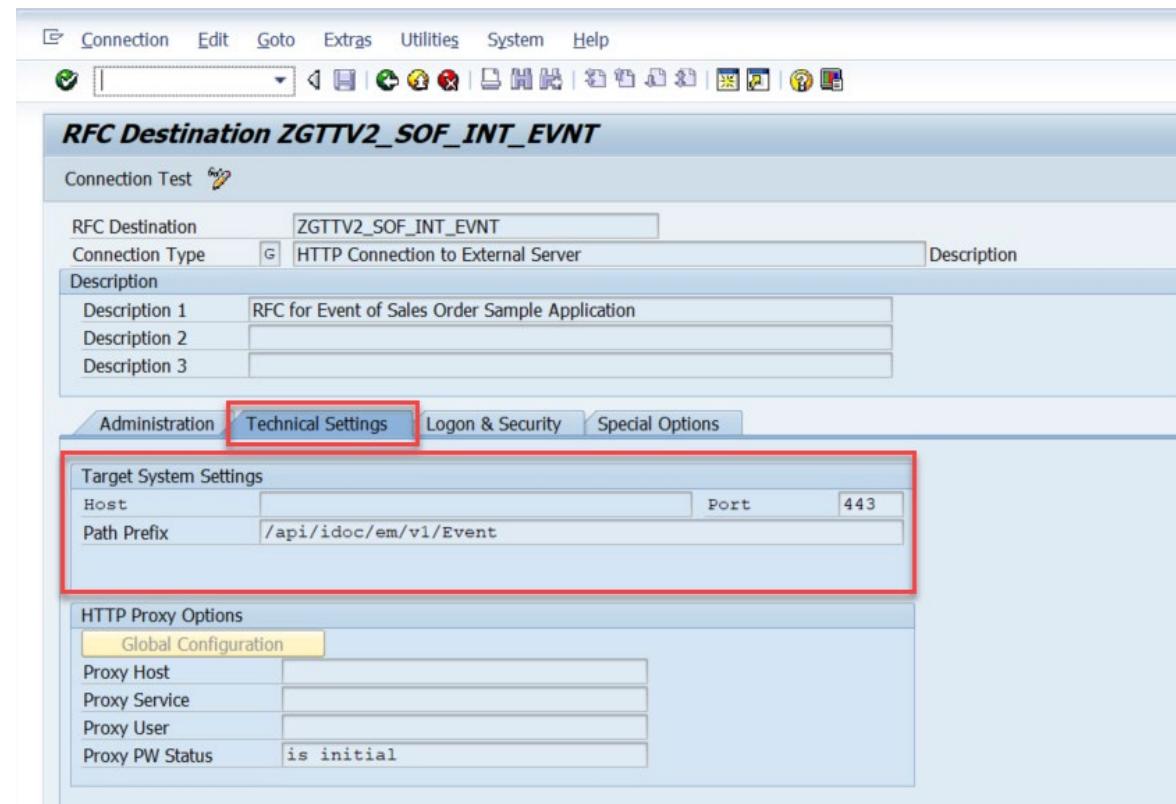
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

For the event:

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

For the tracked Process:

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



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

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

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

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

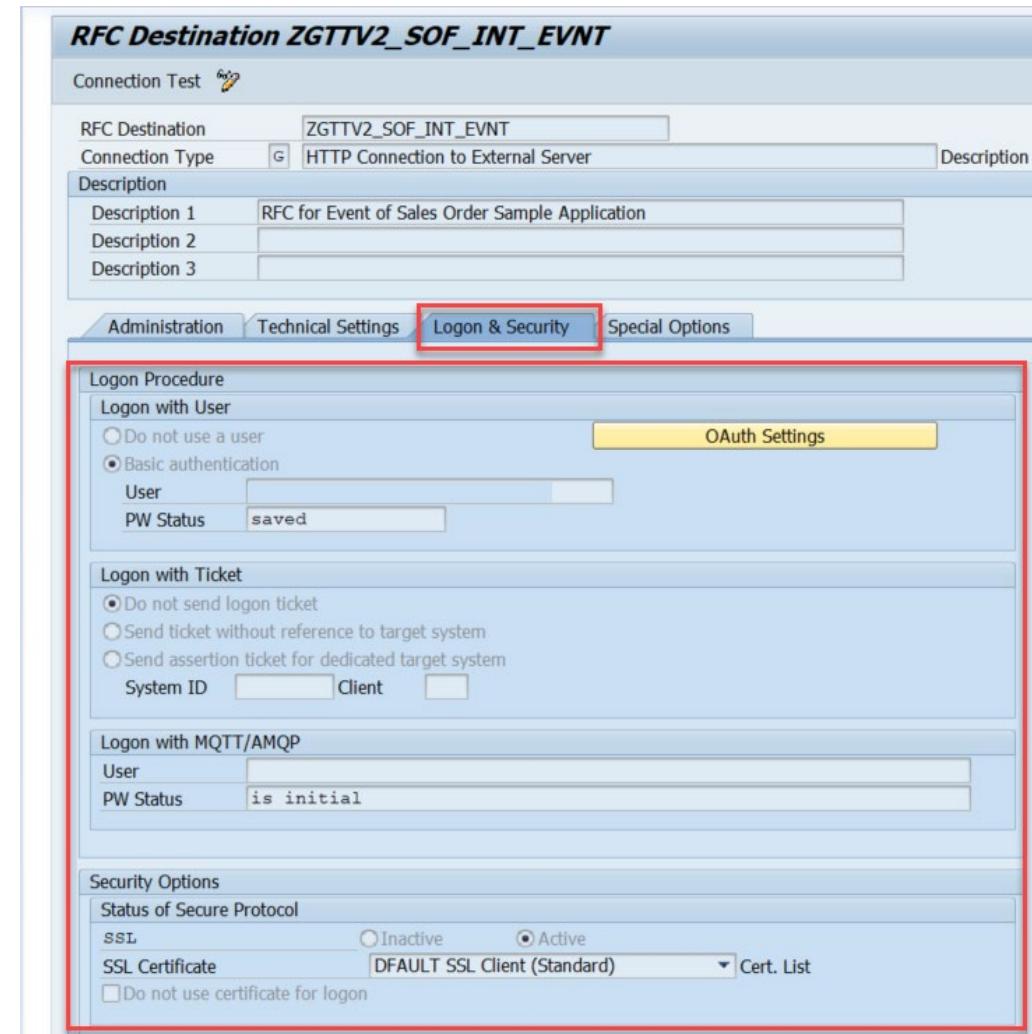
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

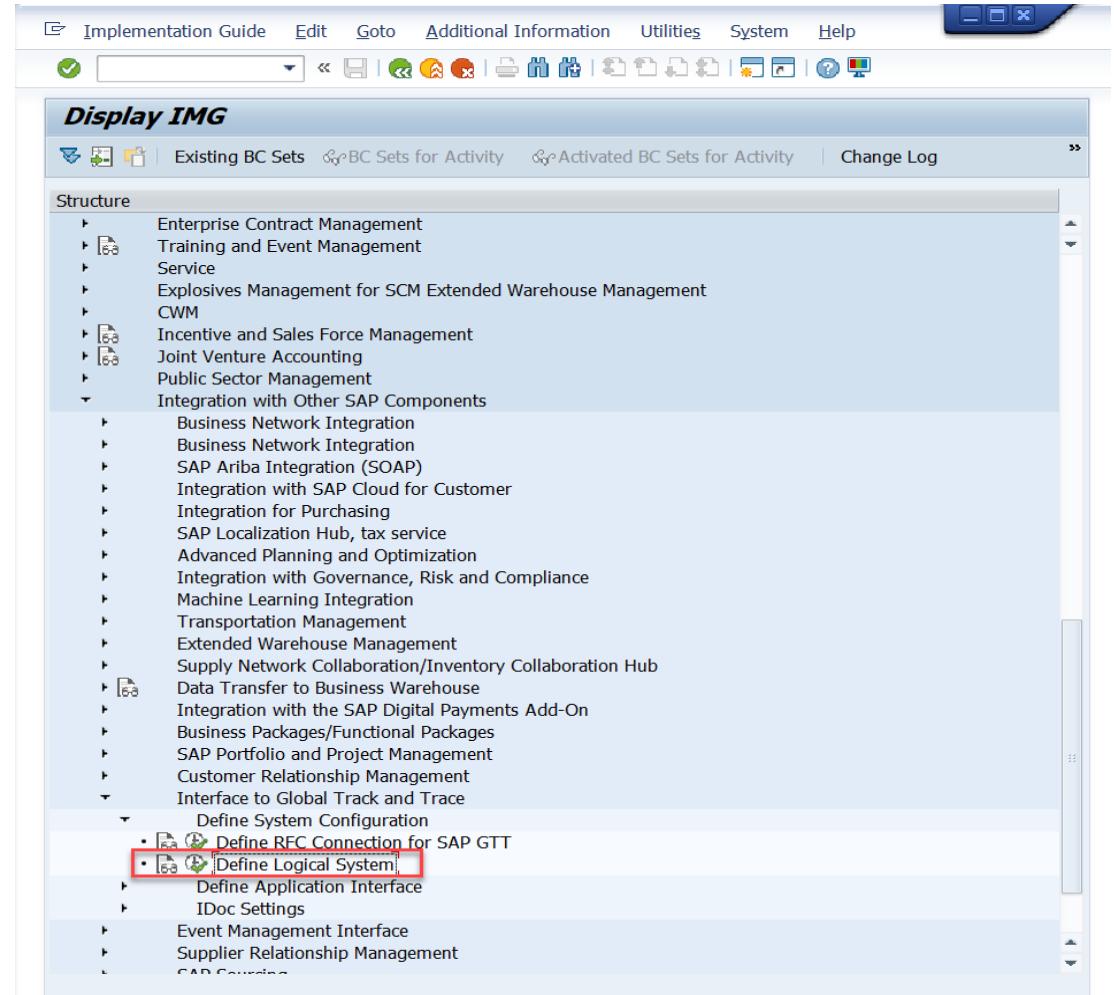
- one for event and
- the other for tracked process.



STEP 2: Define Logical System

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

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



STEP 2: Define Logical System

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

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

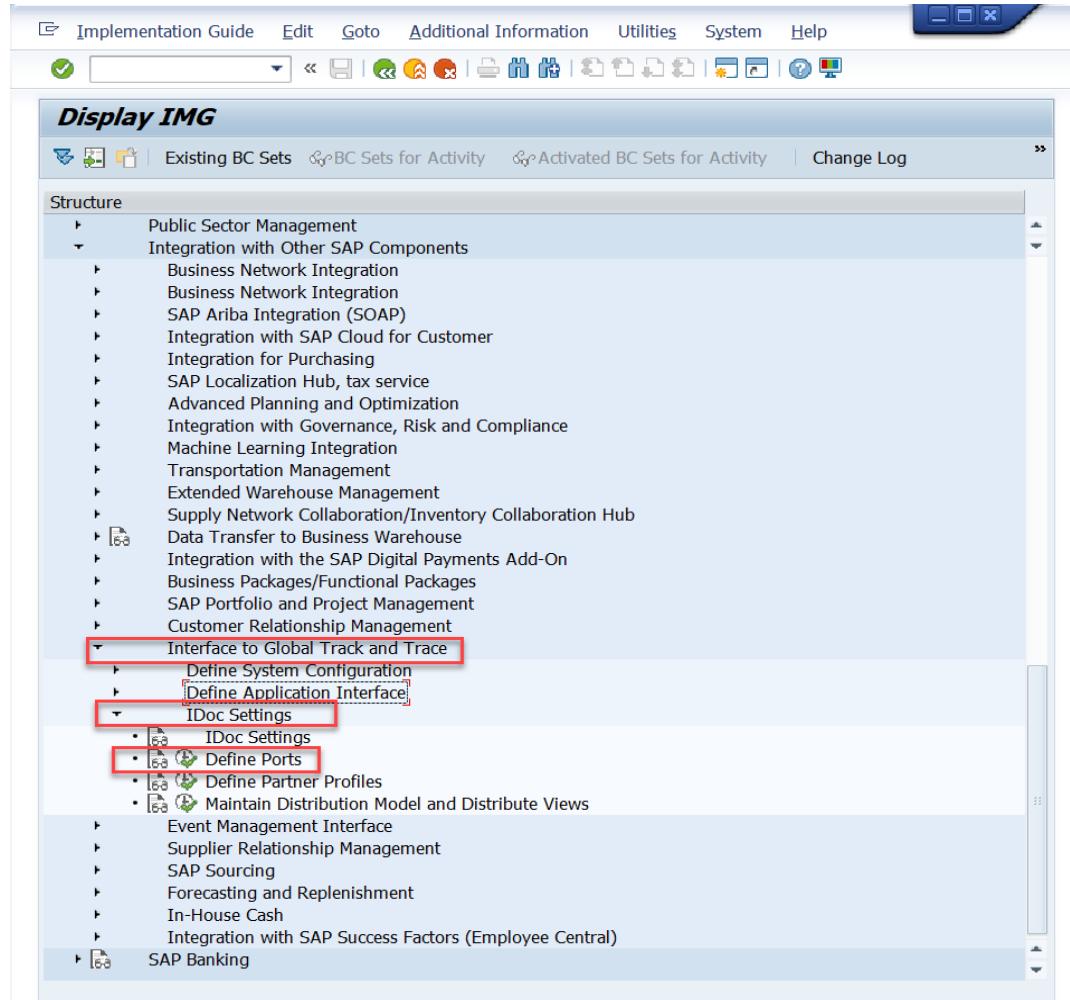
2-4: Save the configuration

| Log.System | Name |
|------------|-------------------|
| ZLSGTTINT | GTT V2 SAMPLE INT |

STEP 3: Define Ports

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

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



STEP 3: Define Ports

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

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

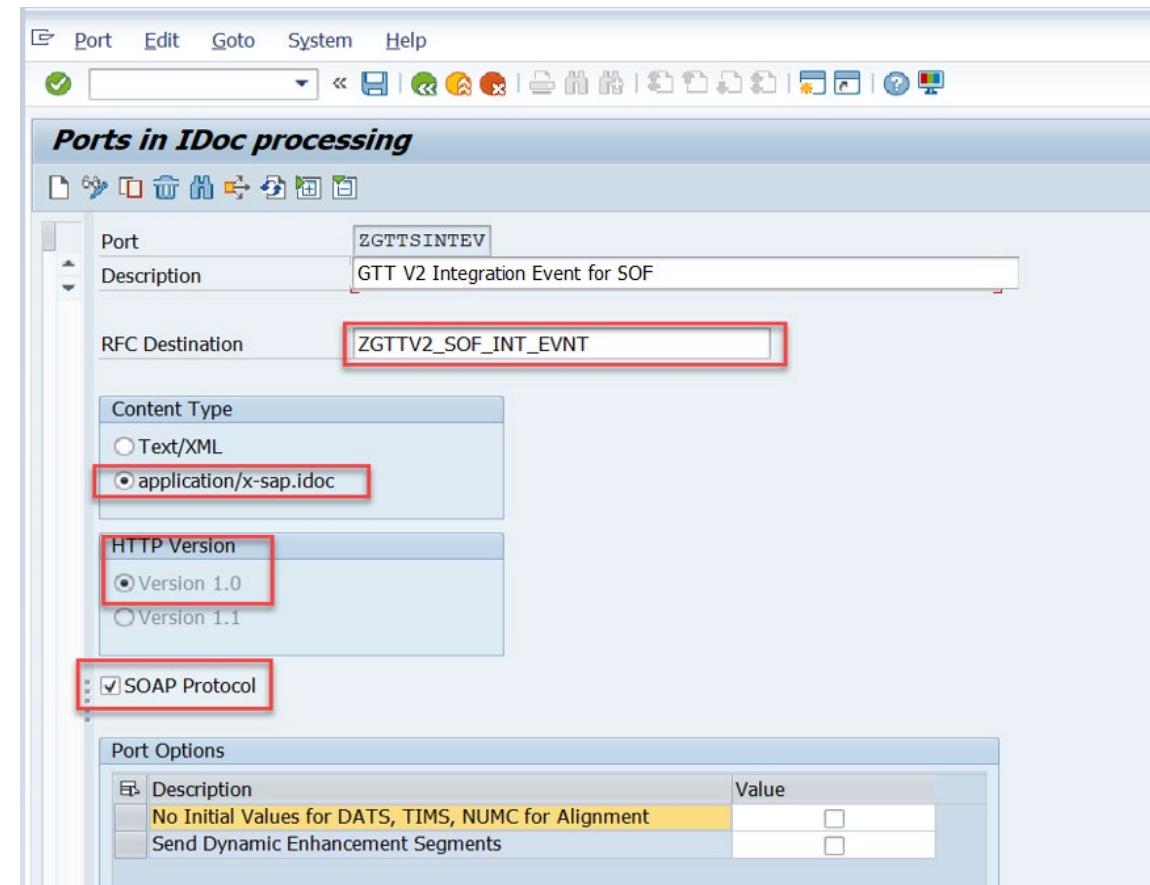
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

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

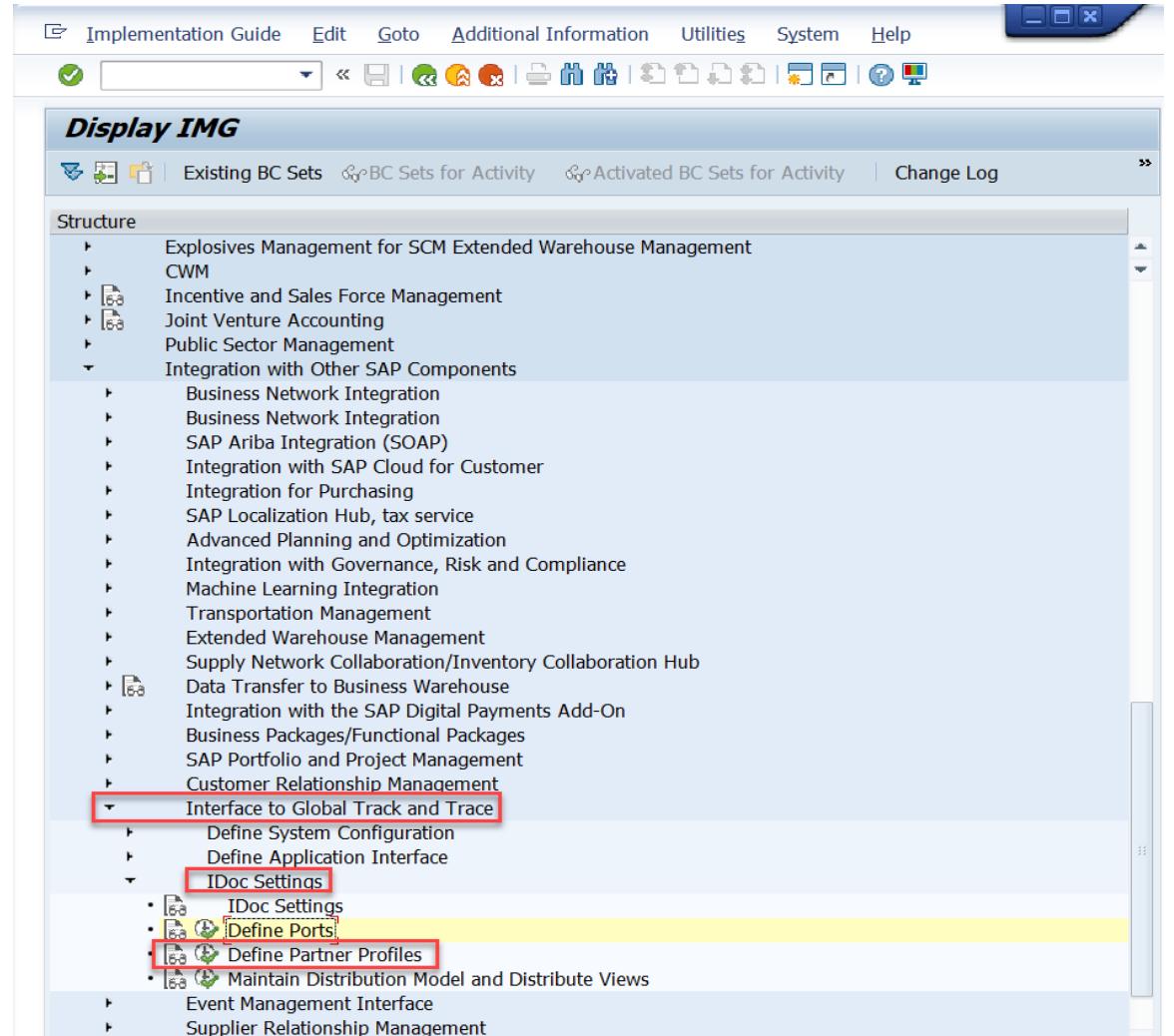


| Port | Description | RFC Destination | Content Type | HTTP Version | SOAP Protocol |
|------------|--|---------------------|------------------------|--------------|---------------|
| ZGTTSINTEV | GTT V2 Integration Event for SOF | ZGTTV2_SOF_INT_EVNT | application/x-sap.idoc | Version 1.0 | Checked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

STEP 4: Define Partner Profiles

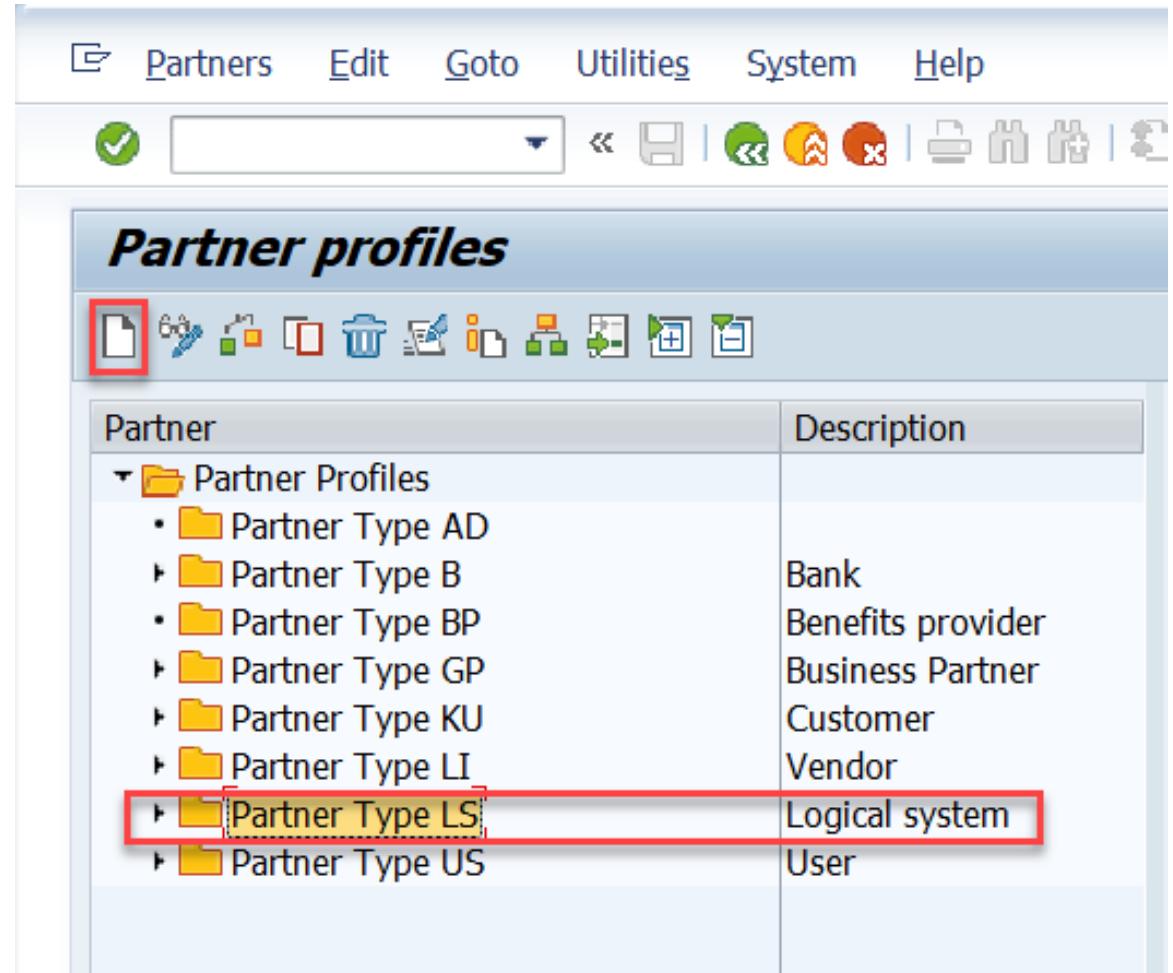
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

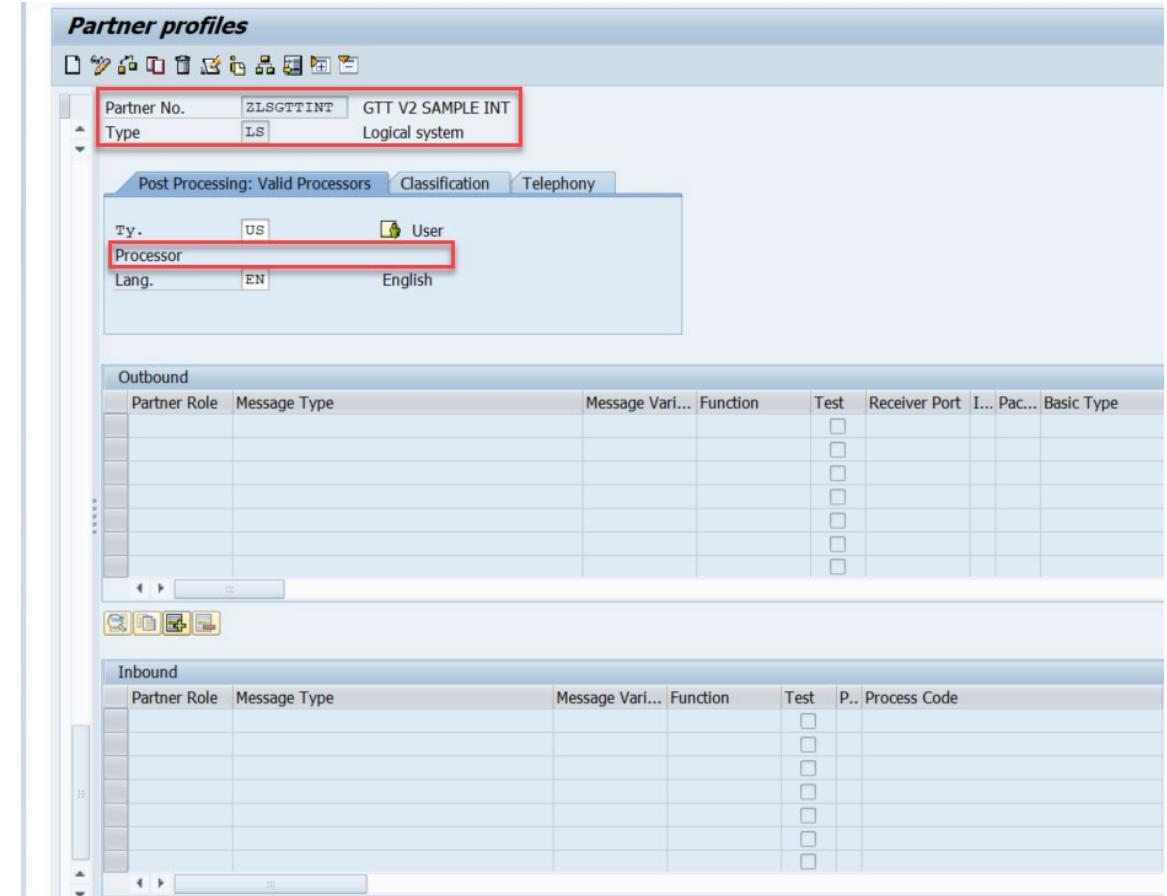
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

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

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



STEP 4: Define Partner Profiles

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

The screenshot shows the SAP Fiori interface for defining partner profiles. At the top, there's a toolbar with various icons. Below it, the main area is titled "Partner profiles". A header bar displays "Partner No. ZLSGTTINT", "Type LS", and "GTT V2 SAMPLE INT Logical system". There are three tabs: "Post Processing: Valid Processors" (selected), "Classification", and "Telephony". Under the "Post Processing" tab, there's a section for "Ty." (set to "US") and "User" (represented by a user icon). Below that, "Processor" and "Lang." (set to "EN") are listed, with "English" next to it. The main content area is divided into two sections: "Outbound" and "Inbound". The "Outbound" section has a table with columns: Partner Role, Message Type, Message Vari..., Function, Test, Receiver Port, I... Pac..., and Basic Type. The "Inbound" section has a similar table. In the bottom right corner of the "Outbound" table, there's a red box highlighting a small "Add" icon (a plus sign inside a circle) located between the "Function" and "Test" columns.

STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

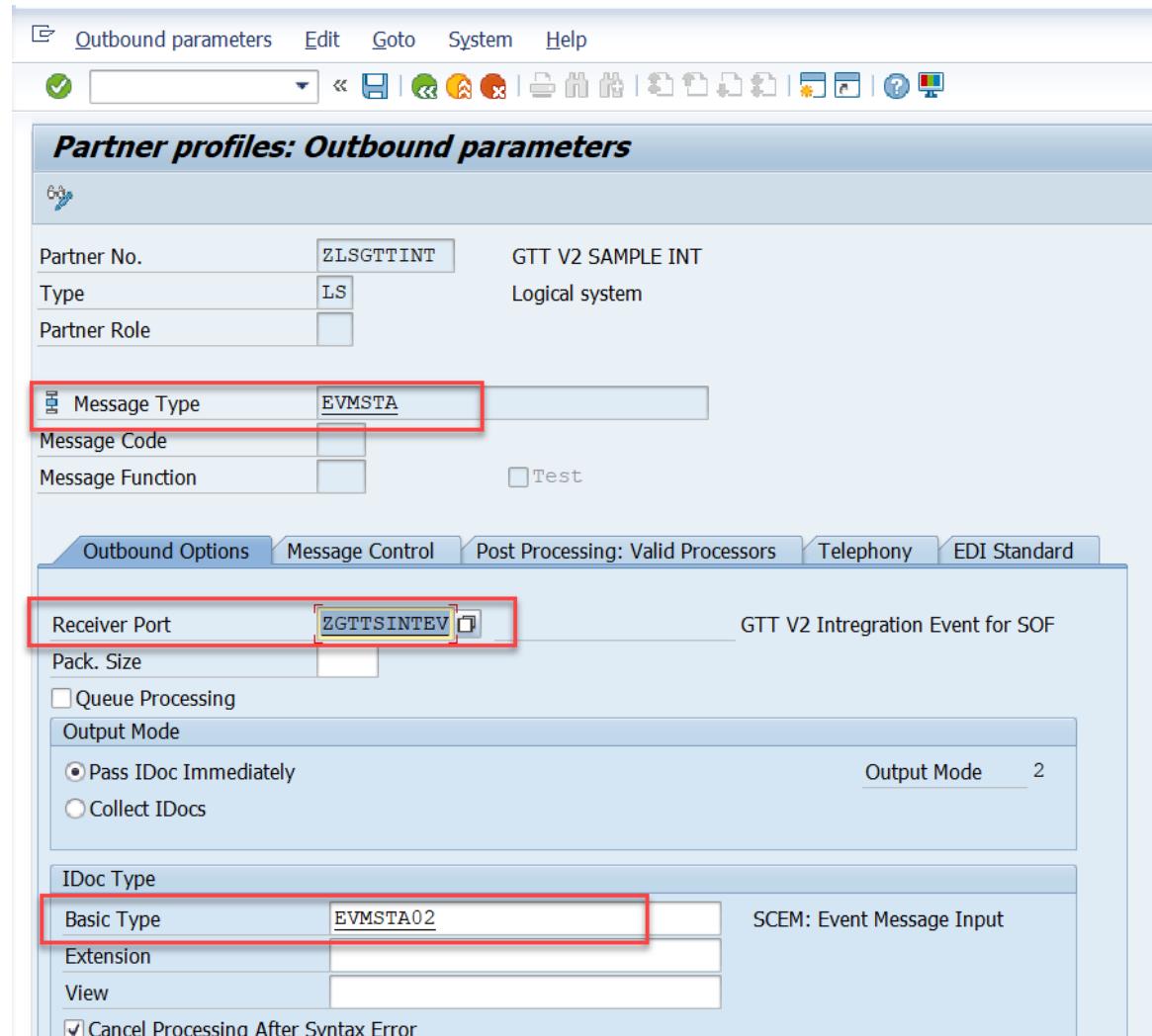
For the tracked Process:

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

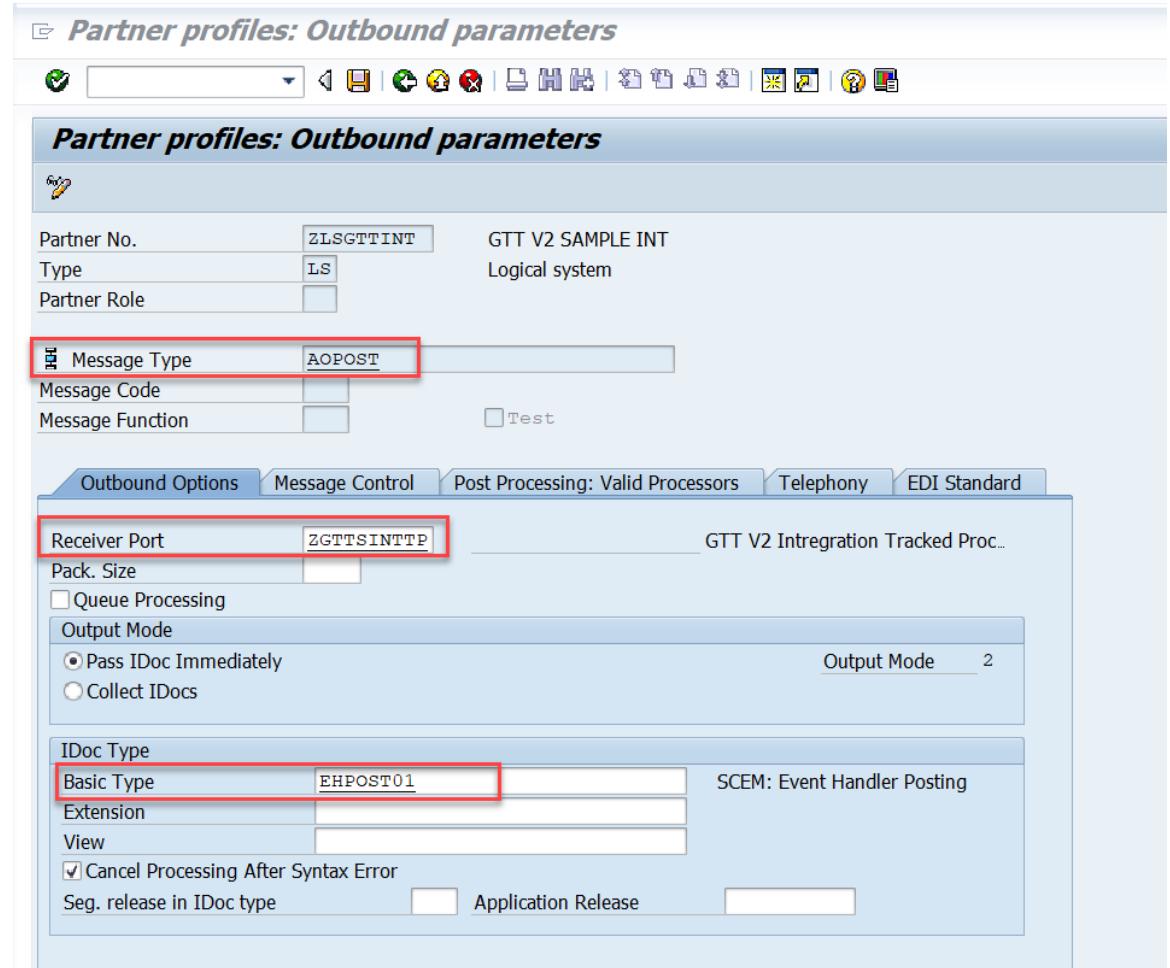
Message Code
Message Function Test

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

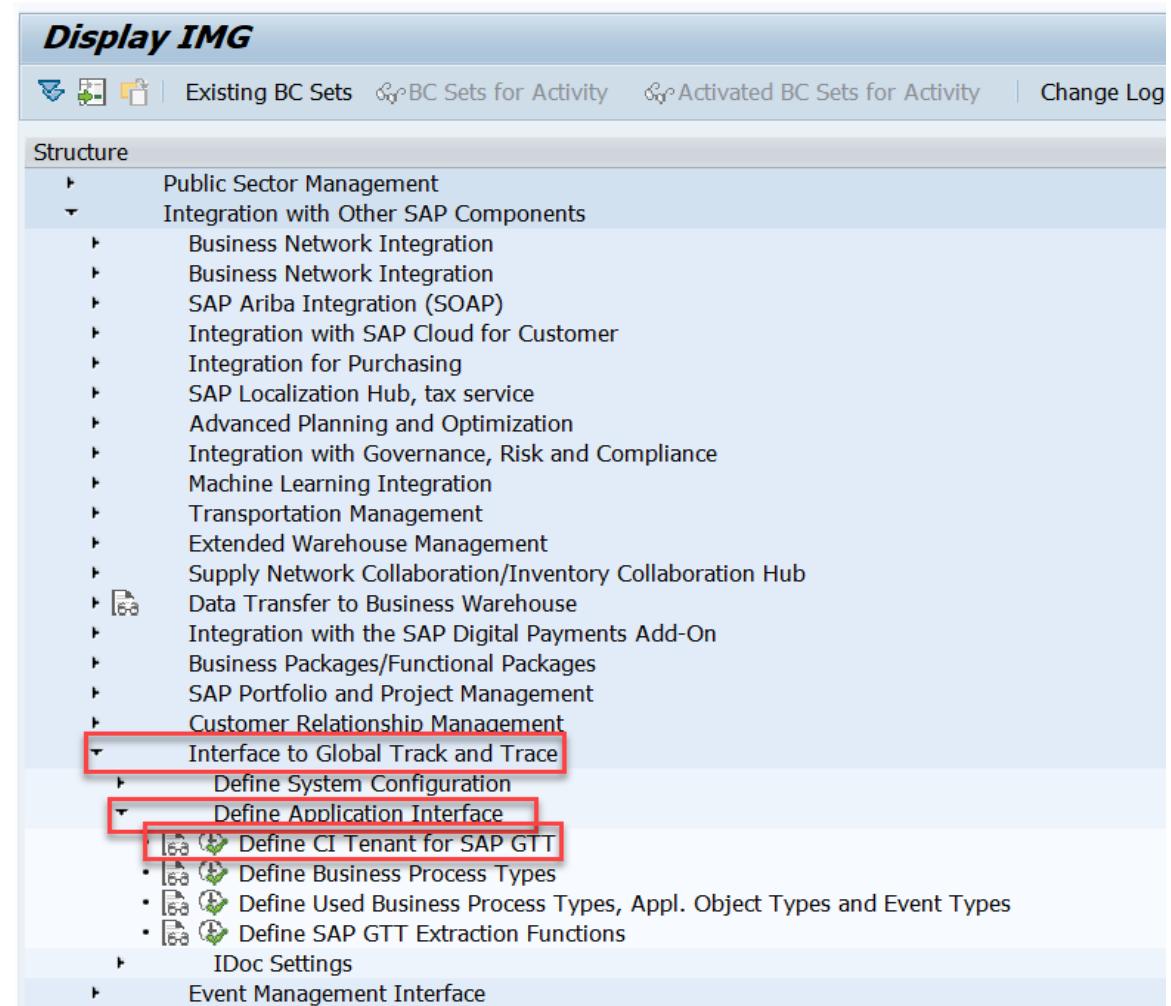
B2. Extractor Configuration



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

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

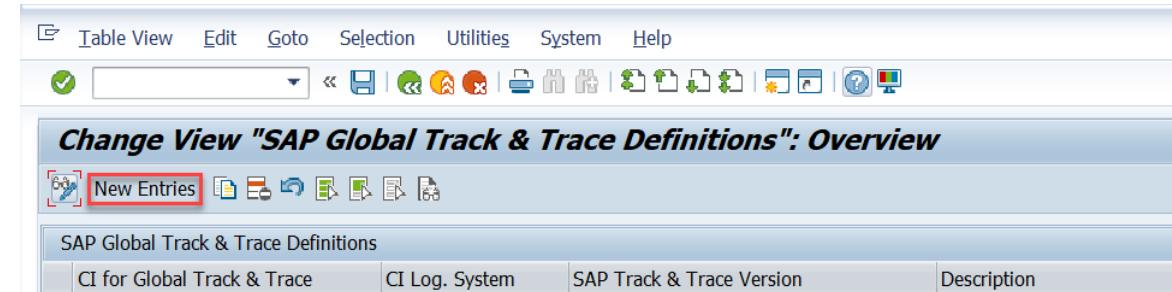
5-2: Choose activity
Define CI Tenant for SAP GTT



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

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

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



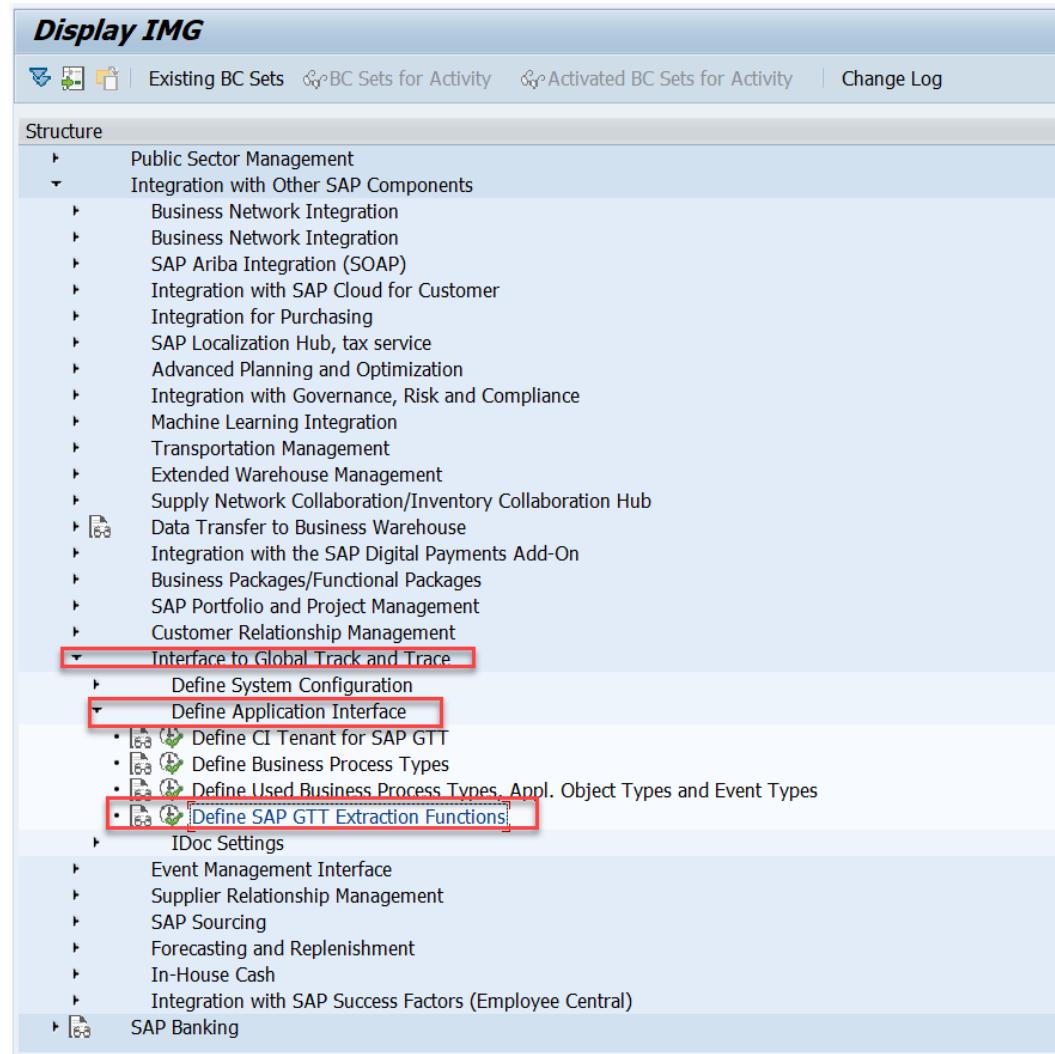
The screenshot shows the SAP Fiori interface for displaying global track and trace definitions. The title bar reads 'Display View "SAP Global Track & Trace Definitions": Overview'. Below the title bar is a toolbar with various icons. The main area displays a table with columns: 'SAP Global Track & Trace Definitions', 'CI for Global Track & Trace', 'CI Log. System', 'SAP Track & Trace Version', and 'Description'. The table currently contains one row with the following values:

| SAP Global Track & Trace Definitions | 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

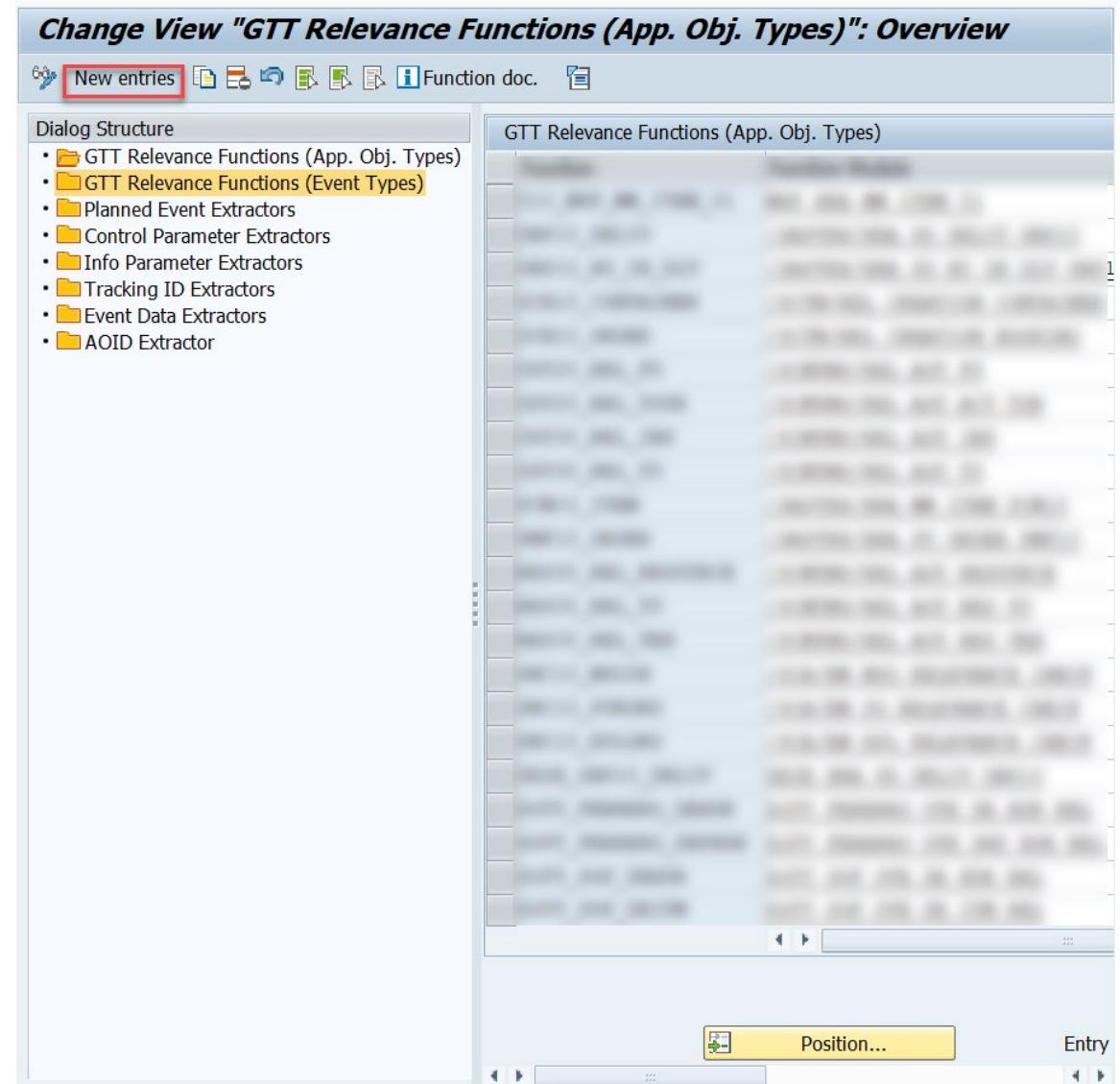
6-1: In **Display IMG** page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

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

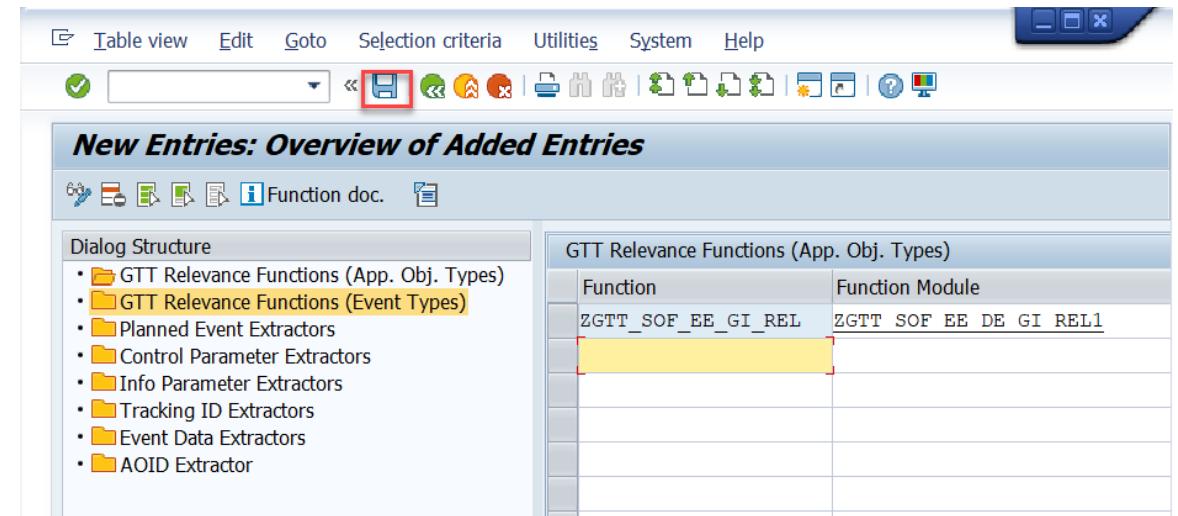
6-4: Input the **Function name** and **Function Module** for the newly created extraction function

The screenshot shows a SAP Fiori application titled "New Entries: Overview of Added Entries". The interface includes a toolbar with icons for search, refresh, and help, along with a "Function doc." button. On the left, a "Dialog Structure" sidebar lists various extractor types: GTT Relevance Functions (App. Obj. Types), GTT Relevance Functions (Event Types) (highlighted in yellow), Planned Event Extractors, Control Parameter Extractors, Info Parameter Extractors, Tracking ID Extractors, Event Data Extractors, and AOID Extractor. The main area displays a table titled "GTT Relevance Functions (App. Obj. Types)". The table has two columns: "Function" and "Function Module". A single row is present, containing the values "ZGTT_SOF_EE_GI_REL" and "ZGTT_SOF_EE_DE_GI_REL1", which are highlighted with a red border.

