



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

Logistics Business Network
March 2021

PUBLIC

Objectives



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

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

Agenda

A Prerequisites

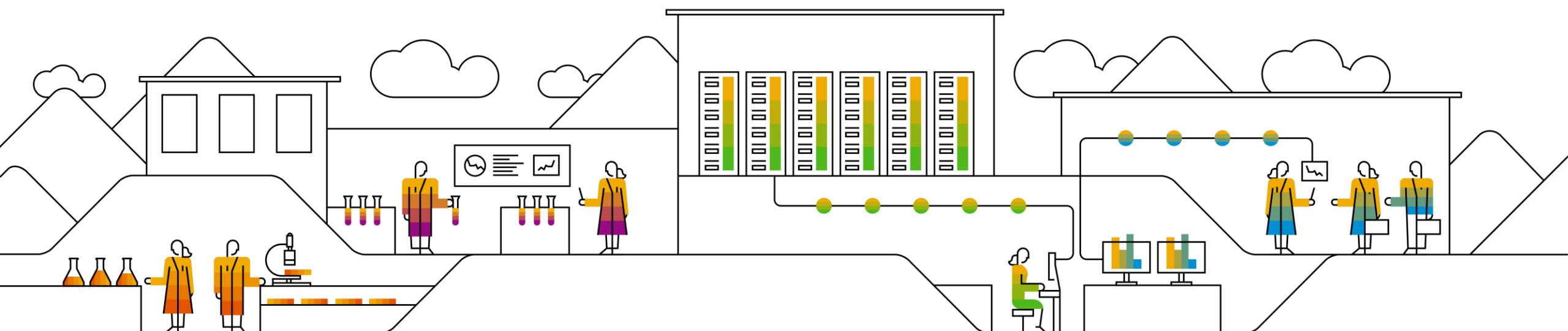
B Configuration and Implementation - Basic

 B1 IDOC Configuration

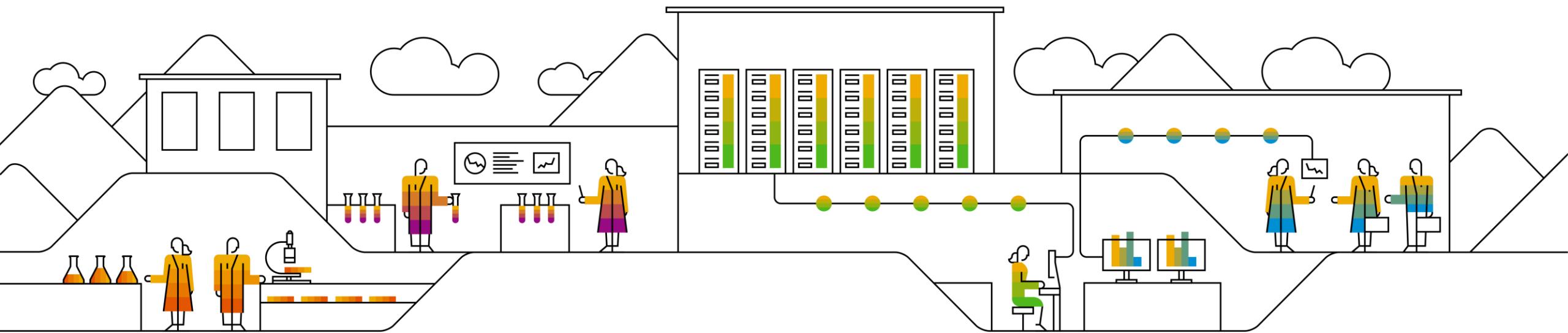
 B2 Extractor Configuration

C Download ABAP Code from GitHub

D Configuration and Coding Guide - Advanced



A) Prerequisites



STEP 1: Check the SAP Version

1-1: The SAP Product Version for GTT Version 2 shall be SAP EHP1 FOR SAP NETWEAVER 7.3 or higher

1-2: SAP NOTE 2937175 shall be implemented

1-3: The ABAP codes to support sample apps for GTT Version 2 shall be implemented in S4 HANA 1909 SP03 on premise, which is not validated in lower release, and not applicable for ECC series of products