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

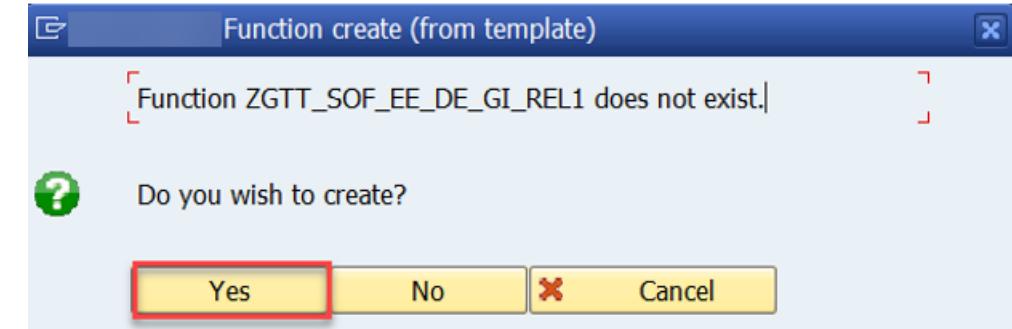
STEP 6: Define GTT Extraction Functions

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 "ZGTT_SOEE_DE_GI_REL1" is listed in the repository browser under the "Function Group" "ZGTT_SOEE" and the "Object Name" "ZGTT_SOEE". The code editor displays the ABAP code for the function module:

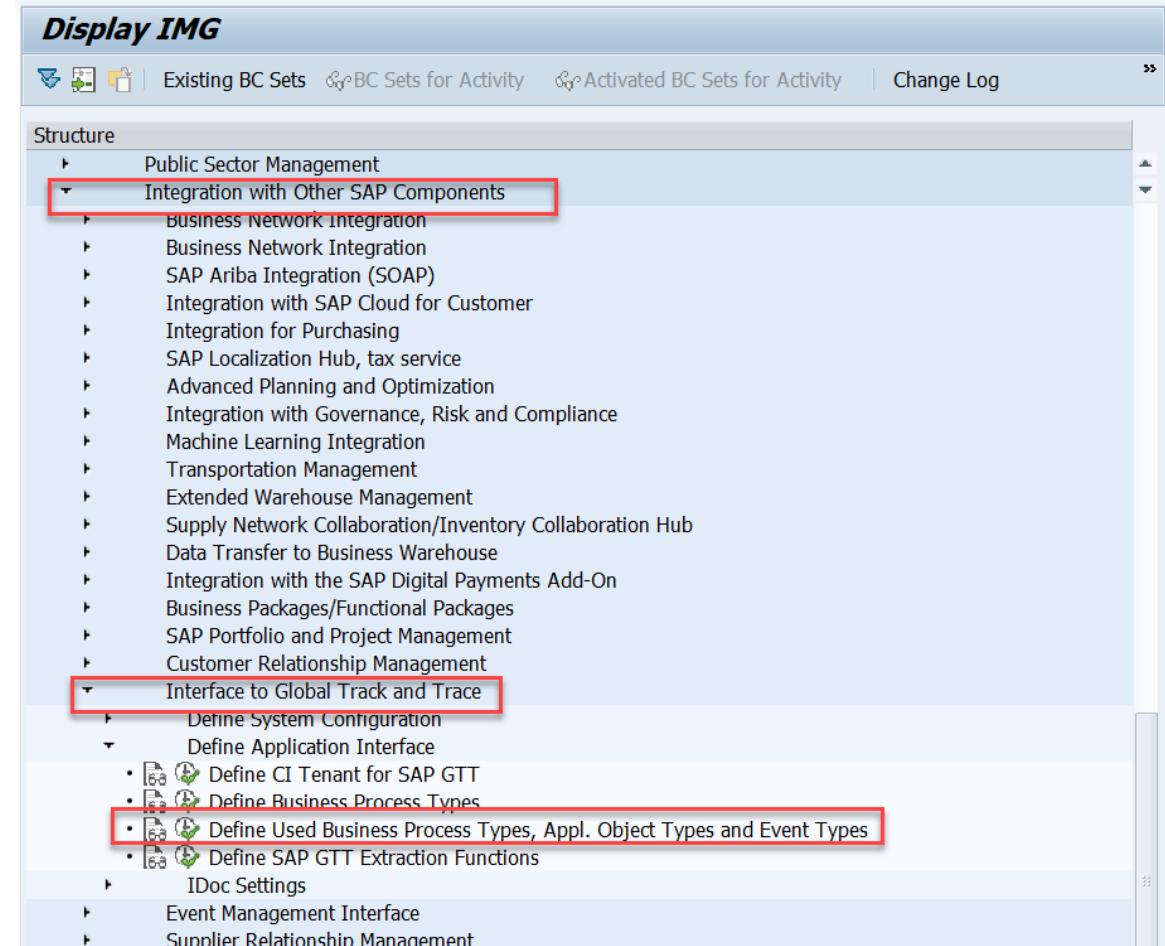
```
1 FUNCTION ZGTT_SOEE_DE_GI_REL1.
2
3     *" Local Interface:
4     *" IMPORTING
5     *"   REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM
6     *"   REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/AOTYPES
7     *"   REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER
8     *"   REFERENCE(I_APPTYPE_TAB) TYPE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

The code editor also shows the scope as "FUNCTION ZGTT_SOEE_DE_GI_REL1", the language as "ABAP", and the current line as "Ln 13 Col 48".

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

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

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



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

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

In the following:

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

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

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

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

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

| Change View "Define Used Business Process Types": Overview | | | |
|--|--|------------------|--|
| Define Used Business Process Types | | | |
| Bus. Proc. Type | Update Mode | BPT Process Mode | Description |
| EPL_NOTIF | Update Task .. | Active | Notification in SAP R/3 Enterprise |
| ESC_DELIV | Update Task .. | Active | Delivery in SAP R/3 Enterprise |
| ESC_FI_CLEARING | Update Task .. | Active | FI Clearing in SAP R/3 Enterprise |
| ESC_MATDOC | Update Task .. | Active | Material Document in SAP R/3 Enterprise |
| ESC_MM_INVOICE | Update Task .. | Active | MM Invoice in SAP R/3 Enterprise |
| ESC_PURORD | Update Task .. | Active | Purchase Order in SAP R/3 Enterprise |
| ESC_PURORD_FASHION | Update Task .. | Active | Purchase Order (Seasonal Procurement) in SAP R/3 Enterprise |
| ESC_SHIPMT | <input checked="" type="checkbox"/> Update Task .. | Active | Shipment (SAP R/3 Enterprise) |
| ESC_SORDER | Update Task .. | Active | Sales Order in SAP R/3 Enterprise |
| ESC_WRKORD | Update Task .. | Active | Workorder (Production, Service, Maintenance) in SAP R/3 Enterprise |
| OCB10_ORDER | Dialog Update .. | Active | Booking Order in Ocean Carrier Booking Process |
| SNC_MSGIN | Dialog Update .. | Active | SNC Inbound messages |
| SNC_PURORD | Dialog Update .. | Active | SNC Purchase Order |
| SNC_RPLORD | Dialog Update .. | Active | SNC Replenishment Order |
| TMS_INS | Update Task .. | Active | Instructions (SAP TM) |
| TMS_RES | Update Task .. | Active | Resources (SAP TM) |
| TMS_TOR | Update Task .. | Active | Transportation Order (SAP TM) |

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

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

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

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

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

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

7-8: Check **GTT Relevant**

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

General Data Control Tables Global Track & Trace Relevance

| | |
|--------------------------|--|
| Sequencing / Destination | |
| Seq. No. | 10 |
| HCI for GTT | ZGTTSOFINTE 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: (empty)
 - Old Main Obj. Table: SHIPMENT_HEADER_OLD (highlighted with a red box)
 - Old Master Table: (empty)
- Reference Between Main and Master Table:**
 - First Field Reference from Main to Master Table: (empty)
 - Second Field Reference from Main to Master Table: (empty)

A red box highlights the "Event on Header level" note next to the Main Obj. Table entry.

Bottom Screenshot (Business Process Type: ESC_DELIV):

- General Data:**
 - Bus. Proc. Type: ESC_DELIV
 - Event Type: ZGTT_SOF_PICKING_INT
 - Text: Picking Event
- 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)
 - Old Main Obj. Table: DELIVERY_ITEM_OLD (highlighted with a red box)
 - Old Master Table: DELIVERY_HEADER_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
 - Uplink Target Fld: VBELN
 - Uplink Const: (empty)
 - Second Field Reference from Main to Master Table:
 - Uplink Field: (empty)
 - Uplink Mode: (empty)
 - Uplink Target Fld: (empty)
 - Uplink Const: (empty)

A red box highlights the "Event on Item level" note next to the Main Obj. Table entry.

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**.

| | |
|--------------------------------|------------------------------|
| Bus. Proc. Type | ESC_SHIPMT |
| Event Type | ZGTT_SOF_CHIN_INT |
| Text | Check In Event |
| General Data | |
| Control Tables | |
| Global Track & Trace Relevance | |
| GTT Rel. Method | Check Function (Function...) |
| GTT Rel. Function | ZGTT_SOF_SHP_CHI_REL |

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

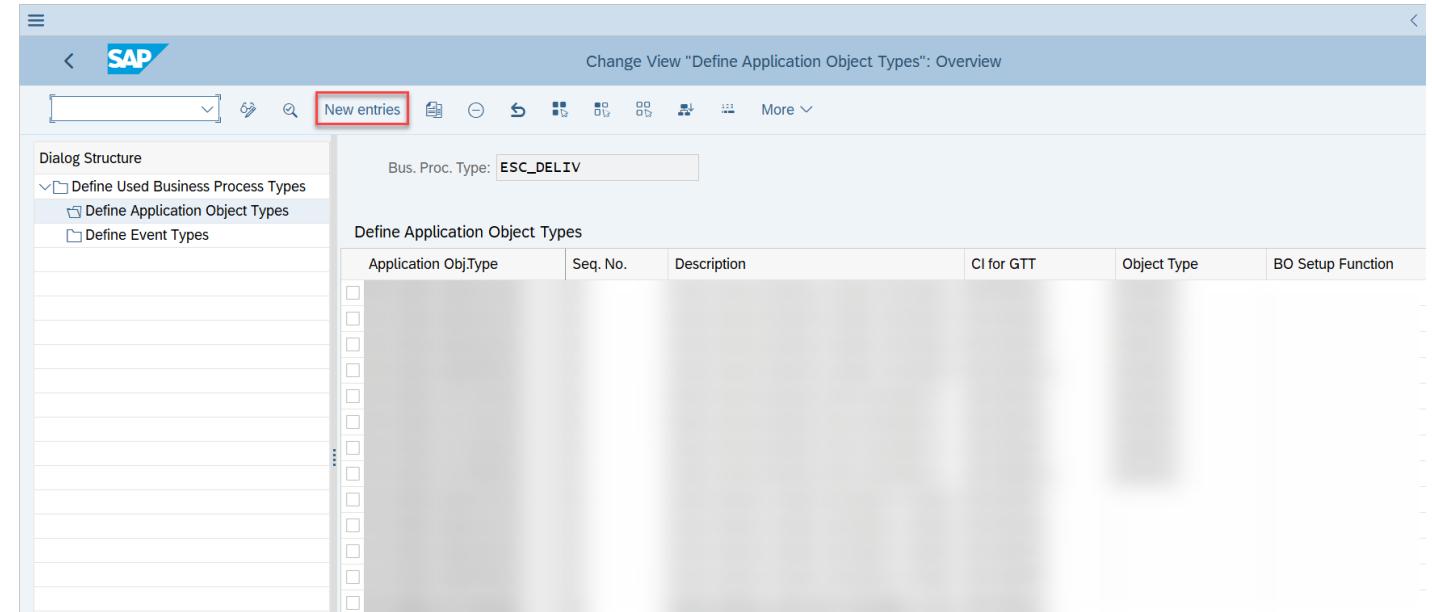
7-12: Double click **Define Application Object Types**

The screenshot shows the SAP Change View "Define Used Business Process Types". The title bar reads "Change View 'Define Used Business Process Types': Overview". The toolbar includes buttons for New Entries, Copy As..., Delete, Undo Change, Select All, Select Block, Deselect All, Configuration Help, and More. The left sidebar, titled "Dialog Structure", 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, titled "Define Used Business Process Types", contains a table with columns: Bus. Proc. Type, Update Mode, BPT Process Mode, and Description. The table lists various business process types, each with a checkbox and a dropdown menu icon. Some rows have a red box around them, specifically the first row (EPL_NOTIF) and the row for ESC_SHIPMT.

| 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 | 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 (V...) | Active | Instructions (SAP TM) |
| TMS_RES | Update Task (V...) | Active | Resources (SAP TM) |
| TMS_TOR | Update Task (V...) | Active | Transportation Order (SAP TM) |

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

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



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

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

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

7-16: Check **GTT Relevant**

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

- Bus. Proc. Type: ESC_SHIPMT
- Appl. Obj. Type: ZGTT_SHP_INT_HD (highlighted with a red box)
- Text: Extract shipment header information to Global Track and Trace Integration

Below this, the General Data tab is selected, showing the following configuration:

- Sequencing / Destination:
 - Seq. No.: 20
 - CI for GTT: ZGTTSOFINST (highlighted with a red box)
 - CI For GTT V2 Integration system Sales Order Sampl
- Business Object Reference:
 - Object Type: [empty]
 - BO Setup Fnct.: [empty]
- Behavior:
 - 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 screenshot shows the SAP Fiori interface for defining business process types, application object types, and event types. It is divided into two main sections: one for the business process type **ESC_SHIPMT** and one for **ESC_DELIV**.

ESC_SHIPMT Configuration:

- General Data:** Bus. Proc. Type: **ESC_SHIPMT**, Appl. Obj. Type: **ZGTT_SHP_INT_HD**, Text: **Extract shipment header information to Global Track and Trace Integration**.
- Control Tables:** Data Source for Created and Updated Objects: Main Obj. Table: **SHIPMENT_HEADER_NEW** (highlighted with a red box), Master Table: **SHIPMENT_HEADER_OLD**. AOT on Header Level (highlighted with a red box).
- Reference Between Main and Master Table:** First Field Reference from Main to Master Table.

ESC_DELIV Configuration:

- General Data:** Bus. Proc. Type: **ESC_DELIV**, Appl. Obj. Type: **ZGTT_DE_INT_ITEM**, Text: **Extract delivery order item information to Global Track and Trace Integration**.
- Control Tables:** Data Source for Created and Updated Objects: Main Obj. Table: **DELIVERY_ITEM_NEW** (highlighted with a red box), Master Table: **DELIVERY_HEADER_NEW**. AOT on Item Level (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**, Uplink Target Fld: **VBELN**, Uplink Const: .
- Global Track & Trace Relevance:** This section is partially visible at the bottom of the screenshot.

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.

The screenshot shows the 'Object Identification' tab of a configuration screen. At the top, there are three input fields: 'Bus. Proc. Type' (ESC_DELIV), 'Appl. Obj. Type' (ZGTT_DE_INT_ITEM), and 'Text' (Delivery Item). To the right of the type field is a tooltip: 'Extract delivery order item information to Global Track and Trace Integration'. Below these fields are five tabs: General Data, Control Tables, Object Identification (underlined), Global Track & Trace Relevance, and Parameter Setup. Under the 'Object Identification' tab, there is a section titled 'Method for determination of AOID'. A dropdown menu labeled 'AOID Method' is set to 'Determine from Field', which is highlighted with a red box. Below this, there are two sections for 'Application Object ID Source': 'First Field to Build Appl. Obj. ID' and 'Second Field to Build Appl. Obj. ID'. Each section contains a dropdown for 'Cntrl Tab. Type' (set to '1 Main Object Table') and an input field for 'AO ID Field' ('VBELN' and 'POSNR' respectively), both of which are also highlighted with red boxes. At the bottom, there is a section titled 'Determine AOID By Function' with an input field for 'AOID Function'.

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

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

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

The screenshot shows a configuration interface for a business process. At the top, there are three input fields: 'Bus. Proc. Type' (ESC_DELIV), 'Appl. Obj. Type' (ZGTT_DE_INT_ITEM), and 'Text' (Delivery Item). Below these, a tooltip reads: 'Extract delivery order item information to Global Track and Trace Integration'. A 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 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 is a 'Tracking ID Fld' input field. Under 'Parameter Setup', three fields are shown: 'Ctrl Data Function' (ZGTT_OTE_DE_ITEM), 'Info Data Function' (empty), and 'Planned Event Function' (ZGTT_EE_DE_ITM), all highlighted with red boxes.

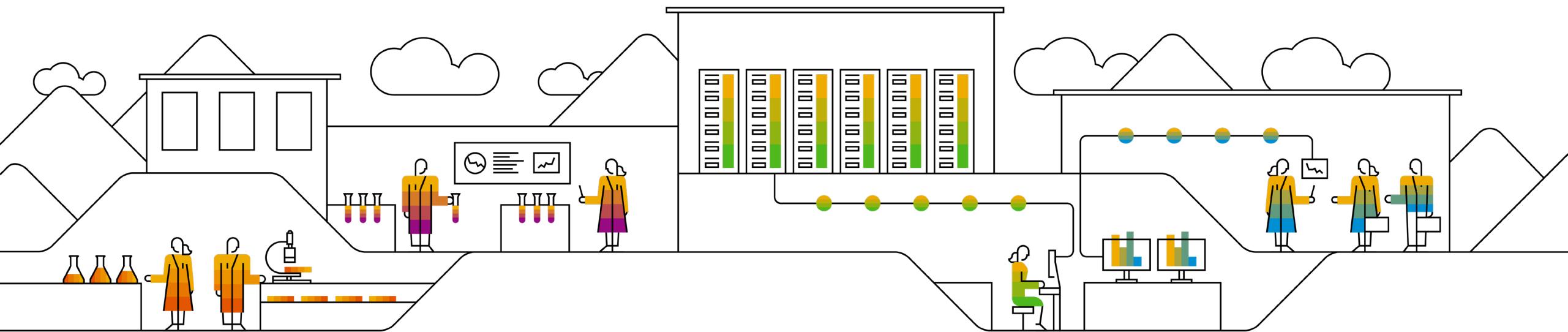
C) Download ABAP Code from GitHub

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

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

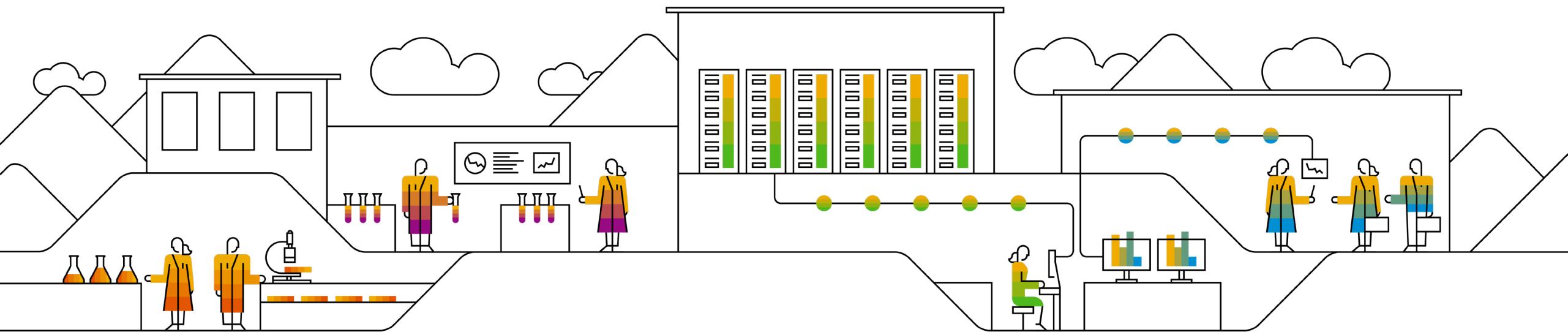
C3. Download Another ABAP Code from GitHub (Only for TPOF)

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 stating that abapGit exists in two flavors: "standalone" and "developer". It then describes the "standalone" version as targeted for users and the "developer" version as targeted for developers contributing to the codebase. A "Prerequisites" section requires SAP BASIS version 702 or higher. The "Install standalone version" section is highlighted with a red border and contains four numbered steps: 1. Download ABAP code, 2. Create a report named ZABAPGIT_STANDALONE, 3. Upload code to the report, and 4. Activate the report. Below this, a note says abapGit is typically used in development systems and can be installed in a local package. A final note states it can be used via transaction SE38.

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

- The *standalone* version is targeted at users. It consist of one (huge) program which contains all the needed code. You run the *standalone* version in transaction `SE38`, executing the program you created.
- The *developer* version is targeted at developers contributing to the abapGit codebase. It consists of all the ABAP programs/classes/interfaces/etc. of the abapGit project. You run the *developer* version with transaction `ZABAPGIT`.

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

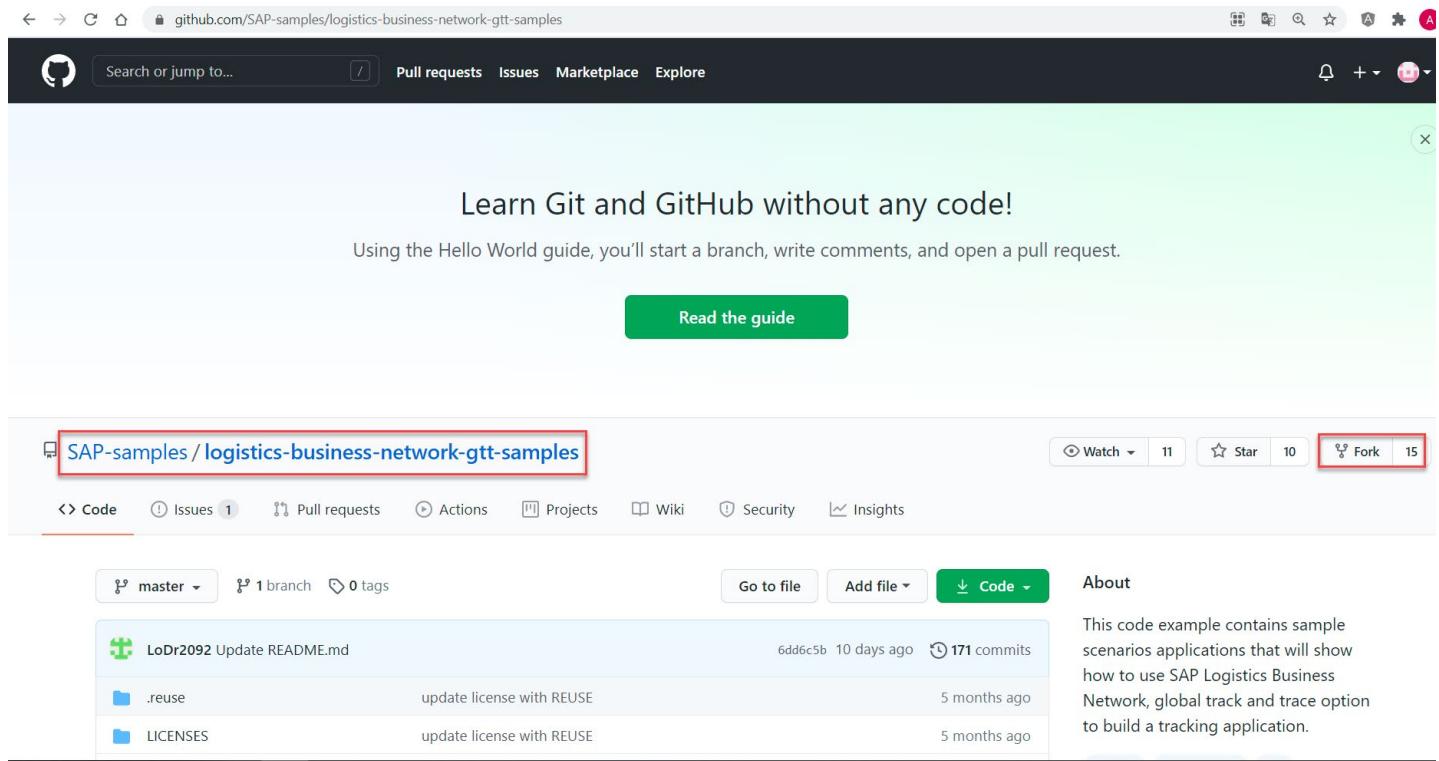
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

STEP 2: Fork Sample Code Repository

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

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



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

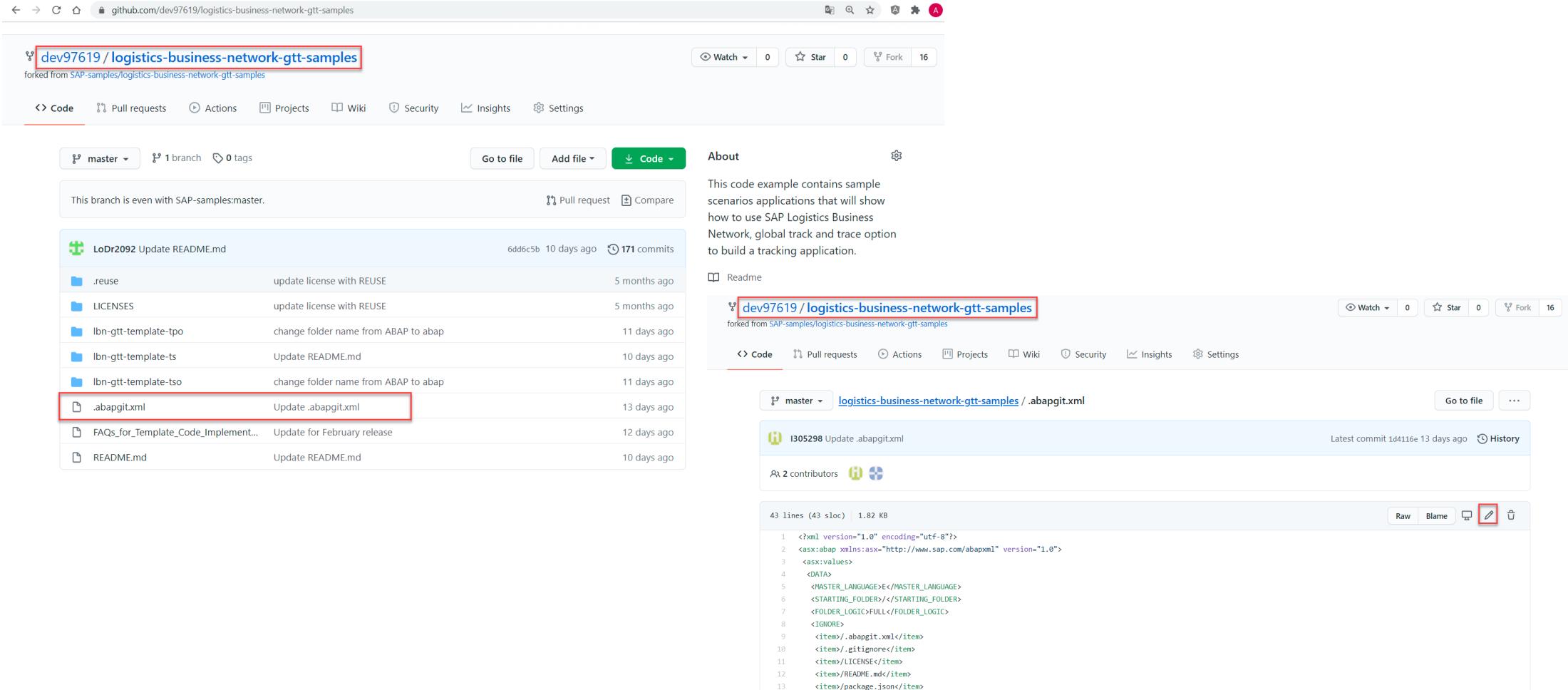
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit 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 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: `LoDr2092 Update README.md`. Below the commits, there is an 'About' section describing the code example. On the right side of the page, there is a preview of the `.abapgit.xml` file content, which is an XML configuration file. The file is 43 lines long and contains various XML elements like `<MASTER_LANGUAGE>`, `<STARTING_FOLDER>`, and `<IGNORE>`.

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.

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

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

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

3-4: Commit change.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The '.abapgit.xml' file is open in the editor, showing the XML configuration. A specific line of code is highlighted with a red box: '<STARTING_FOLDER>/</STARTING_FOLDER>'. To the right, a 'Commit changes' dialog is displayed, also with a red box highlighting the same line in the 'Update .abapgit.xml' text area. The dialog includes fields for an optional extended description and two radio button options: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request.' Below the dialog are 'Commit changes' and 'Cancel' buttons.

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

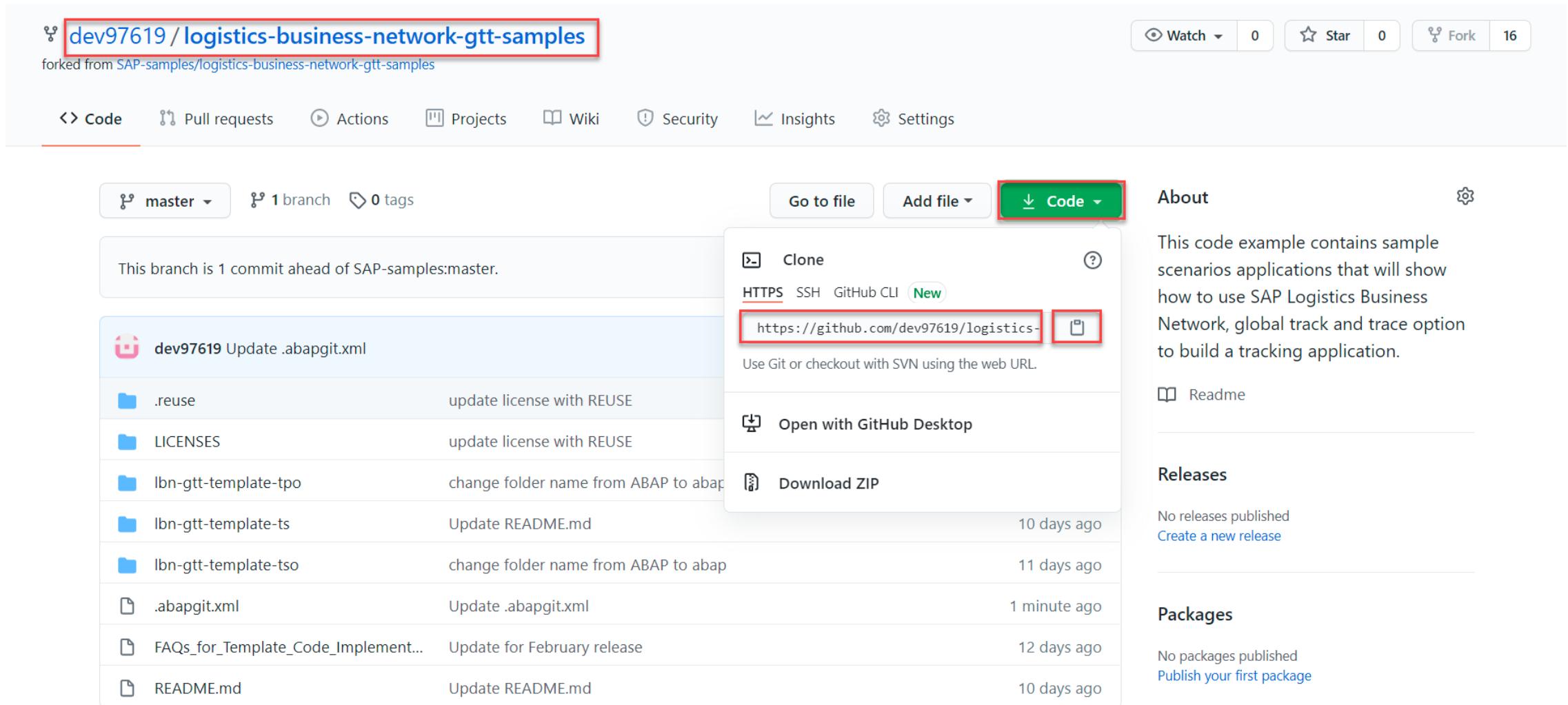
-o- Commit directly to the master branch.

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

Commit changes Cancel

STEP 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. 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 0 stars, 16 forks, and 0 issues. The 'About' section describes the code example as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options. The 'Readme' and 'Releases' sections are also visible.

Code

Watch 0 Star 0 Fork 16

Code

Clone

HTTPS SSH GitHub CLI New

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

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

Ibn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

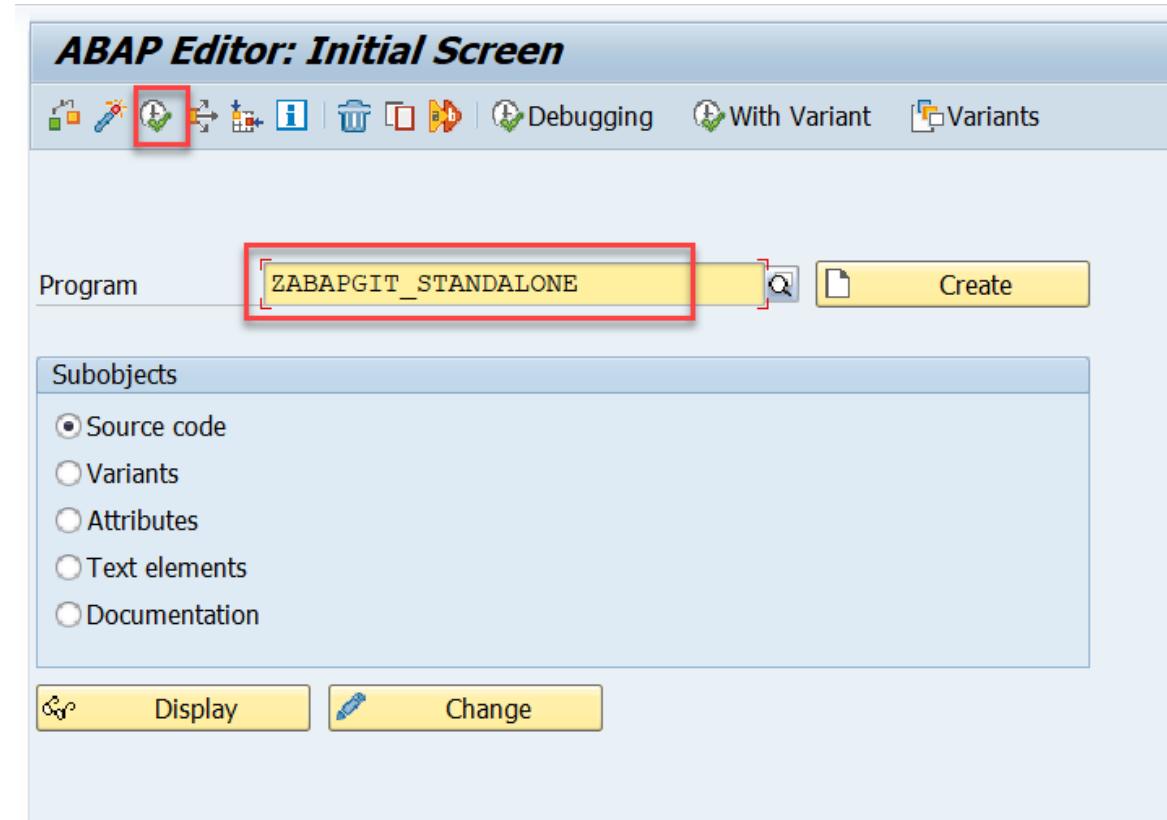
FAQs_for_Template_Code_Implement... Update for February release

README.md Update README.md

STEP 4: Download ABAP Code from GitHub

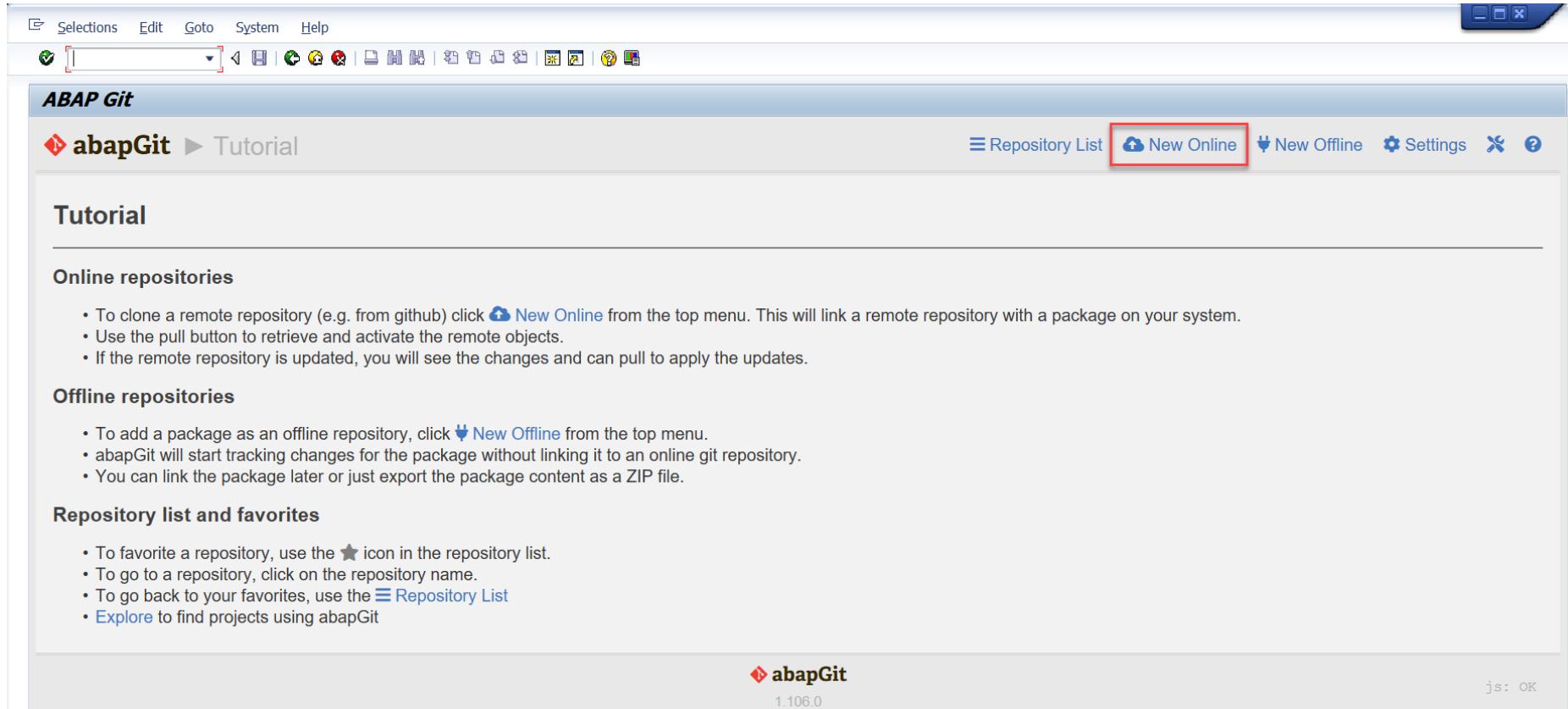
4-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

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

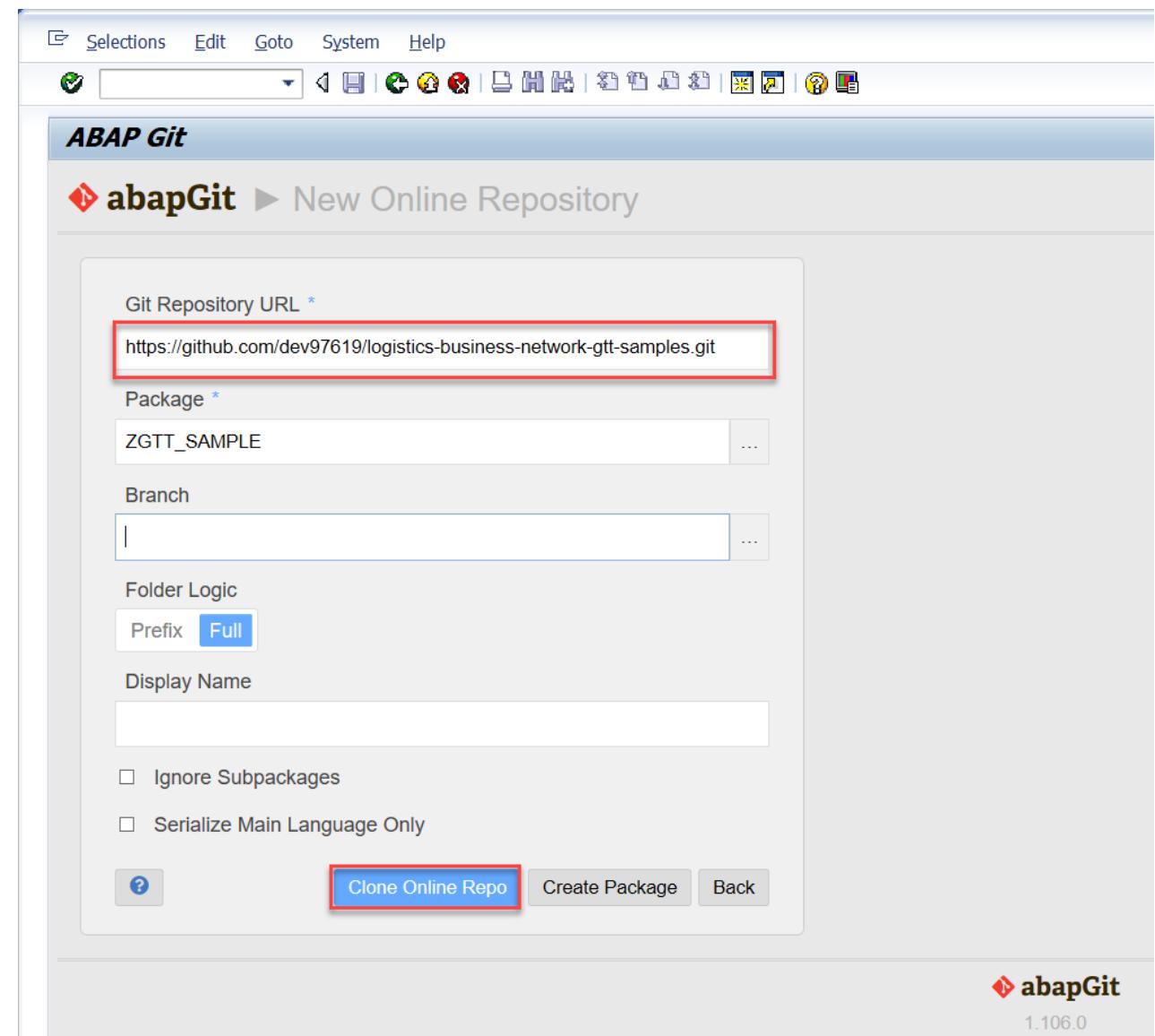
Caution:

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

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

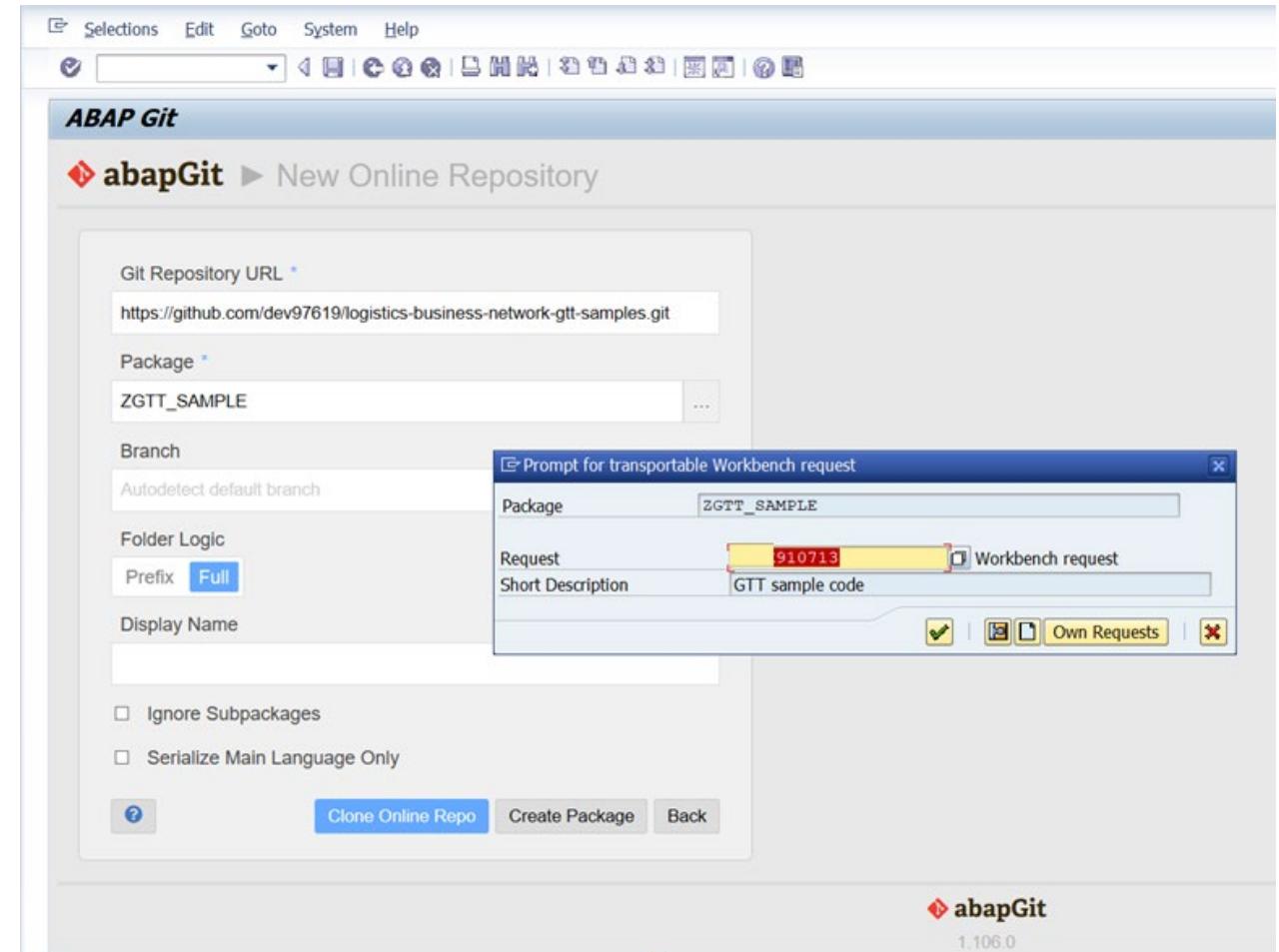
4-6: Set *Full* for **Folder Logic**

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



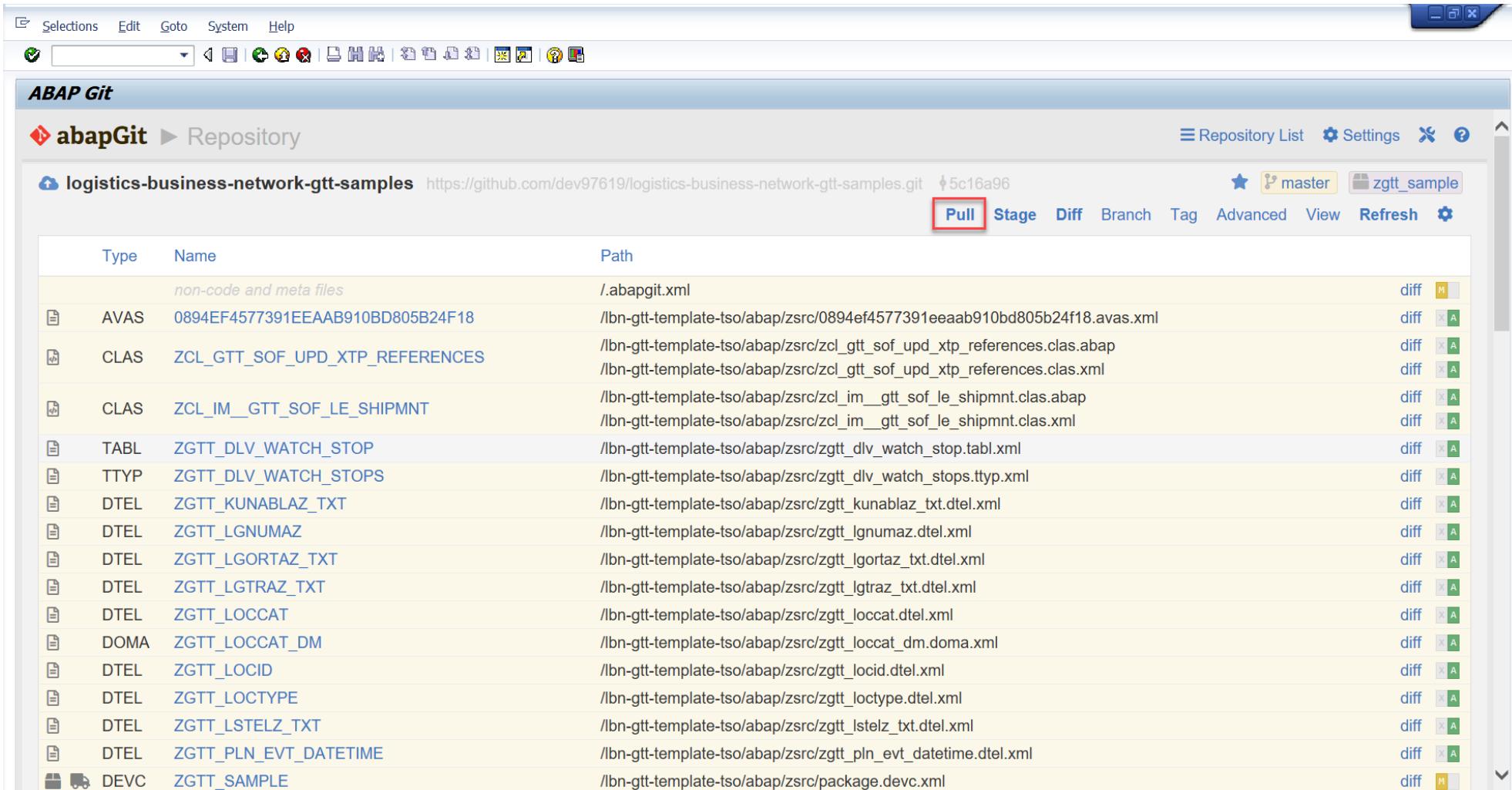
STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

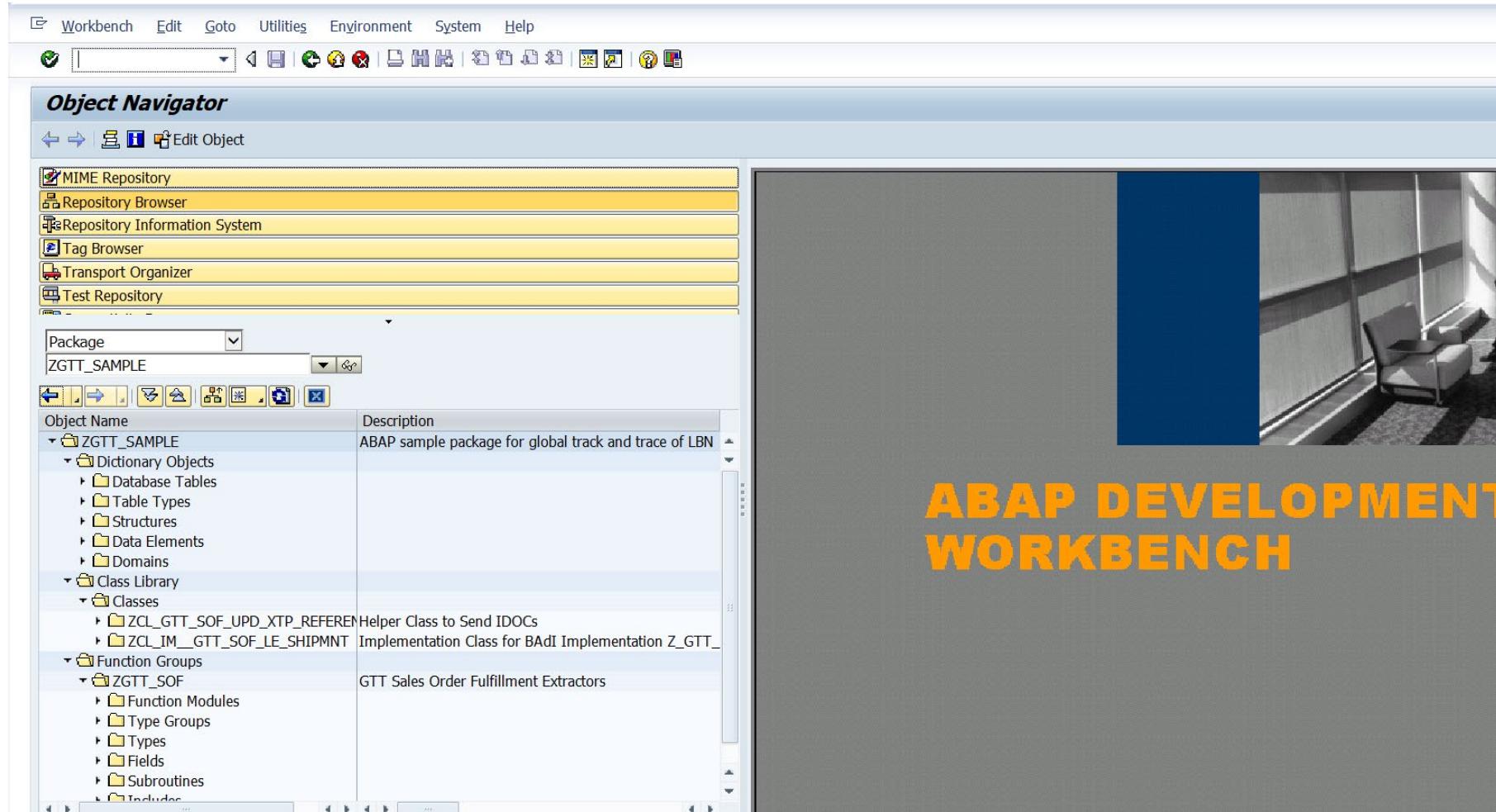


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the 'abapGit' repository. A breadcrumb navigation shows 'Repository'. The repository name is 'logistics-business-network-gtt-samples' with the URL 'https://github.com/dev97619/logistics-business-network-gtt-samples.git'. The commit hash '5c16a96' is displayed. The master branch is selected. A 'zgtt_sample' folder is visible. A toolbar at the top right includes 'Repository List', 'Settings', 'Refresh', and other options. The central part of the screen displays a table of files with columns for Type, Name, and Path. The 'Pull' button in the toolbar is highlighted with a red box. The table lists numerous files, mostly XML or ABAP classes, under various types like AVAS, CLAS, TABL, TTYP, DTEL, DOMA, and DEV. Each file entry includes a 'diff' link and a status indicator icon.

| Type | Name | Path | diff |
|------|----------------------------------|---|------------|
| | non-code and meta files | .abapgit.xml | [H] |
| 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_LGNUCMAZ | /lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml | [diff] [A] |
| DTEL | ZGTT_LGORAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml | [diff] [A] |
| DTEL | ZGTT_LGTRAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml | [diff] [A] |
| DTEL | ZGTT_LOCCAT | /lbn-gtt-template-tso/abap/zsrc/zggt_loccat.dtel.xml | [diff] [A] |
| DOMA | ZGTT_LOCCAT_DM | /lbn-gtt-template-tso/abap/zsrc/zggt_loccat_dm.doma.xml | [diff] [A] |
| DTEL | ZGTT_LOCID | /lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml | [diff] [A] |
| DTEL | ZGTT_LOCTYPE | /lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml | [diff] [A] |
| DTEL | ZGTT_LSTELZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml | [diff] [A] |
| DTEL | ZGTT_PLN_EVT_DATETIME | /lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml | [diff] [A] |
| DEV | ZGTT_SAMPLE | /lbn-gtt-template-tso/abap/zsrc/package.devco.xml | [diff] [H] |

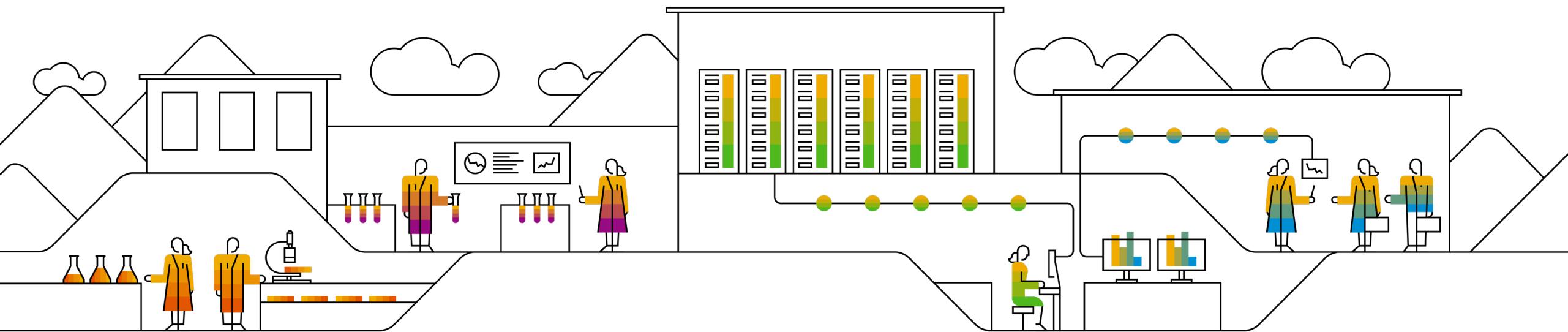
STEP 4: Download ABAP Code from GitHub

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



C) Download ABAP Code from GitHub

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

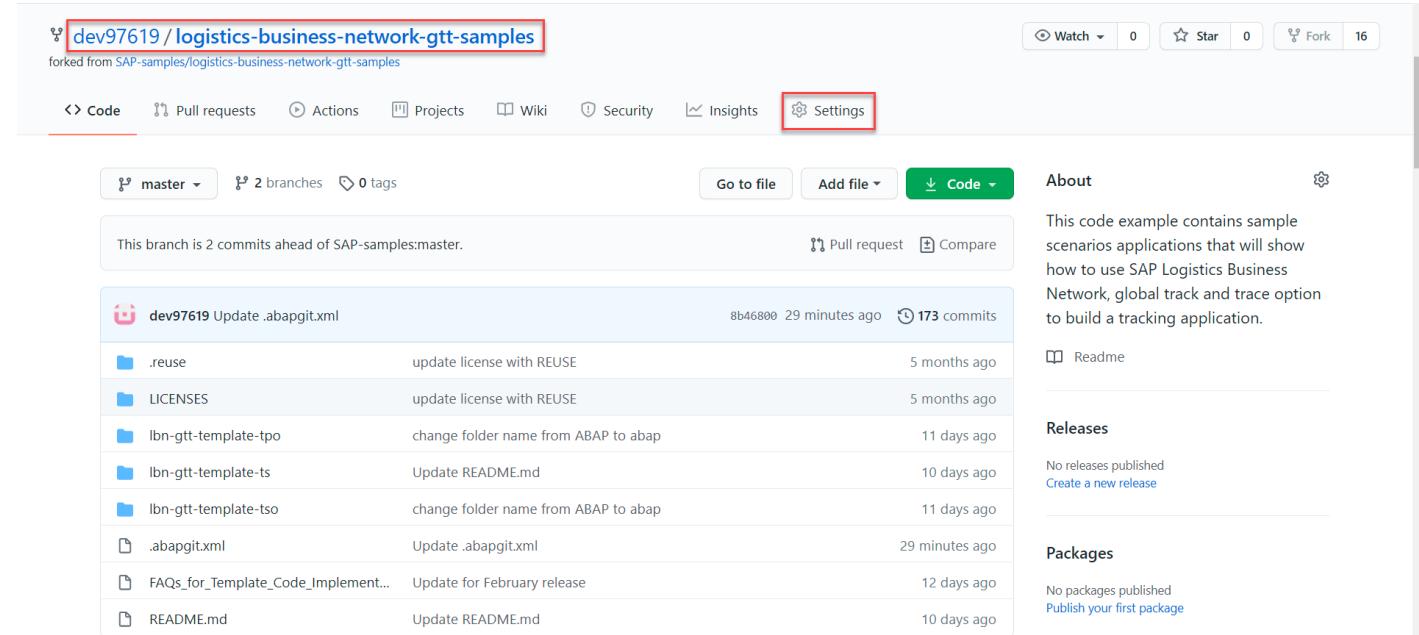


STEP 1: Delete the User's Account Repository

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

In the latest release, there is some code changes in the public sample code, you need to update the local code according to the latest public sample code.

1-2: Navigate to the user's account repository, and click "Settings".



STEP 1: Delete the User's Account Repository

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

The screenshot shows a GitHub repository settings page for 'github.com/dev97619/logistics-business-network-gtt-samples'. At the top, there is a 'Theme Chooser' section with a 'None' dropdown and a 'Save' button. Below it is a 'Danger Zone' section with four options: 'Change repository visibility', 'Transfer ownership', 'Archive this repository', and 'Delete this repository'. The 'Delete this repository' button is highlighted with a red border.

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

None Save

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

Choose a theme

Danger Zone

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

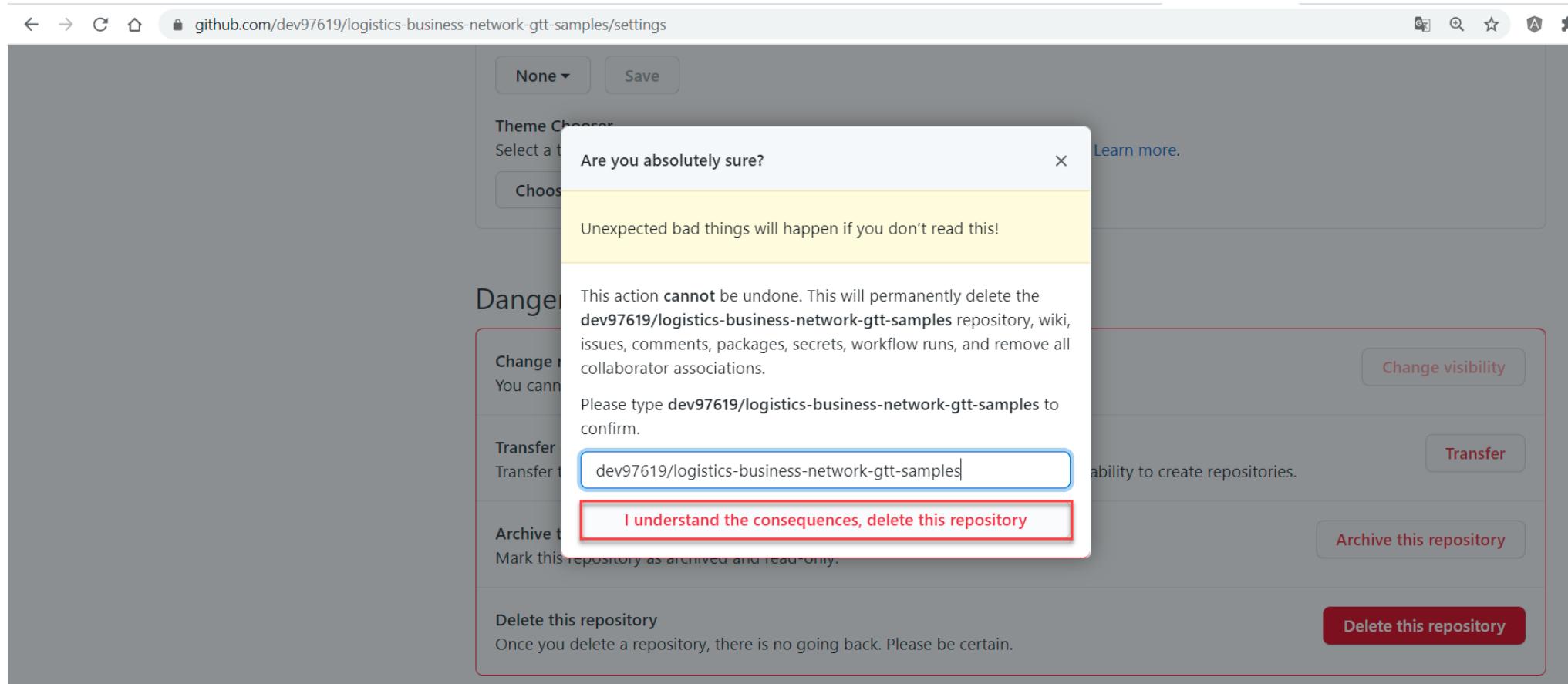
Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. [Transfer](#)

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

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

STEP 1: Delete the User's Account Repository

1-4: The popup shows some warning messages. Follow the instructions then click the button "I understand the consequences, delete this repository".



STEP 1: Delete the User's Account Repository

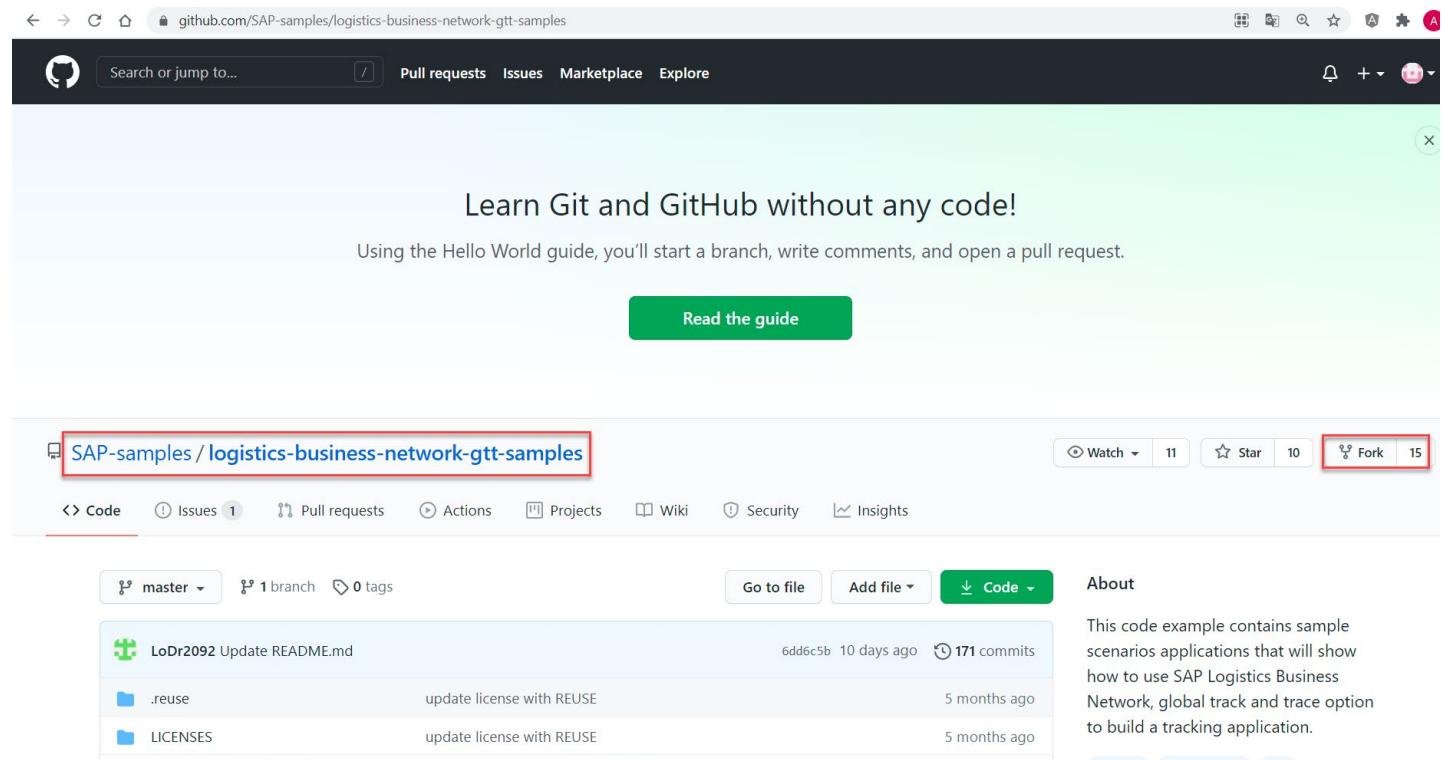
1-5: The user's account repository is deleted.