TIPs:

1, SAP version reference: <https://support.sap.com/en/my-support/software-downloads/support-package-stacks/product-versions.html#section>

2, Note-assistant reference: <https://support.sap.com/en/my-support/knowledge-base/note-assistant.html>

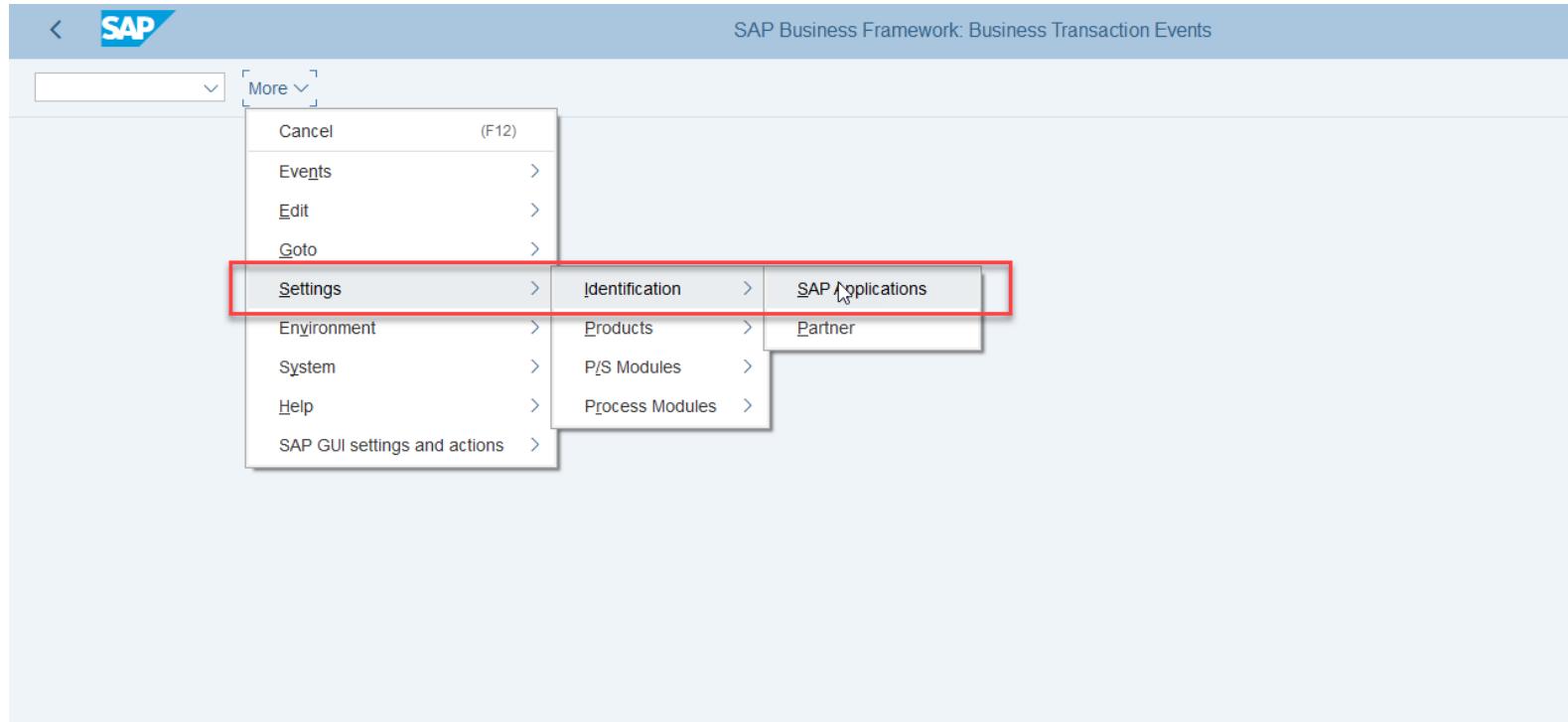
SAPNotes								
11 SAP Note(s) found								
SAP Component	Number	Versi...	Score	Title	Changed On	Status	Responsible	Category
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process	Thomas Rumbach	Program error
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer	Thomas Rumbach	Advance development
SCM-EM-AS	2834395	1	1	Solving ATC issues	27.09.2019	Released for Customer	D046164	Program error
SCM-EM-AS	2819787	1	1	TM-EM integration - analyzing errors	25.07.2019	In Process	Bernd Sieger	Help for error analysis
SCM-EM-AS-CNF	2798670	1	1	IMG activity inactive: Define SAP EM Extraction Functions	29.05.2019	Released for Customer	Bernd Sieger	Program error
SCM-EM-AS	2609449	4	1	Delete orphaned entries in table /SAPTRX/AOTREF (2)	11.07.2019	Pilot Release	Bernd Sieger	Workaround of missing
SCM-EM-AS	2502086	2	1	Aligning the BAPI processing mode with the communication mode	11.07.2017	Pilot Release	Bernd Sieger	Special development
SCM-EM-AS	2339984	2	1	Orphaned EM inbound queues in application systems	18.04.2019	Released for Customer	Bernd Sieger	Consulting
SCM-EM-AS	2159436	1	1	Runtime-Error "ABAP Programming" when trying to save delivery. System QSC-800	22.04.2015	In Process	D025889	Program error
SCM-EM-AS	1507998	4	1	Expert Consulting in the area of SAP Event Management	09.05.2011	Released for Customer	Florian Frey	Consulting
IS-R-PUR-PCC	896191	3	1	FAQ: EM seasonal procurement (Consulting, Tips, Customizing)	13.07.2006	Released for Customer	Andreas Lange	FAQ