The screenshot shows a dark-themed GitHub interface. At the top, there is a navigation bar with a search bar, a pull requests button, an issues button, a marketplace button, and an explore button. On the right side of the top bar are icons for notifications, a plus sign, and a profile picture. Below the navigation bar, a message box contains the text "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." This message is highlighted with a red rectangular border. To the left of the message box, there is a "Create your first project" section with a "Create repository" button (which is green) and an "Import repository" button. Further down, there is a "Working with a team?" section with a "Create an organization" button. On the right side of the interface, there is a large, semi-transparent callout box with a light green background. The title of the callout is "Learn Git and GitHub without any code!". Below the title, it says "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." At the bottom of the callout, there are two buttons: a green "Read the guide" button and a white "Start a project" button.

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’

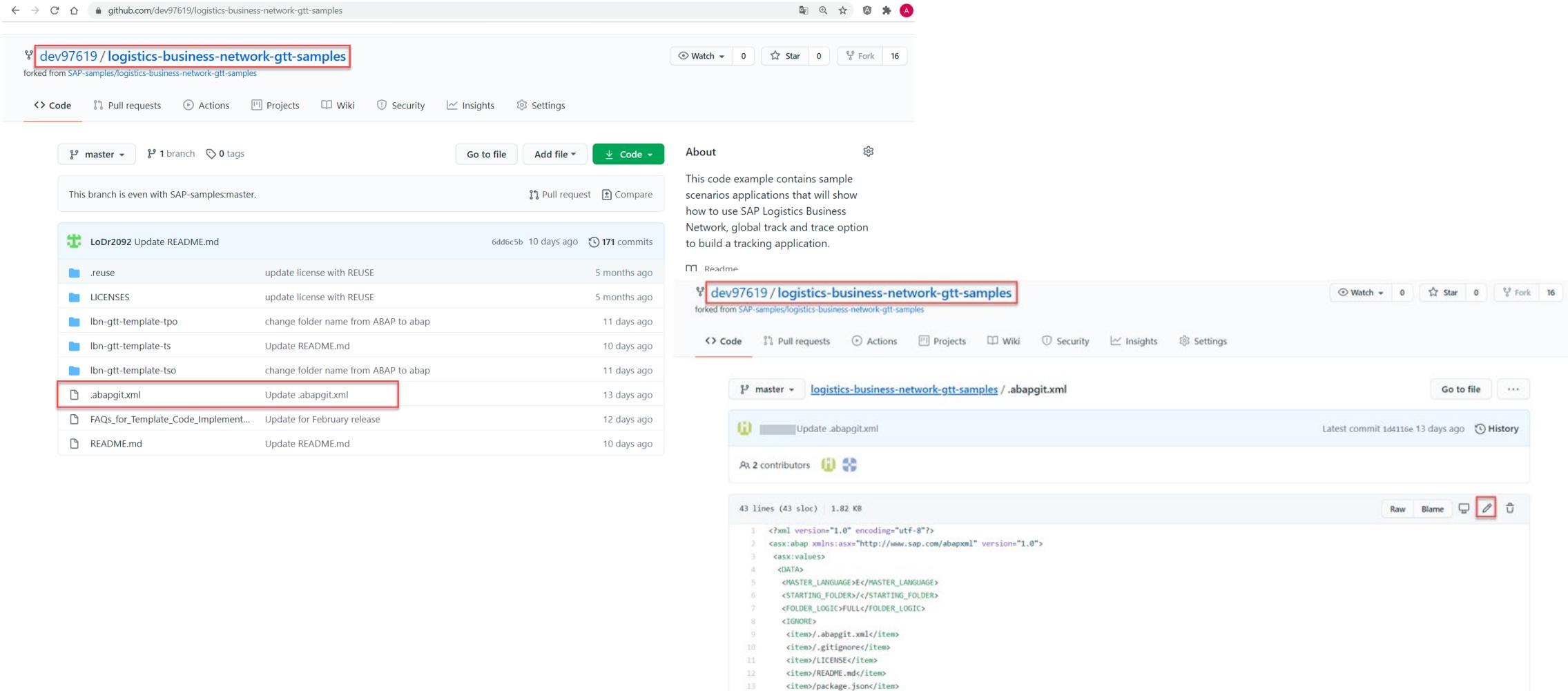
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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

Repository Page:

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

.abapgit.xml Content:

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

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

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

3-4: Commit change.

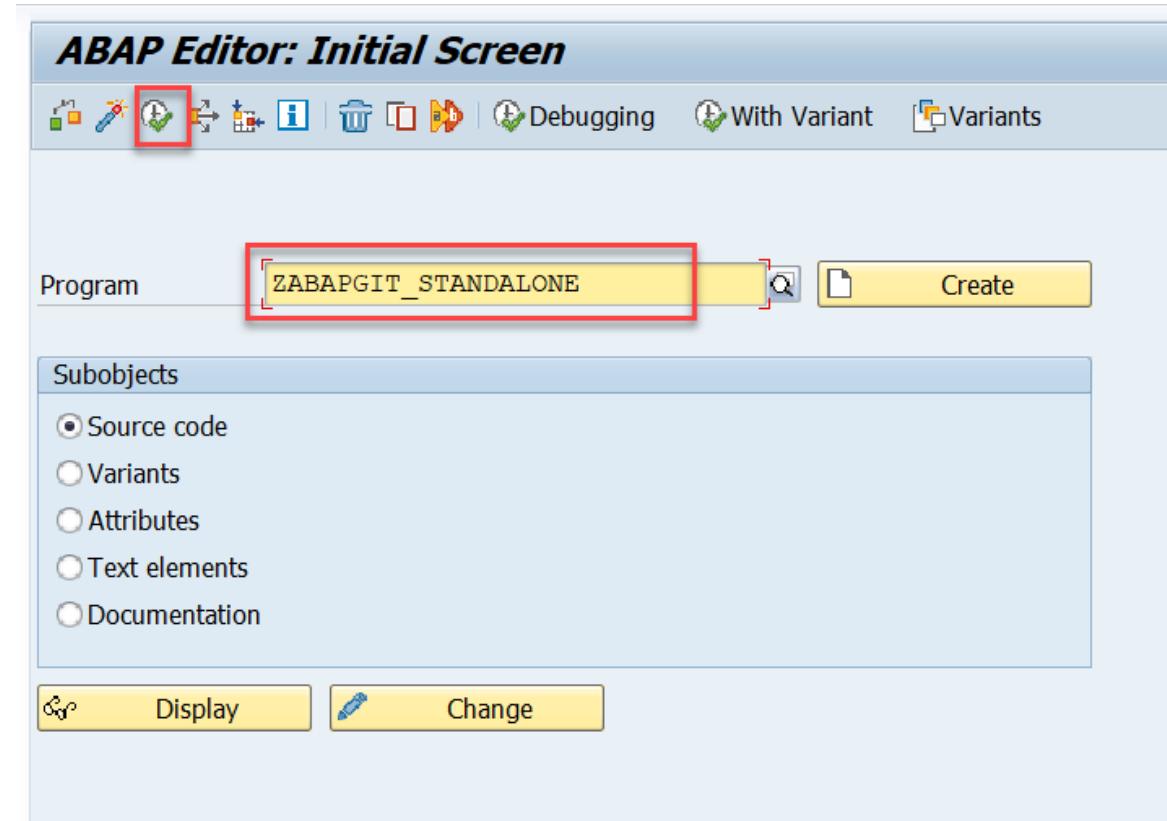
The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. In the code editor, the '.abapgit.xml' file is open, showing XML configuration. Line 6 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 has a 'Commit changes' button at the bottom left, which is also highlighted with a red box. The dialog includes fields for a commit message ('Update .abapgit.xml') and a description, and two radio button options for committing: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request.'

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

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.

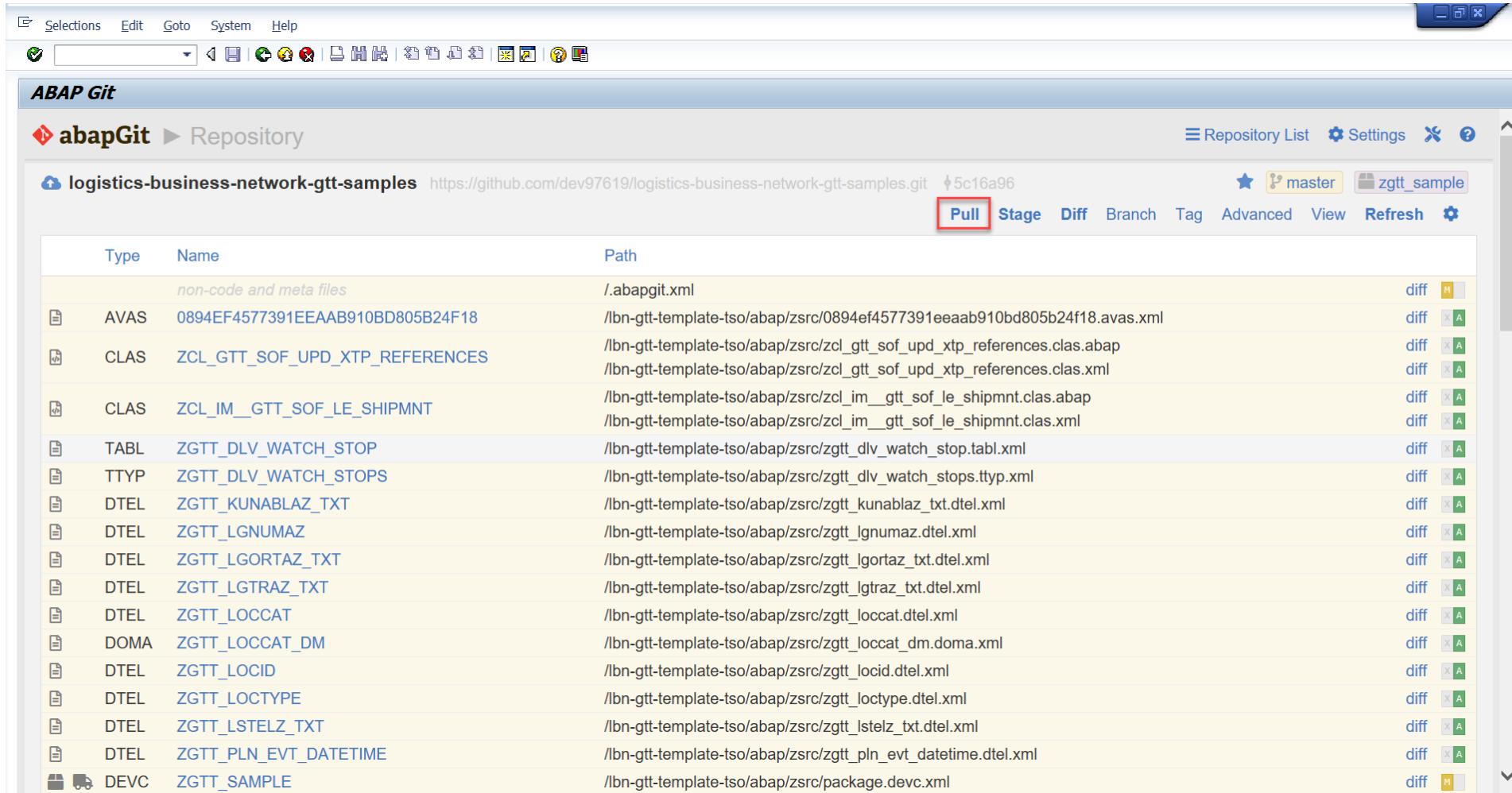


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons for file operations like Selections, Edit, Goto, System, and Help. Below the toolbar is a menu bar with 'ABAP Git' selected. The main area is titled 'abapGit' and 'Repository List'. It features a search bar with a placeholder 'Filter:' and checkboxes for 'Only Favorites' and 'Detail'. A table lists repositories with columns: Name, Url, Package, Branch, and Action. One repository is listed: 'logistics-business-network-gtt-samples' with Url 'github.com/dev97619/logistics-business-network-gtt-samples.git', Package 'zgtt_sample', Branch 'master', and Action buttons for 'Check', 'Stage', 'Patch', and 'Settings'. The 'Action' column for the last row has a red box around the rightmost button. At the bottom, there's a footer with the 'abapGit' logo and version '1.106.0', and a status message 'js: OK'.

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

STEP 4: Update ABAP Code from GitHub

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

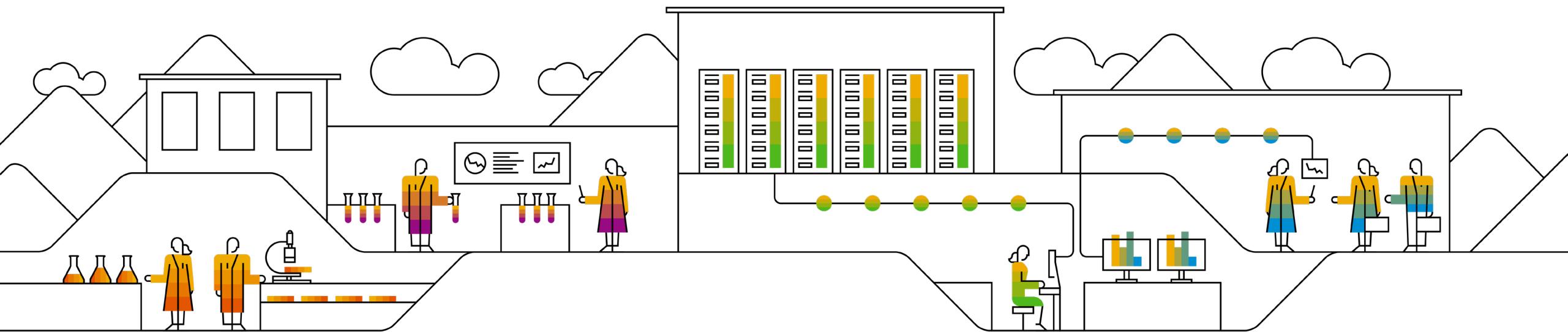


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git". Underneath, the repository path "abapGit > Repository" is shown, along with the URL "logistics-business-network-gtt-samples" and the commit hash "5c16a96". To the 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

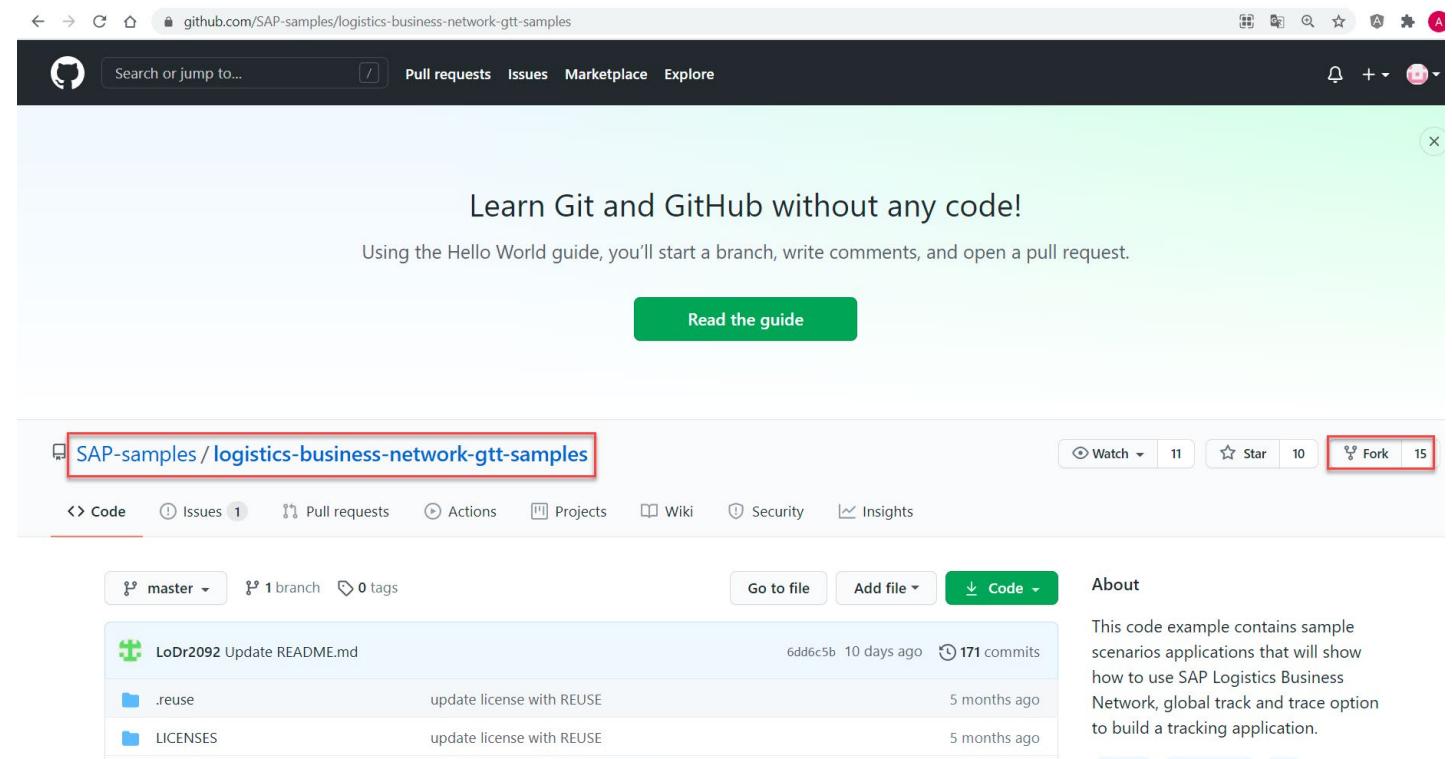
Prerequisite:

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’

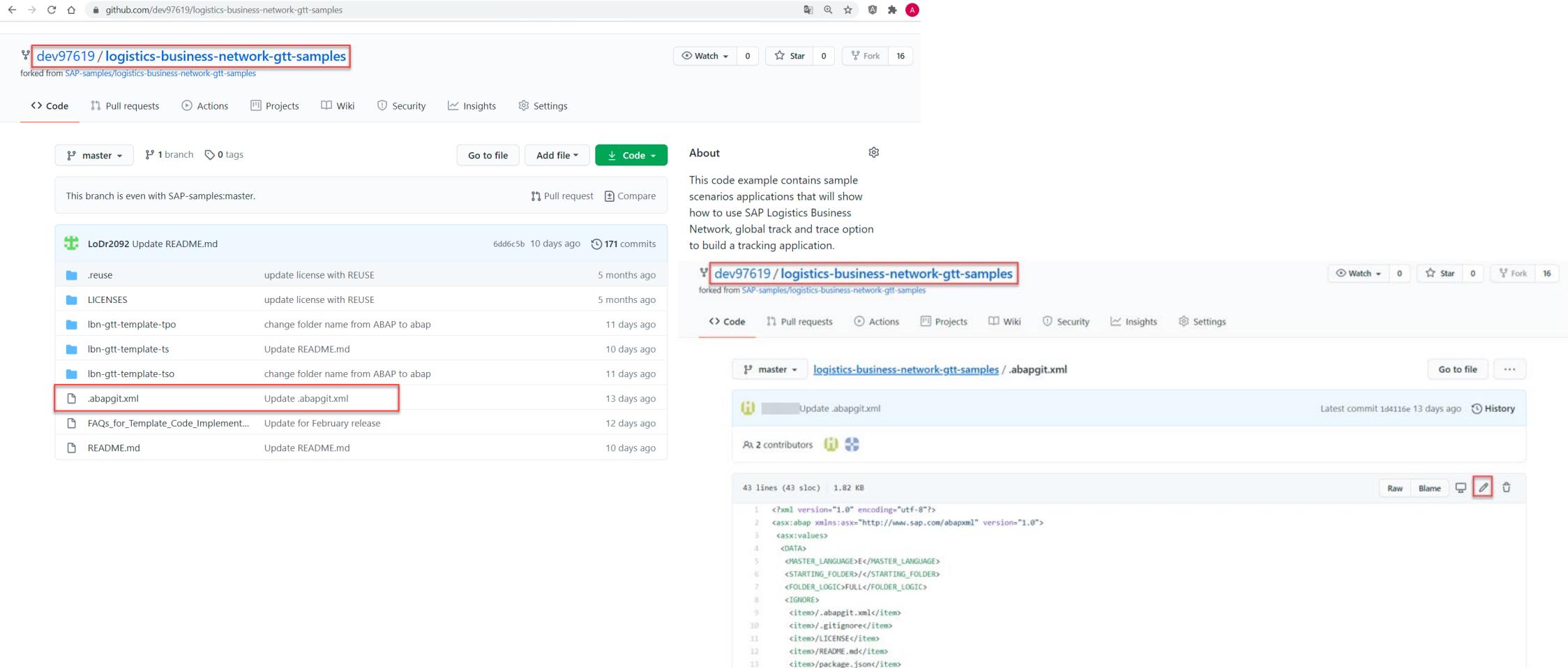
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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. A red box highlights the repository name 'dev97619 / logistics-business-network-gtt-samples' in the header and the '.abapgit.xml' file entry in the commit list. The commit list shows several recent changes, with the last commit being 'Update .abapgit.xml' by 'LoDr2092' 13 days ago. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages', each with a 'No [item] published' message and a 'Create a new [item]' or 'Publish your first [item]' link.

| File | Description | Time Ago |
|-------------------------------------|--------------------------------------|--------------|
| .abapgit.xml | Update .abapgit.xml | 13 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 |
| 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
- Content (partial XML code):

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

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

2-4: Commit change.

The screenshot shows a GitHub commit dialog for the file '.abapgit.xml' in the repository 'logistics-business-network-gtt-samples'. The file content is as follows:

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

A red box highlights the line '<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>'. The commit message field contains 'Update .abapgit.xml'.

Commit changes

Update .abapgit.xml

Add an optional extended description...

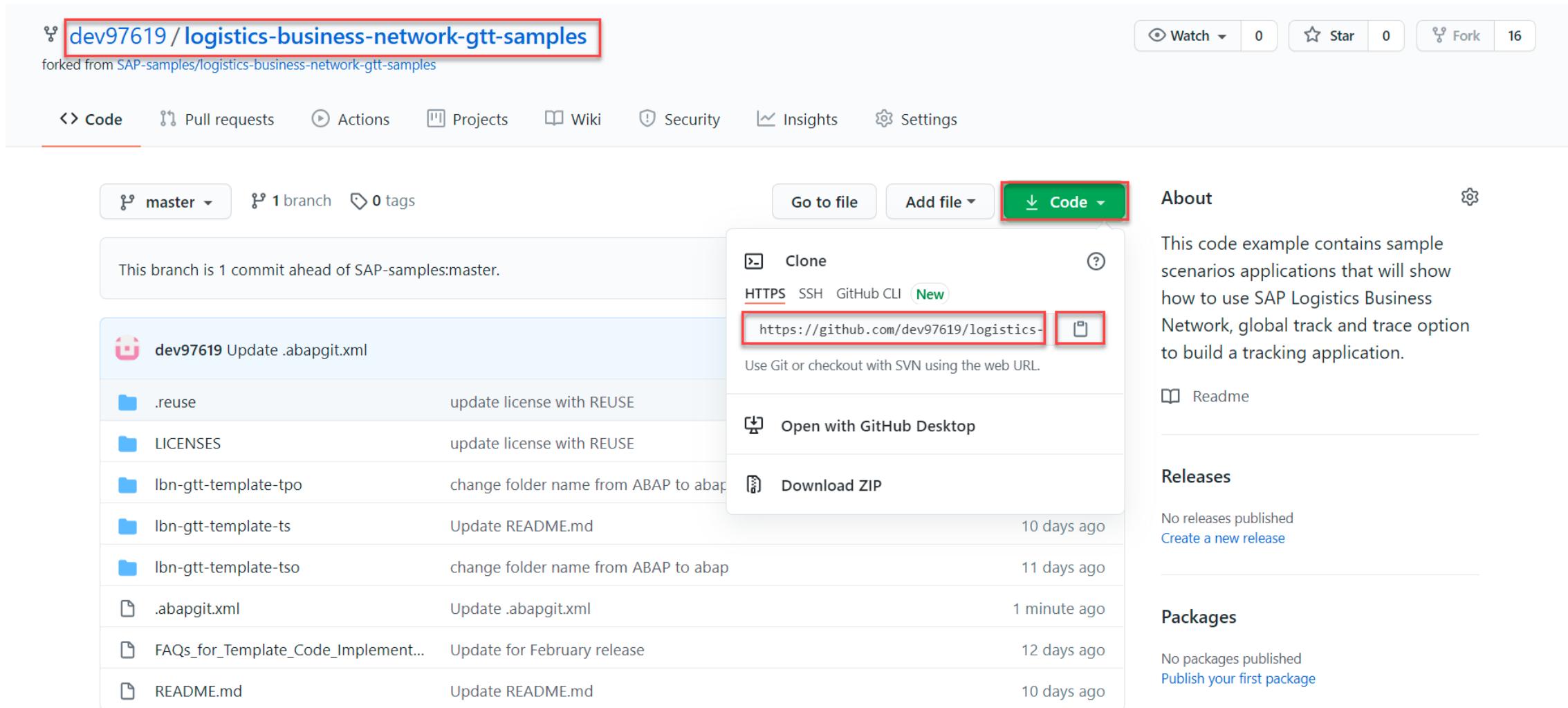
-o- Commit directly to the master branch.

⚡ 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 URL 'https://github.com/dev97619/logistics...' is highlighted with a red box, and a copy icon is also highlighted with a red box. The repository has 0 stars, 16 forks, and 0 issues. The 'About' section describes the repository as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options. The 'Readme' and 'Releases' sections are also visible.

Code example:

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

Repository details:

- Watch: 0
- Star: 0
- Fork: 16

Branches:

- master (selected)
- 1 branch
- 0 tags

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

Commits:

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

Code dropdown menu:

- Clone
- HTTPS
- SSH
- GitHub CLI
- New
- https://github.com/dev97619/logistics-business-network-gtt-samples
- Copy icon
- Open with GitHub Desktop
- Download ZIP

About:

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

Readme:

Releases:

No releases published
Create a new release

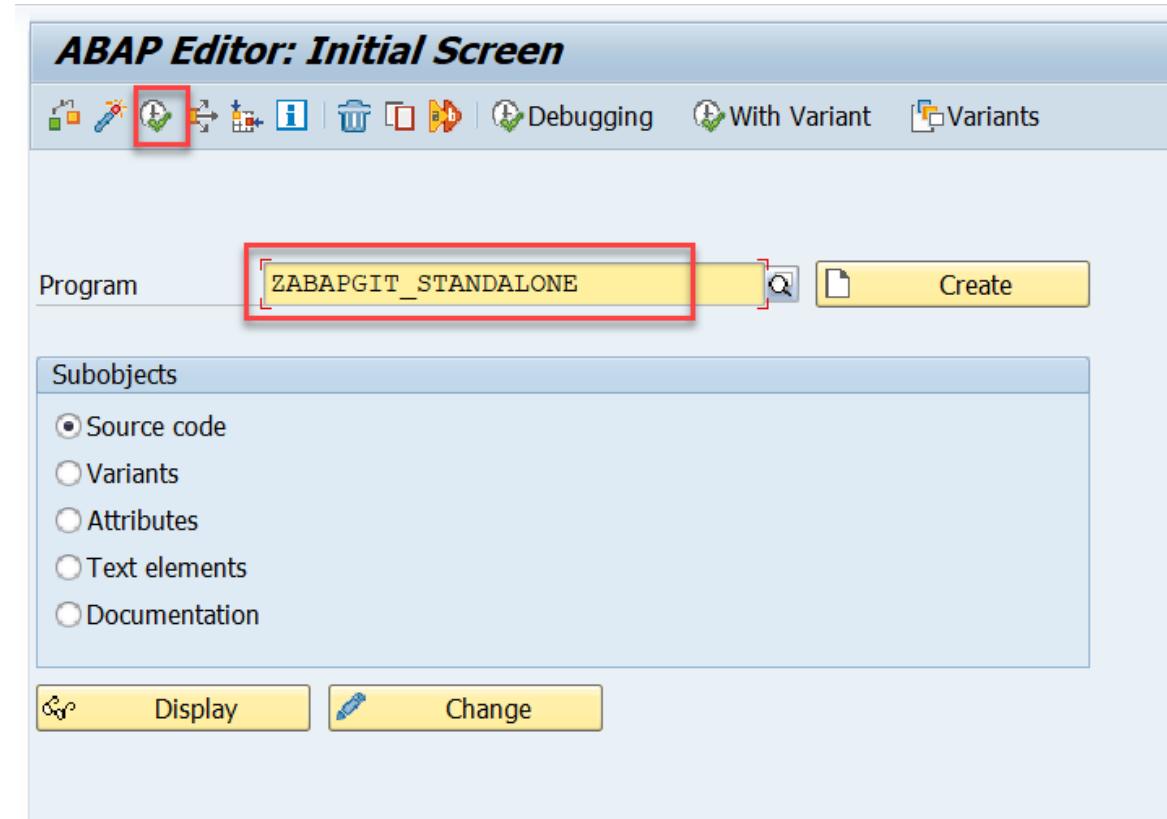
Packages:

No packages published
Publish your first package

STEP 3: Remove 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 ABAP Git interface with the following details:

- Toolbar:** Selections, Edit, Goto, System, Help.
- Repository List:** **Name:** logistics-business-network-gtt-samples, **Url:** github.com/dev97619/logistics-business-network-gtt-samples.git, **Package:** zgtt_sample, **Branch:** master, **Action:** Check | Stage | Patch | Settings | 
- Footer:** abapGit 1.106.0, js: OK

STEP 3: Remove TSOF Repository in ABAPGit

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

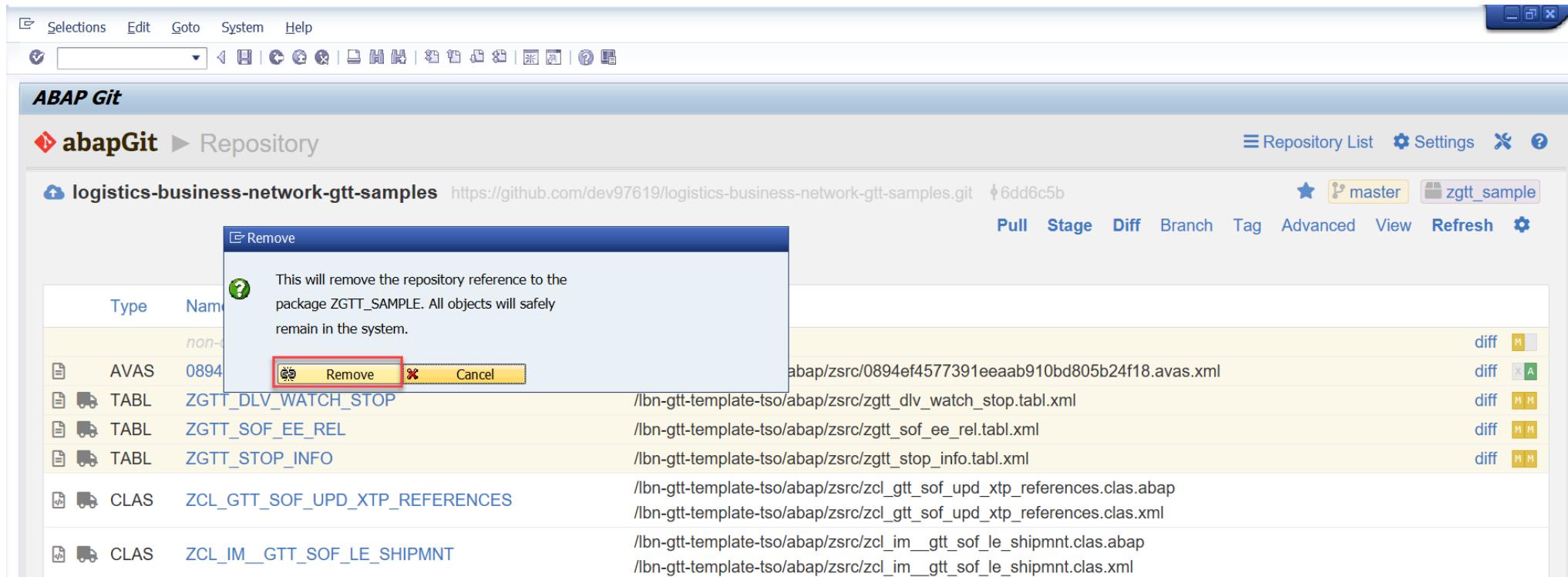
The screenshot shows the ABAPGit interface with the following details:

- Toolbar:** Selections, Edit, Goto, System, Help.
- Header:** ABAP Git, abapGit, Repository, Repository List, Settings, Refresh, Help.
- Repository Information:** logistics-business-network-gtt-samples, https://github.com/dev97619/logistics-business-network-gtt-samples.git, commit 6dd6c5b.
- Branch:** master (highlighted).
- Advanced Menu:** A dropdown menu with the following options:
 - Reset Local (Force Pull)
 - Checkout commit
 - Background Mode
 - Change Remote
 - Make Off-line
 - Force Stage
 - Transport to Branch
 - Add all objects to transport request
 - Syntax Check
 - Run Code Inspector
 - Update Local Checksums
 - Beta - Data
 - Remove** (highlighted with a red box)
 - Uninstall
- Table:** A list of repository contents with columns: Type, Name, Path.

| Type | Name | Path |
|-------------------------|---------------------------------|--|
| non-code and meta files | | |
| AVAS | 0894EF4577391EEAB910BD805B24F18 | ./abapgit.xml |
| TABL | ZGTT_DLV_WATCH_STOP | /lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeab910bd/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml |
| TABL | ZGTT_SOF_EE_REL | /lbn-gtt-template-tso/abap/zsrc/zggt_sof_ee_rel.tabl.xml |
| TABL | ZGTT_STOP_INFO | /lbn-gtt-template-tso/abap/zsrc/zggt_stop_info.tabl.xml |
| CLAS | ZCL_GTT_SOF_UPD_XTP_REFERENCES | /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_referen |
| CLAS | ZCL_IM_GTT_SOF_LE_SHIPMNT | /lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt |
| TTYP | ZGTT_DLV_WATCH_STOPS | /lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xr |
| DTEL | ZGTT_KUNABLAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_kunablaz_txt.dtel.xml |
| DTEL | ZGTT_LGNUMAZ | /lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml |
| DTEL | ZGTT_LGORTAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml |

STEP 3: Remove TSOF Repository in ABAPGit

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

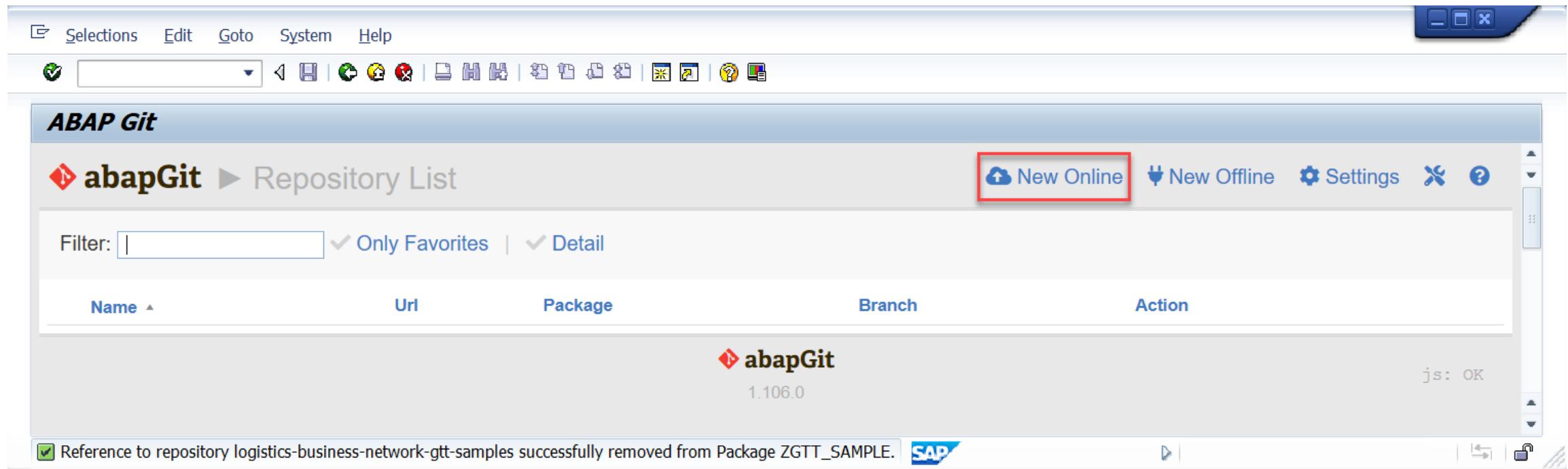


3-6: After repository removal you will see the following message:



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>

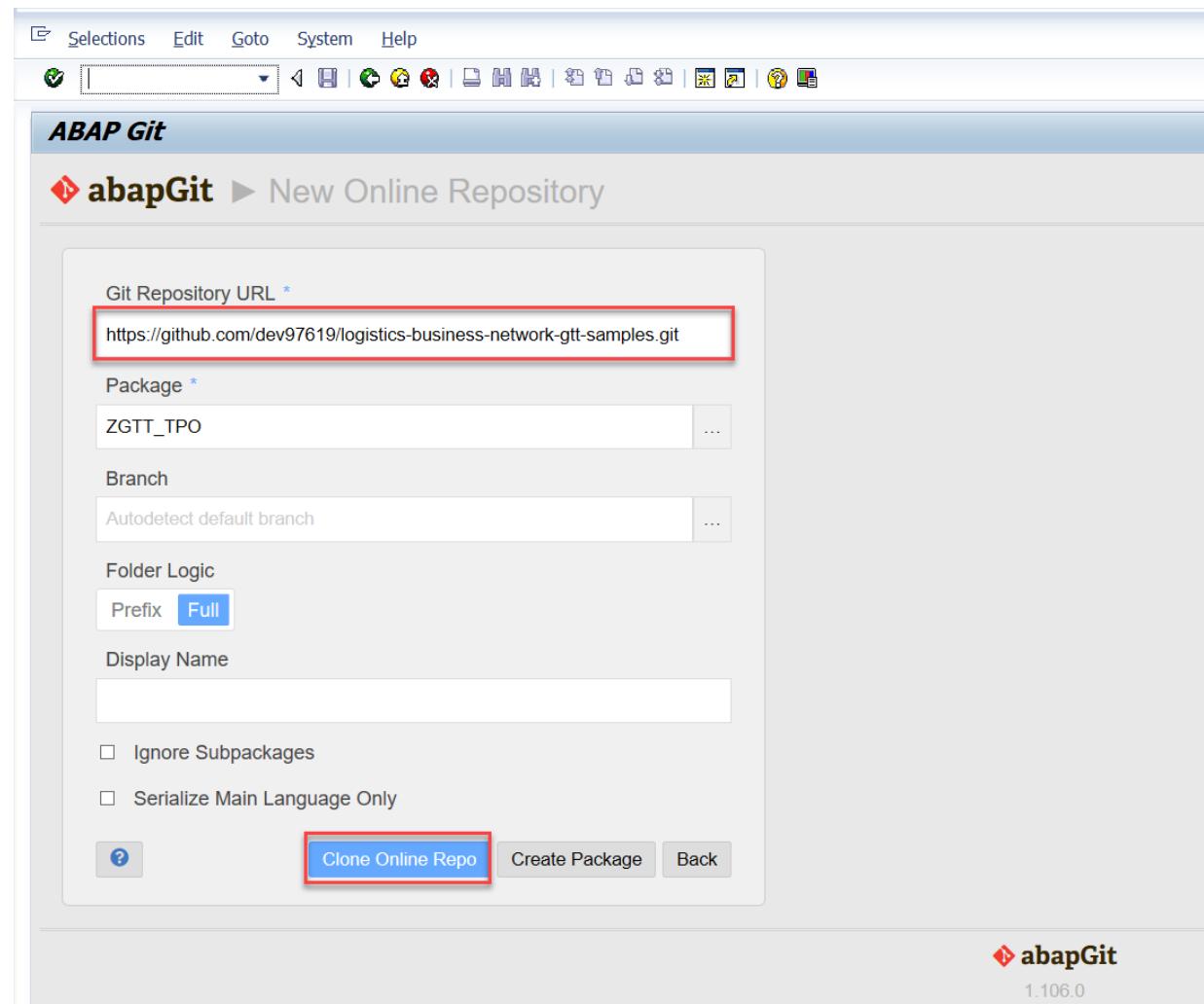
Caution:

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.

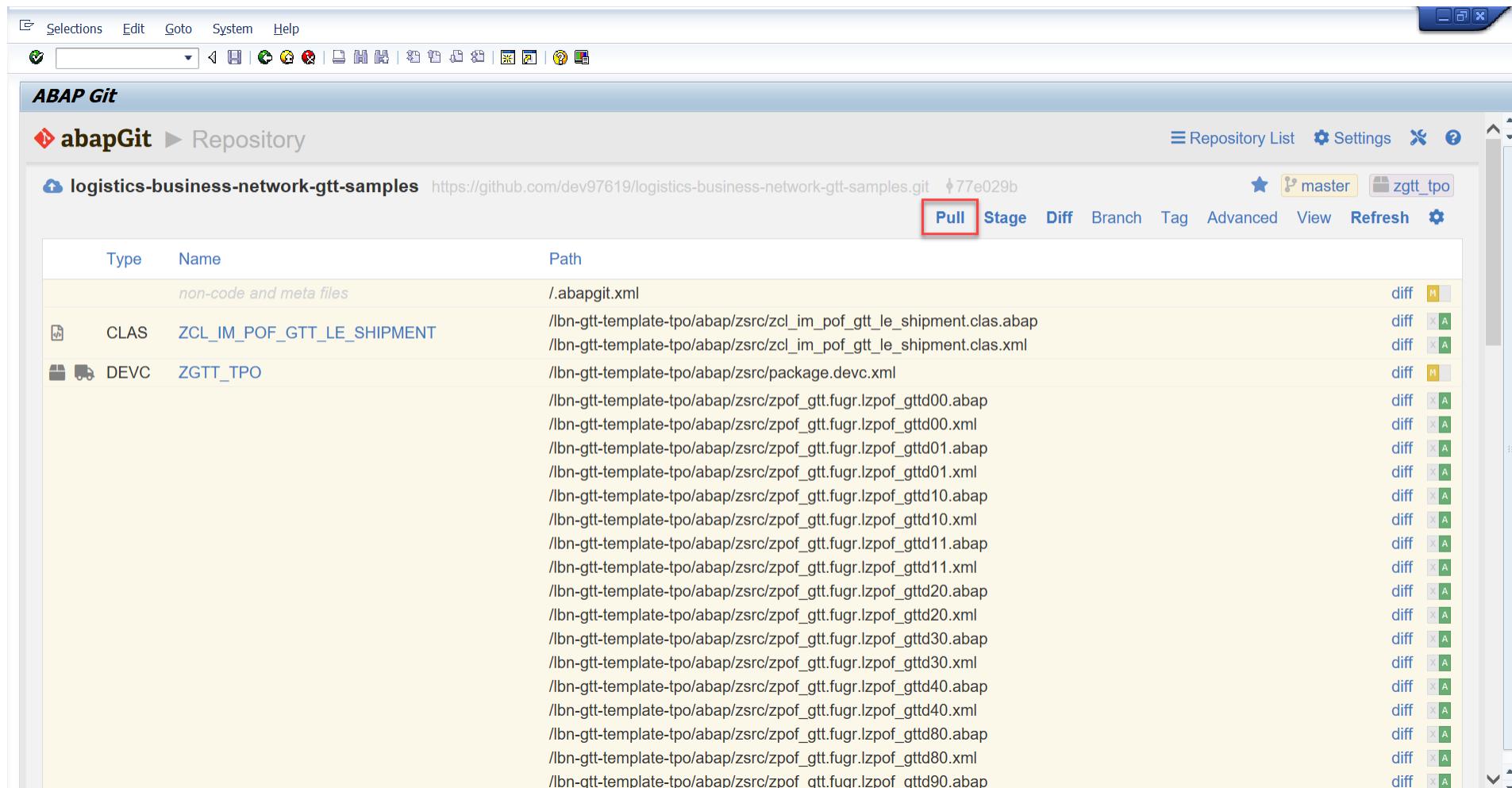
4-4: Set **Full** for **Folder Logic**

4-5: Click **Clone Online Repo** to download the code.



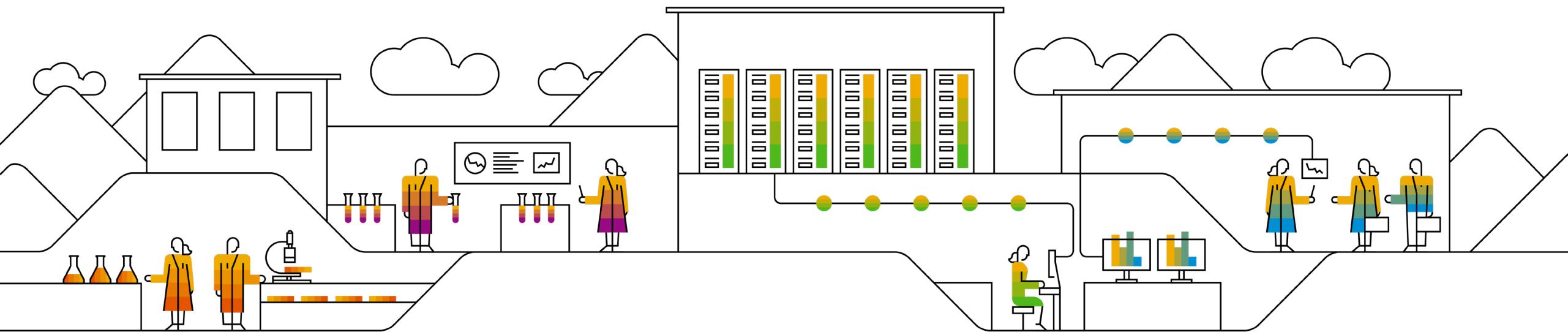
STEP 4: Download ABAP Code from GitHub

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



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.

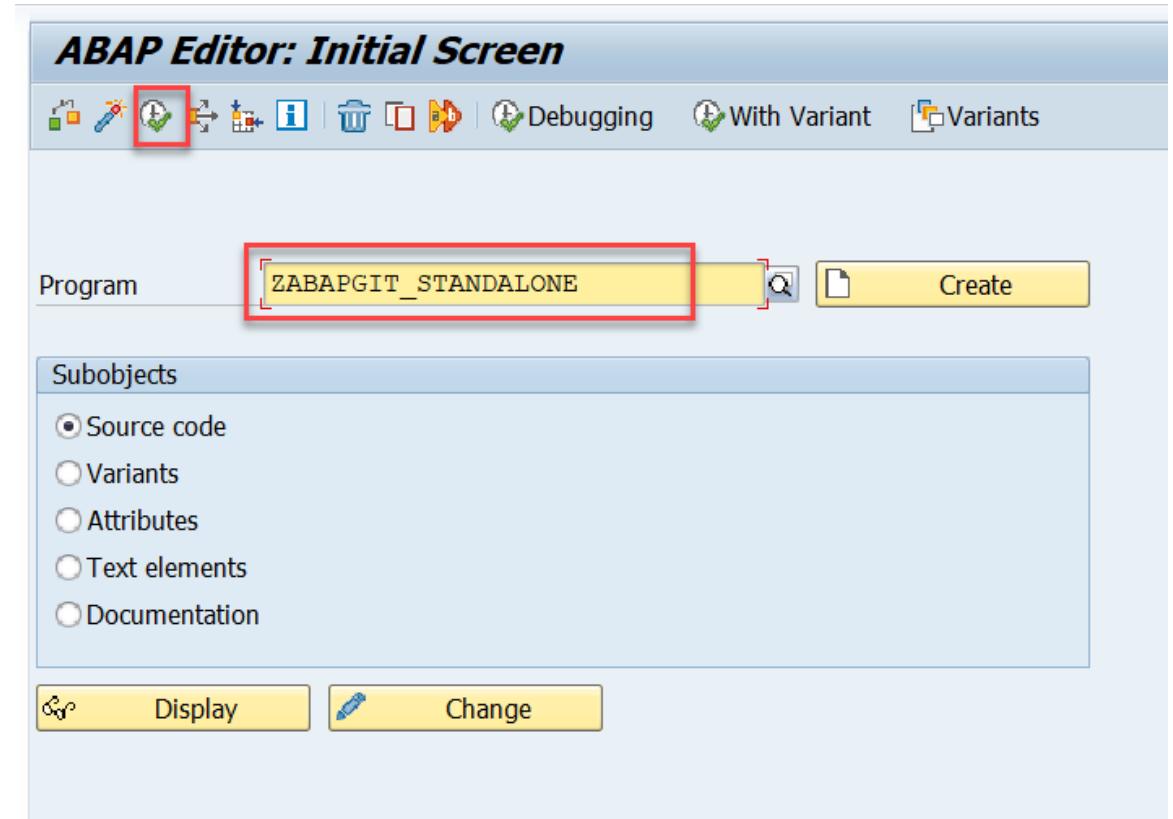
When installation is complete, a new report is created, **ZABAPGIT_STANDALONE**.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar contains links for "Getting Started" (Installation, Upgrading, Uninstalling, UI features), "Setup" (SSL setup, Proxy configuration, Development version), "Online Projects" (Installing online repo, Keeping code up to date, Uninstall repository, First project, Moving package into git, Contributing to a project), "Offline Projects" (Import zip, Export zip), and "Reference" (Repo Settings (abapgit.xml), Supported object types, Icon Legend, User Exits, Authorizations, Namespaces). The main content area starts with a "Summary" section stating that abapGit exists in two flavours: standalone or developer version. It then describes the standalone version as targeted at users and the developer version as targeted at developers contributing to the codebase. Below this is a "Prerequisites" section requiring SAP BASIS version 702 or higher. A red box highlights the "Install standalone version" section, which lists four 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 this, it notes that abapGit is typically used in development systems and can be installed in a local \$ package. A final note says "Now you can use abapGit by executing the report in transaction SE38".

STEP 2: Download ABAP Code

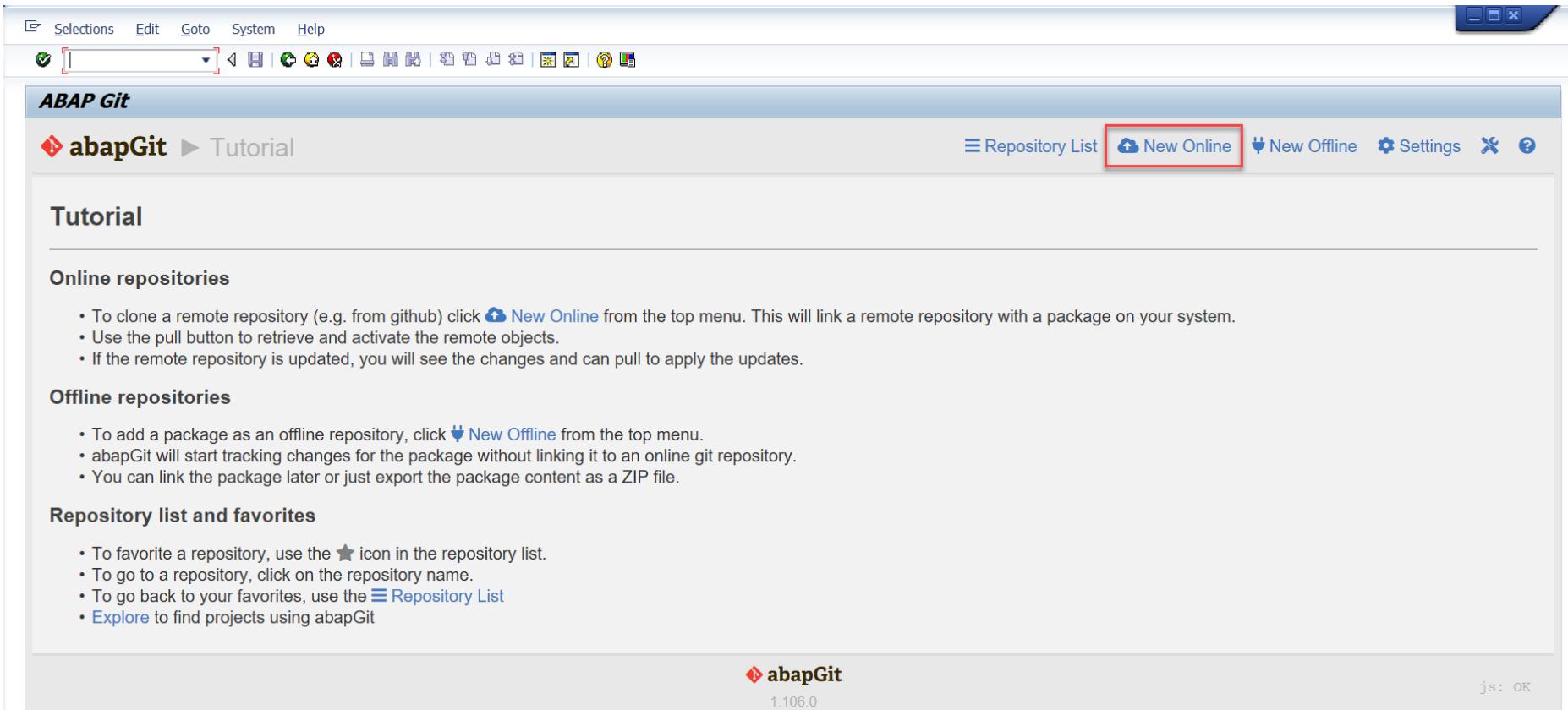
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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

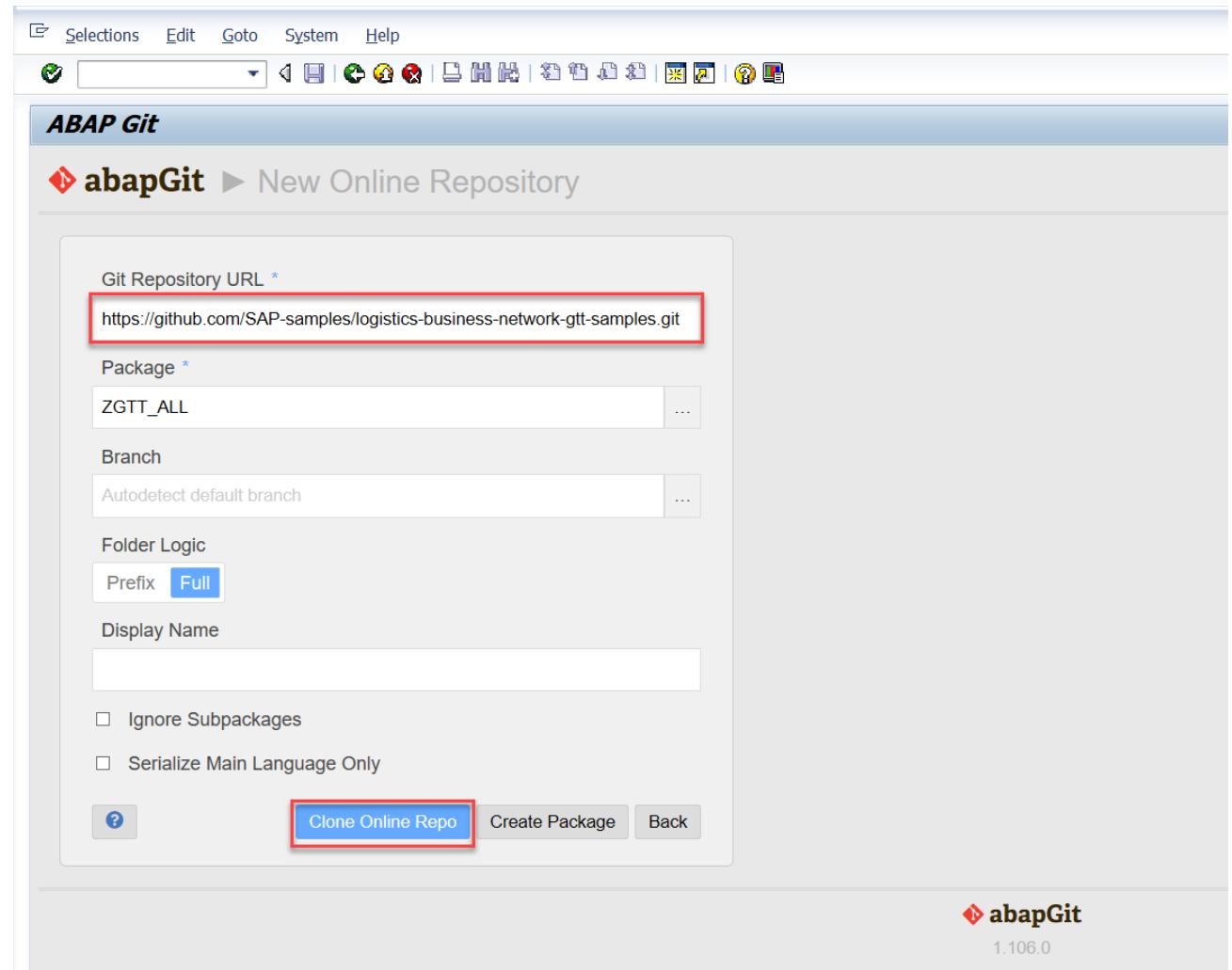
2-4: Fill in the **Git Repository URL**:

<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.

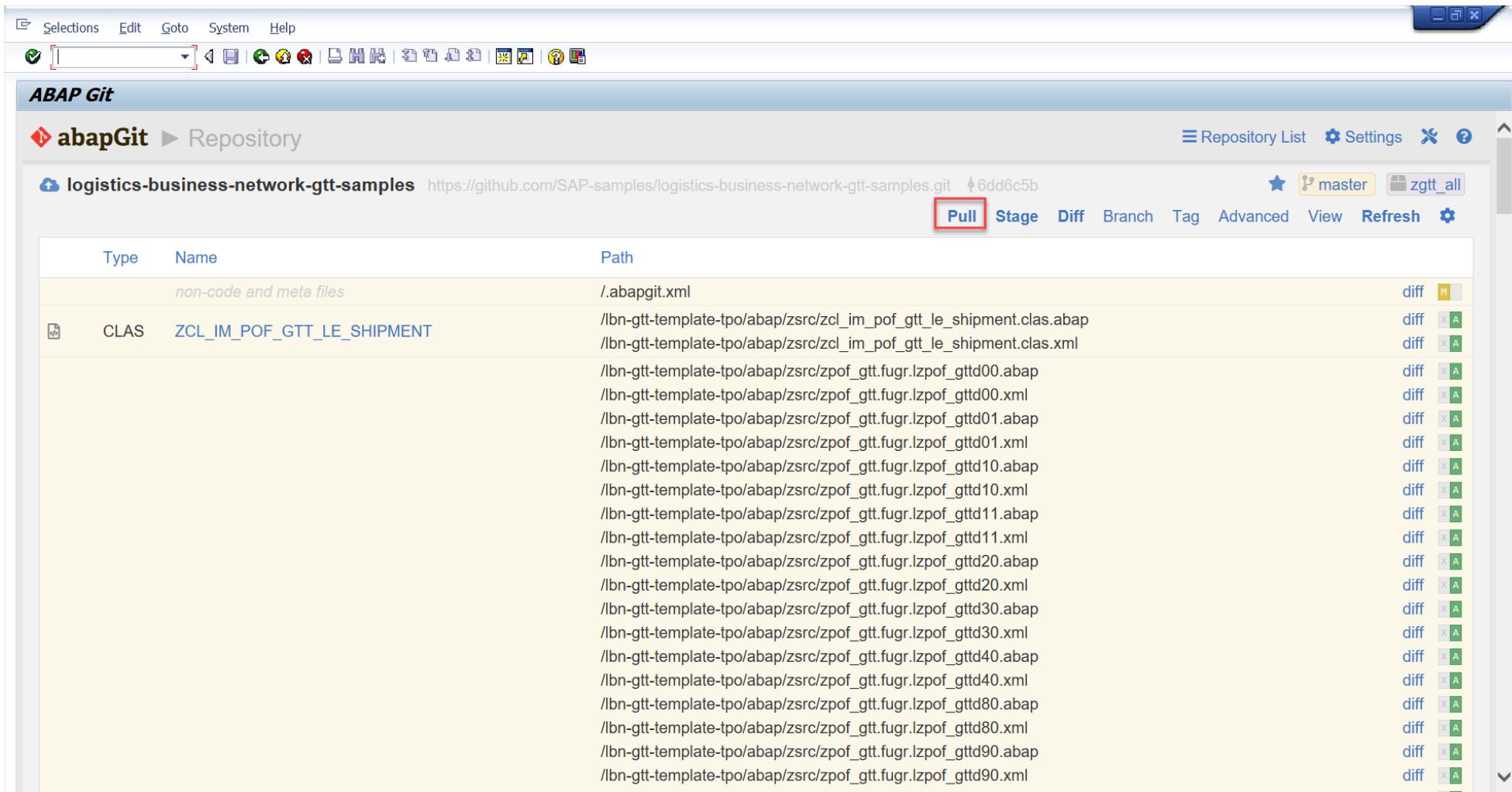
2-6: Set *Full* for **Folder Logic**

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.

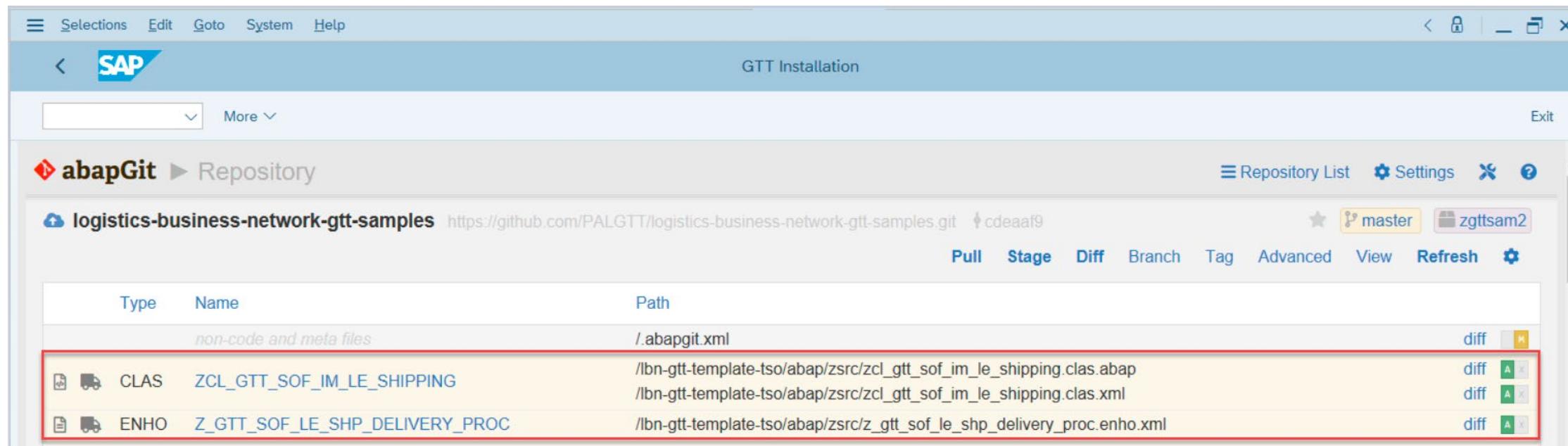


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the path 'abapGit > Repository'. A repository card for 'logistics-business-network-gtt-samples' 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 'Path' column lists numerous ABAP and XML files under the directory '/bn-gtt-template-tpo/abap/zsrc/'. The 'diff' column shows status indicators for each file.

| 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 | diff |
| | | /bn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt00.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt01.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt10.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt11.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt20.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt30.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt40.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt80.xml | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.abap | A |
| | | /bn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpos_gtt90.xml | A |

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

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



The screenshot shows the SAP GTT Installation interface. At the top, there's a navigation bar with 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below it is a SAP logo and the title 'GTT Installation'. The main area is titled 'abapGit' and 'Repository'. It displays a GitHub repository named 'logistics-business-network-gtt-samples' with the URL <https://github.com/PALGTT/logistics-business-network-gtt-samples.git>. The commit hash is 'cdeaaaf'. A yellow box highlights the 'master' branch. Below the repository info is a toolbar with 'Pull', 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a settings icon.

| 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_sof_im_le_shipping.clas.abap /Ibn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_im_le_shipping.clas.xml | [diff icon] [diff icon] |
| ENHO | Z_GTT_SOFTWARE_SHIPPING_DELIVERY_PROC | /Ibn-gtt-template-tso/abap/zsrc/z_gtt_sof_im_le_shp_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_SHIP_DELIVERY_PROC

2-2) Delete the BADI implementation class ZCL_GTT_SOF_IM_LE_SHIPPING

Notes:

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

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

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

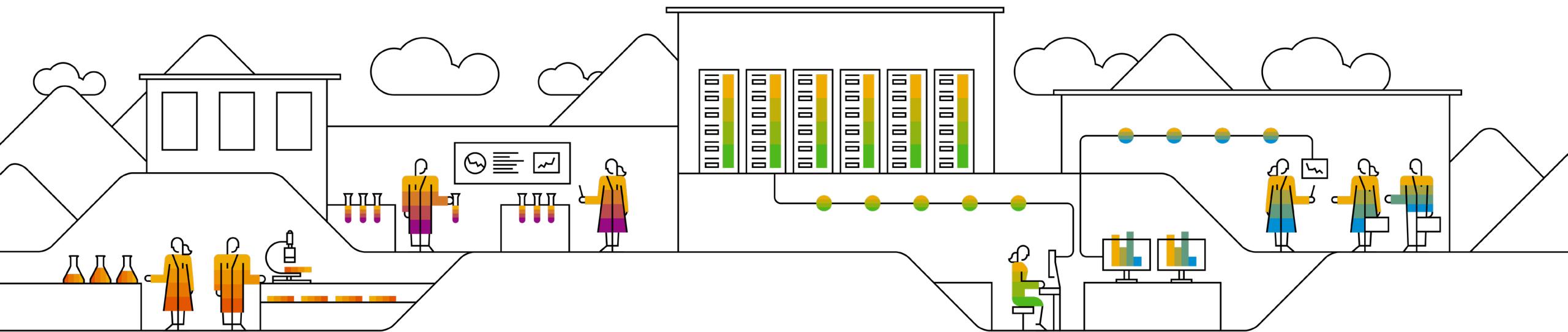
For option 1: Use transaction code SE19 and deactivate the BADI implementation.

The screenshot shows the SAP BAdI Builder interface. The main title bar reads "BAdI Builder: Initial Screen for Implementations". Below it, there's a toolbar with buttons for "Check", "Delete implementation", "Copy implementation", "Rename implementation", "Application help", and "More". The main area is titled "Edit Implementation" and has a radio button selected for "New BAdI". An input field labeled "Enhancement implementation:" contains the value "Z_GTT_SOF_LE_SHP_DELIVERY_PROC", which is highlighted with a red box. Below this, there's another radio button for "Classic BAdI" and an input field for "Implementation:". At the bottom of this section are two buttons: "Display" and "Change".

A secondary window titled "Enhancement Implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC Display" is open. It shows the same enhancement implementation "Z_GTT_SOF_LE_SHP_DELIVERY_PROC". The "Properties" tab is selected, showing the "BAdI Implementation" as "Z_GTT_SOF_IM_LE_SHIPPING" and the "Description" as "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list". There are checkboxes for "Default Implementation", "Example Implementation", and a checked checkbox for "Active". Under "Runtime Behavior", there's a checked checkbox for "Implementation is active" and a note below it stating "Runtime Behavior: The implementation will not be called".

D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

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

The image displays two screenshots of the SAP Business Network Global Track and Trace Version 2 interface, specifically focusing on the "Define Application Object Types" screen.

Screenshot 1: Define Application Object Types - Details

- Dialog Structure:** Shows the selected path: Define Used Business Process > Define Application Object Type.
- General Data:** Bus. Proc. Type: **ESC_SORDER**, Appl. Obj. Type: **ZGTT_SO_INT_HD** (highlighted with a red box). Extract sales order header information to Global Track and Trace Integration, Text: Sales Order Header.
- Object Identification:** Method for determination of AOID: Determine from Field, First Field to Build Appl. Obj. ID: Cntrl Tab. Type: 1 Main Object Table, AO ID Field: **VBELN**.
- Global Track & Trace Relevance:** Not visible in this screenshot.
- Parameter Setup:** Not visible in this screenshot.

Screenshot 2: IDOC Integration Configuration

- IDOC Integration:** Tracked Process: SalesOrder, Integration Switch: ON (highlighted with a red box).
- Tracked Process Mapping:** ERP Object Type: Others, Application Object Type: **ZGTT_SO_INT_HD** (highlighted with a red box).
- Tracked Process / Events (2):** SalesOrderEvent (Name), E1EHPAO (IDOC), E1EHPAO (Event Code).
- User Model Fields:** A table mapping fields to IDOC segments and fields.

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

2: Maintain Tracking ID Type

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

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

The image shows two screenshots illustrating the configuration of Tracking ID Types. On the left, the SAP AOT interface displays a process named 'ESC_SORDER' with a tracking ID setup. The 'Tr.ID Code Set' field is highlighted with a red box and contains the value 'SALES_ORDER'. On the right, the SAP Business Network Global Track and Trace interface shows a tracked process named 'Sales Order Fulfillment'. It lists several items, including 'SalesOrder' and 'SalesOrderItem', both of which have their 'Tracking Id Type' fields highlighted with red boxes and set to 'SALES_ORDER'. A modal dialog box is also shown, titled 'Edit Tracked Process', with the 'Name' field set to 'SalesOrder' and the 'Tracking Id Type' field set to 'SALES_ORDER', also highlighted with a red box.

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:

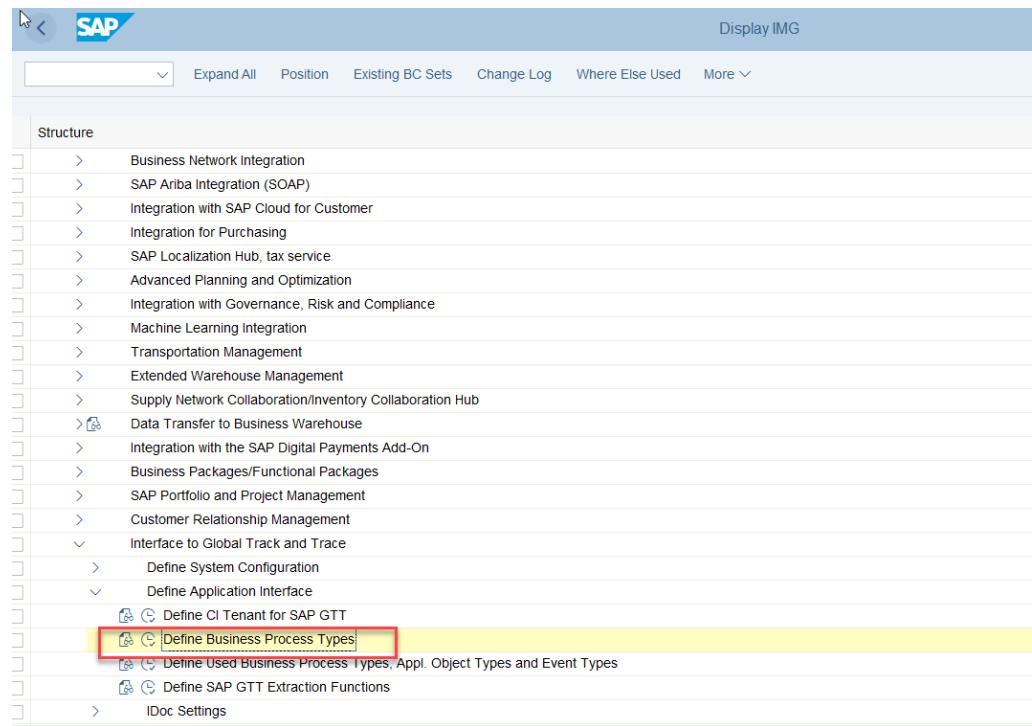
| Category | Business Process Type | Function Module Name | Description |
|--------------------------------------|-----------------------|--------------------------------|---|
| Control Parameter Extractors | ESC_DELIV | ZGTT_SOF_OTE_DE_HD | Function for setup of control parameters of delivery header |
| 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

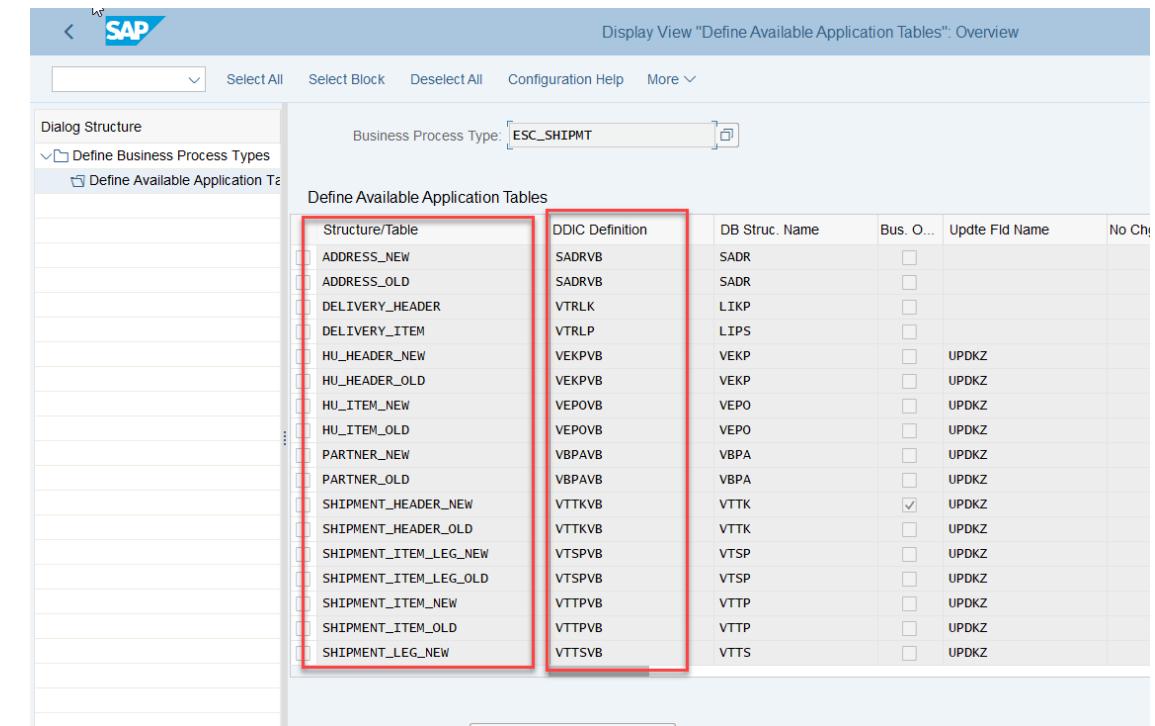
5-1: In Display IMG page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

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 content area is titled 'Structure' and lists various SAP 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 main content area is titled 'Dialog Structure' and shows a list of 'Define Available Application Tables'. A search bar at the top right is set to 'ESC_SHIPMT'. Two specific columns are highlighted with red boxes: 'Structure/Table' and 'DDIC Definition'. The 'Structure/Table' column lists table names like ADDRESS_NEW, ADDRESS_OLD, DELIVERY_HEADER, etc. The 'DDIC Definition' column lists corresponding table names like SADR, SADR, VTRLK, etc. A table structure is also visible on the right side of the screen.

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

Function Builder: Display ZGTT_SOF_OTE_SHP_HDR_REL

Function Module: ZGTT_SOF_OTE_SHP_HDR_REL active

Attributes Import Export Changing Tables Exceptions Source Code

```
28:     <ls_xvttk>      TYPE vttkvb.  
29:  
30:     * <1> Check if Main table is Shipment or not.  
31:     IF i_app_object-maintabdef <> gc_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-trrelfunc  
37:                     i_app_object-appobjtype  
38:                     i_appsys.  
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:  
46:     * <3> Check Relevance of AOT: YN_OTE  
47:     PERFORM check_act_relevance_shp  
48:         USING <ls_xvttk>  
49:             CHANGING lv_act_relevance.  
50:     CHECK lv_act_relevance IS NOT INITIAL.  
51:  
52:     When shipment is newly created, check relevance of GTT: only when delivery has been assigned.  
53:     IF <ls_xvttk>-updtk EQ gc_insert.  
54:         PERFORM check_delivery_assignment  
55:             USING i_all_appl_cables  
56:  
Scope: FUNCTION zgtt_sof_ote_shp_hdr_rel | ABAP | Ln 18 Col 50 | | | | |
```

7: Coding Tips in the Tracking ID function modules

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

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

See sample code of function:

ZGTT_ADD_TRACKID_OTE_SHPHDR

SAP Function Builder: Display ZGTT_ADD_TRACKID_OTE_SHPHDR

Display <> Change Other Object... Enhance Check Activate Test/Execute Where...

Function Module: ZGTT_ADD_TRACKID_OTE_SHPHDR active

Attributes Import Export Changing Tables Exceptions Source Code

```
76 e_trackiddata-trxcod = 'SHIPMENT_ORDER'.
77 e_trackiddata-trxid = <ls_xvttk>-tnum.
78 CONCATENATE '0' sy-datum sy-zeitat INTO e_trackiddata-start_date.
79 e_trackiddata-end_date = '099991231000000'.
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>-tnum <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>-tnum <ls_xvttk>-signi INTO e_trackiddata-trxid.
96     e_trackiddata-msrid = space.
97     APPEND e_trackiddata.
98   ENDIF.
99 ELSEIF <ls_xvttk>-updckz = 'U'.
100  READ TABLE lt_yvttl INTO ls_yvttl INDEX 1.
101  CHECK ls_yvttl IS NOT INITIAL.
102
103  IF <ls_xvttk>-vsart <> ls_yvttl-vsart.
104    IF <ls_xvttk>-vsart = '01' AND <ls_xvttk>-extil IS NOT INITIAL.
```

8: Coding Tips in the Control Parameter function modules

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

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

See sample code of function: *ZGTT_SOF_OTE_SHP_HD*

8: Coding Tips in the Control Parameter Function Modules

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

The screenshot shows the SAP Manage Models app interface. The top navigation bar includes the SAP logo, Model Details (with a dropdown menu), Internal - Test, Help, and User icons. The current model is 'sof' (Active). The page title is 'Sales Order Fulfillment'. The main navigation tabs are Tracked Process, Field Type Pool, Event Type Pool, Code List, **IDOC Integration** (which is selected), Visibility Provider Integration, Planned Event Extension, and Event to Action.

Under the IDOC Integration tab, there are two input fields: 'Tracked Process:' (set to 'Shipment') and 'Integration Switch:' (set to 'ON').

The 'Tracked Process Mapping' section shows the mapping between ERP Object Type ('Others') and Application Object Type ('ZGTT_SHP_INT_HD').

The 'Tracked Process / Events (26)' table lists tracked processes and their corresponding IDOC segments and event codes. The rows include:

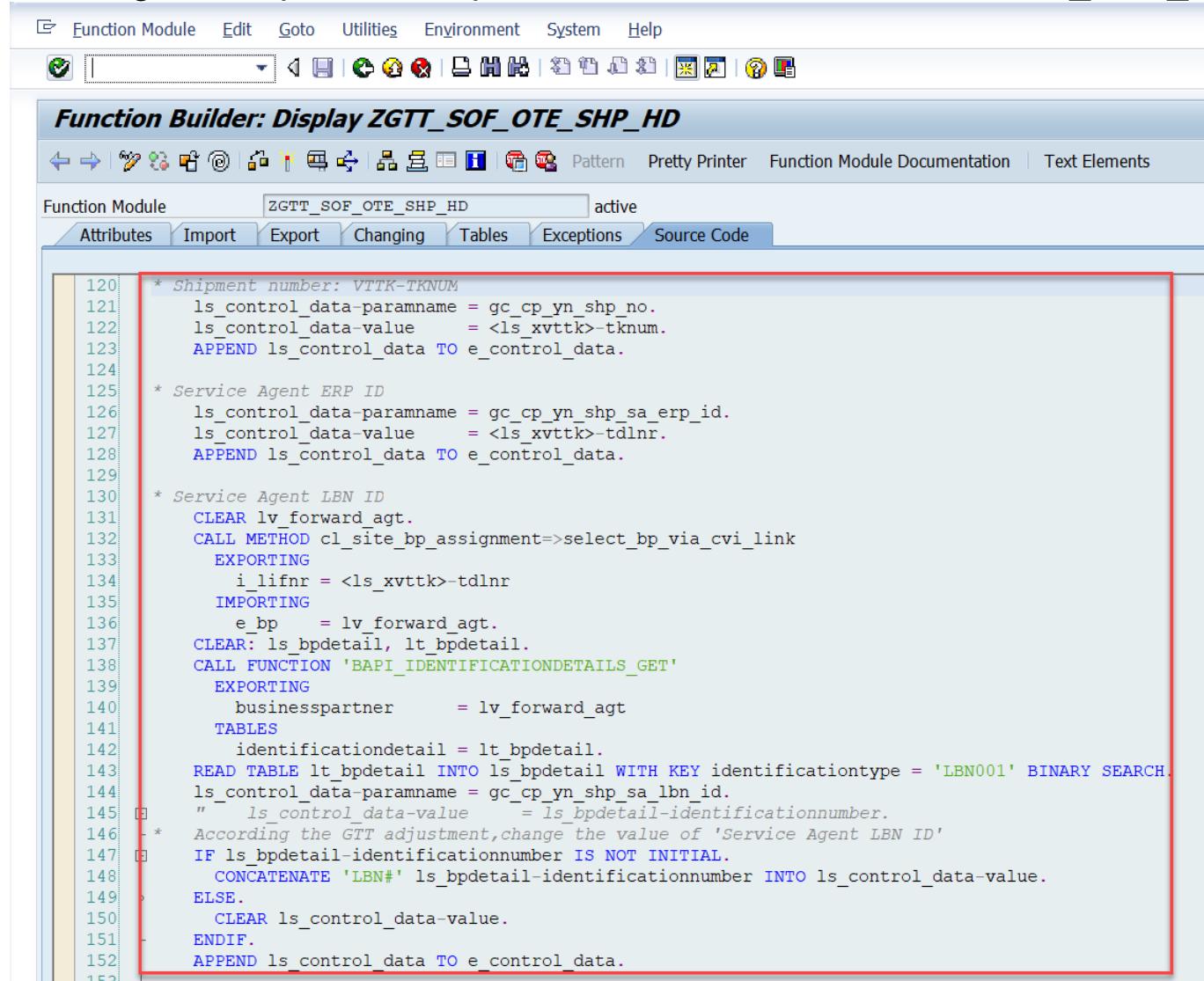
| Name | IDOC | Event Code |
|------------------------|------------|------------|
| Tracked Process | | |
| ShipmentEvent | E1EHPAO | |
| Event Types | | |
| LoadingStart | E1EVMHDR02 | LOAD_BEGIN |
| POD | E1EVMHDR02 | POD |
| Departure | E1EVMHDR02 | DEPARTURE |
| Arrival | E1EVMHDR02 | ARRIV_DEST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

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

| Field | IDOC Segment | IDOC Field |
|---------------------------|--------------|----------------------------|
| shipmentNo | E1EHPCP | YN_SHP_NO |
| serviceAgentLbNId | E1EHPCP | YN_SHP_SA_LBN_ID |
| transportationMode | E1EHPCP | YN_SHP_TRANSPORTATION_MODE |
| dangerousGoods | E1EHPCP | YN_SHP_CONTAIN_DGOODS |
| forwardingAgentTrackingId | E1EHPCP | YN_SHP_FA_TRACKING_ID |
| > stops | | |
| shippingType | E1EHPCP | YN_SHP_SHIPPING_TYPE |
| > resourceTPs | | |

8: Coding Tips in the Control Parameter Function Modules

Main logic of shipment is implemented in function module ZGTT_SOF_OTE_SHP_HD



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOF_OTE_SHP_HD". The function module "ZGTT_SOF_OTE_SHP_HD" is active. The "Source Code" tab is selected. The code is as follows:

```
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
    EXPORTING
        i_lifnr = <ls_xvttk>-tdlnr
    IMPORTING
        e_bp     = lv_forward_agt.
133 CLEAR: ls_bpdetail, lt_bpdetail.
134 CALL FUNCTION 'BAPI_IDENTIFICATIONDETAILS_GET'
    EXPORTING
        businesspartner      = lv_forward_agt
    TABLES
        identificationdetail = lt_bpdetail.
135 READ TABLE lt_bpdetail INTO ls_bpdetail WITH KEY identificationtype = 'LBN001' BINARY SEARCH.
136 ls_control_data-paramname = gc_cp_yn_shp_sa_lbn_id.
137 "   ls_control_data-value     = ls_bpdetail-identificationnumber.
138 * According the GTT adjustment, change the value of 'Service Agent LBN ID'
139 IF ls_bpdetail-identificationnumber IS NOT INITIAL.
140     CONCATENATE 'LBN#' ls_bpdetail-identificationnumber INTO ls_control_data-value.
141 ELSE.
142     CLEAR ls_control_data-value.
143 ENDIF.
144 APPEND ls_control_data TO e_control_data.
```

9: Coding Tips in the Planned Event function modules

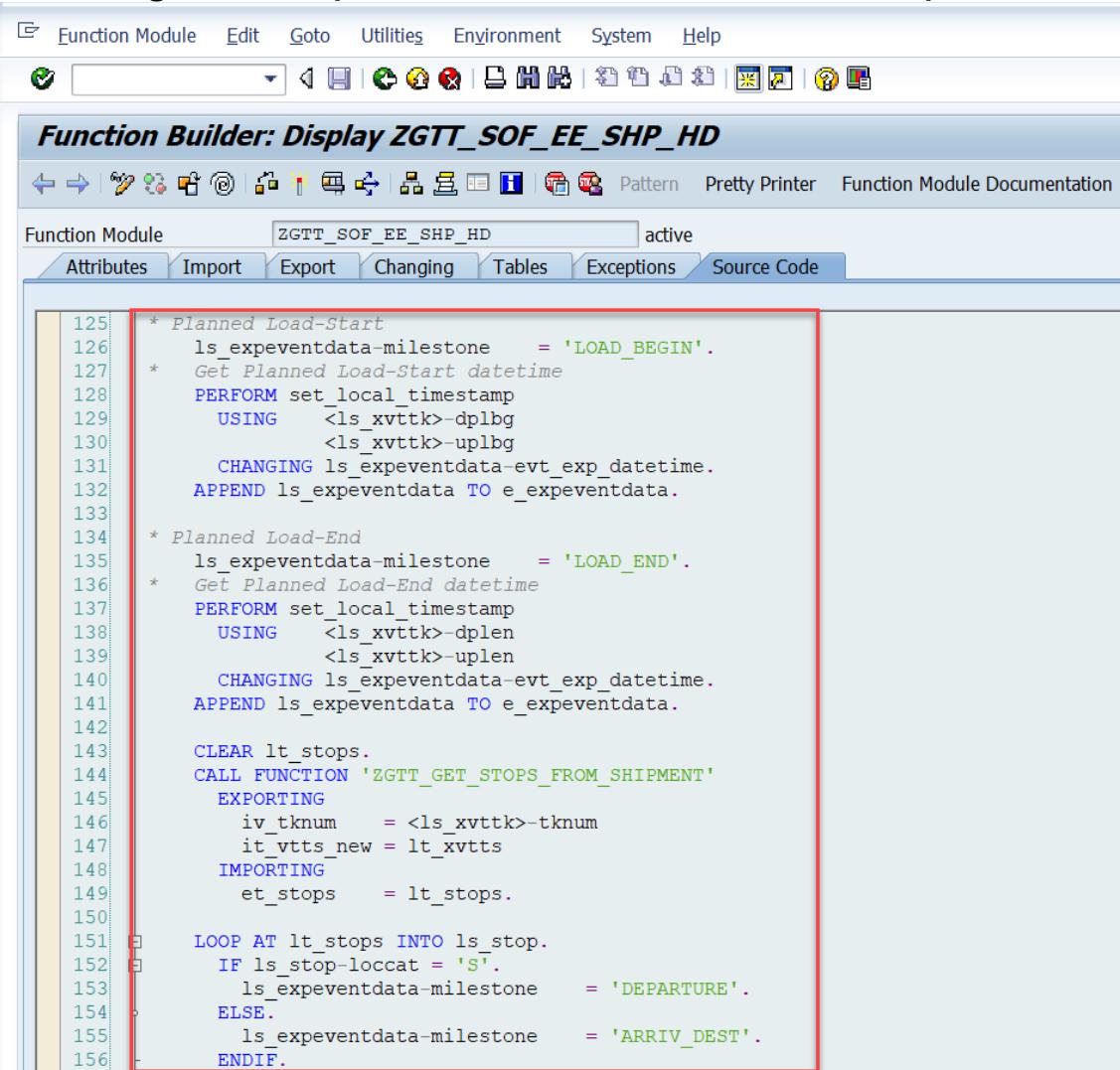
To customize the Planned Event function modules, key points are as follows:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. As default except no change made on the model configuration, SAP Business Network Global Track and Trace version 2 asks for full transport for all the planned events, which means that all the events needs to be extracted in all cases, no matter whether their values have been changed. If nothing is transported, the planned events will be removed in SAP Business Network Global Track and Trace Version 2.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT_EXP_DATETIME* is optional, but need to be filled with relevant time zone *EVT_EXP_TZONE* together if it needs to be transported.
6. The field *LOC_ID1* is optional, but need to be filled with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* app in SAP Business Network Global Track and Trace Version 2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

| Name | IDOC | Event Code |
|----------------------------------|------------|------------|
| 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 written in ABAP and handles the logic for planned events at stops. A red box highlights the section from line 125 to 156, which sets milestones for departure and arrival/destination based on stop location.

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

10: Coding Tips in the Event Data function modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding Event Type.
2. Add customization logics to fill the output table `CT_TRACKINGHEADER`, `CT_TRACKLOCATION`, `C_EVENTID_MAP`.
3. If the event has user-defined fields in Manage Models application, fill the table `CT_TRACKPARAMETERS`.
4. Add two technical parameters with fixed names '`ACTUAL_TECHNICAL_TIMEZONE`' and '`ACTUAL_TECHNICAL_DATETIME`' which are recommended for fixing IDOC sequencing issue (after object creation in S/4 actual event might be processed before object creation in SAP Business Network Global Track and Trace via TP request, which leads to an error)
5. If the event has reference table information, fill in the table `CT_TRACKREFERENCES`.
6. The field `CT_TRACKINGHEADER-SRCCOD`, `SRCID`, `SRCTX` is used for event reason transport.
7. In *Manage Models* app, click tab *IDOC Integration* to map the user-defined parameter names and model field names.

See sample code of function: `ZGTT_SOF_EE_DE_PICKING`

10: Coding Tips in the Event Data Function Modules

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

The screenshot shows the SAP Model Details interface for the 'sof' model, which is active. The 'IDOC Integration' tab is selected. The interface includes sections for Tracked Process Mapping, Tracked Process / Events, and User Model Fields.

Tracked Process Mapping:

- Tracked Process: DeliveryItem
- Integration Switch: ON

Tracked Process / Events (4):

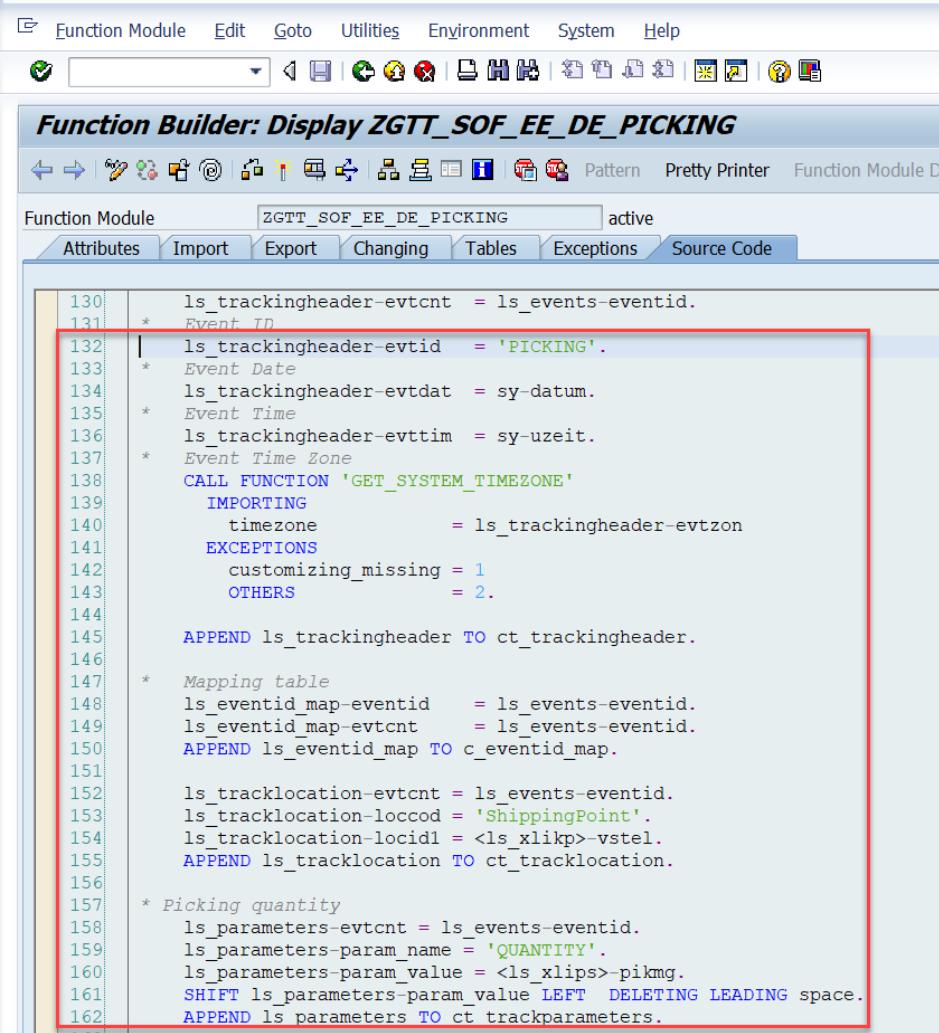
| Name | IDOC | Event Code |
|------------------------|----------|------------|
| Tracked Process | | |
| DeliveryItemEvent | E1EHPAO | |
| Event Types | | |
| Picking | E1EVMPAR | PICKING |
| Packing | E1EVMPAR | PACKING |
| DeliveryItemPOD | E1EVMPAR | DLV POD |

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 name "ZGTT_SOF_EE_DE_PICKING" is active. The code editor displays the following ABAP code:

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

11: Enhancement codes for cross-processes tracking

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

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

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

- Replace your Delivery AOT type name in Method *BEFORE_UPDATE* of BADI implementation *Z_GTT_SOF_LE_SHIPMNT*

The screenshot shows the SAP Business Add-In Builder interface for the 'Display Implementation Z_GTT_SOF_LE_SHIPMNT' screen. 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: `Z_GTT_SOF_LE_SHIPMNT` (Active)
- Implementation Short Text: `GTT - Enhancement to update the impacted delivery orders`
- Definition Name: `BADI_LE_SHIPMENT`
- Runtime Behavior: `Implementation will be called`

The 'Interface' tab is selected, showing:

- Interface Name: `IF_EX_BADI_LE_SHIPMENT`
- Name of Implementing Class: `ZCL_IM_GTT_SOF_LE_SHIPMNT`

A table lists the methods and their implementations:

| Method | Implementation Type | Description |
|----------------------------|---------------------|--|
| <code>AT_SAVE</code> | ABAP ABAP code | Process Shipments During "At Save" Context |
| <code>BEFORE_UPDATE</code> | ABAP ABAP code | Process Shipments During "Before Update" Context |
| <code>IN_UPDATE</code> | ABAP ABAP code | Process Shipments During "In Update" Context |

The rows for `BEFORE_UPDATE` and `IN_UPDATE` are highlighted with a red border.

At the bottom, there is a field for 'Default Implementation Class:' with a placeholder value.

11: Enhancement codes for cross-processes tracking

The cross processes tracking scenarios cover below:

Shipment -> Delivery and Delivery Item:

1\ Tracking ID (Delta Transport)

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

2\ Shipment Composition (Full Transport)

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

3\ Planned Event in Delivery (Full Transport)

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

4\ Planned Event in Delivery Item (Full Transport)

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

12: Known Issues

1. Planned Event Extension not enabled

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

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

2. IDOC sequencing issue

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

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

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

13: Solution of IDOC Sequencing Issue

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

2. Create CI tenant.

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

3. Create RFC destination

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

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

The image contains two 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.