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**

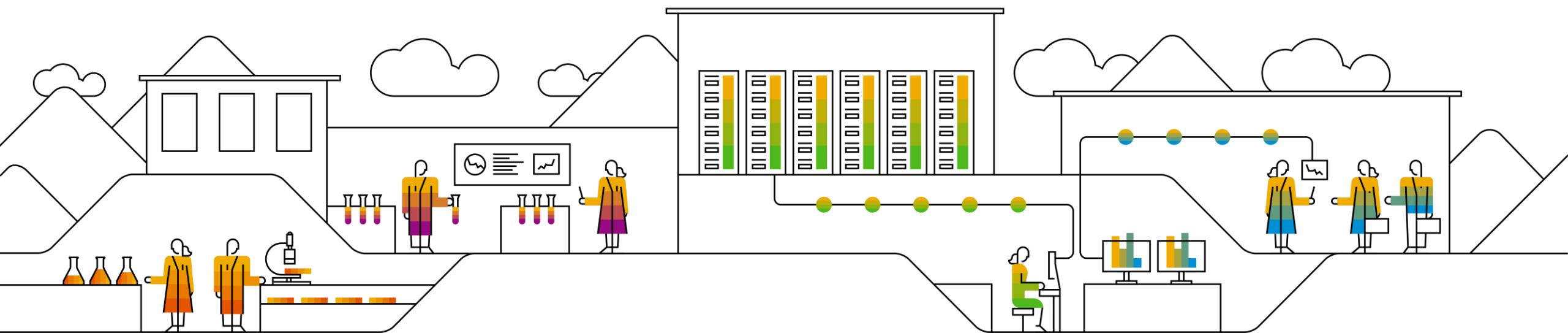
The screenshot shows a SAP application window titled "Change View 'BTE Application Indicator'. Overview". The window displays a list of applications and their status. The application "PI-EM" is selected, indicated by a red box around its row. The "Text" column for PI-EM contains the value "SAP Event Manager Integration". The "Active" checkbox for PI-EM is checked, also indicated by a red box. 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 application PM has its "Active" checkbox checked. The application PMIPUR has its "Active" checkbox unchecked. The application PRICAT has its "Active" checkbox unchecked. The application PSRV has its "Active" checkbox checked. The application QBEXT has its "Active" checkbox checked. The application QBEXTP has its "Active" checkbox checked. The application QILPO has its "Active" checkbox checked. The application RDSVFI has its "Active" checkbox checked. The application RDSVMD has its "Active" checkbox checked.

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 GTT

1-1: Log on to the business client

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

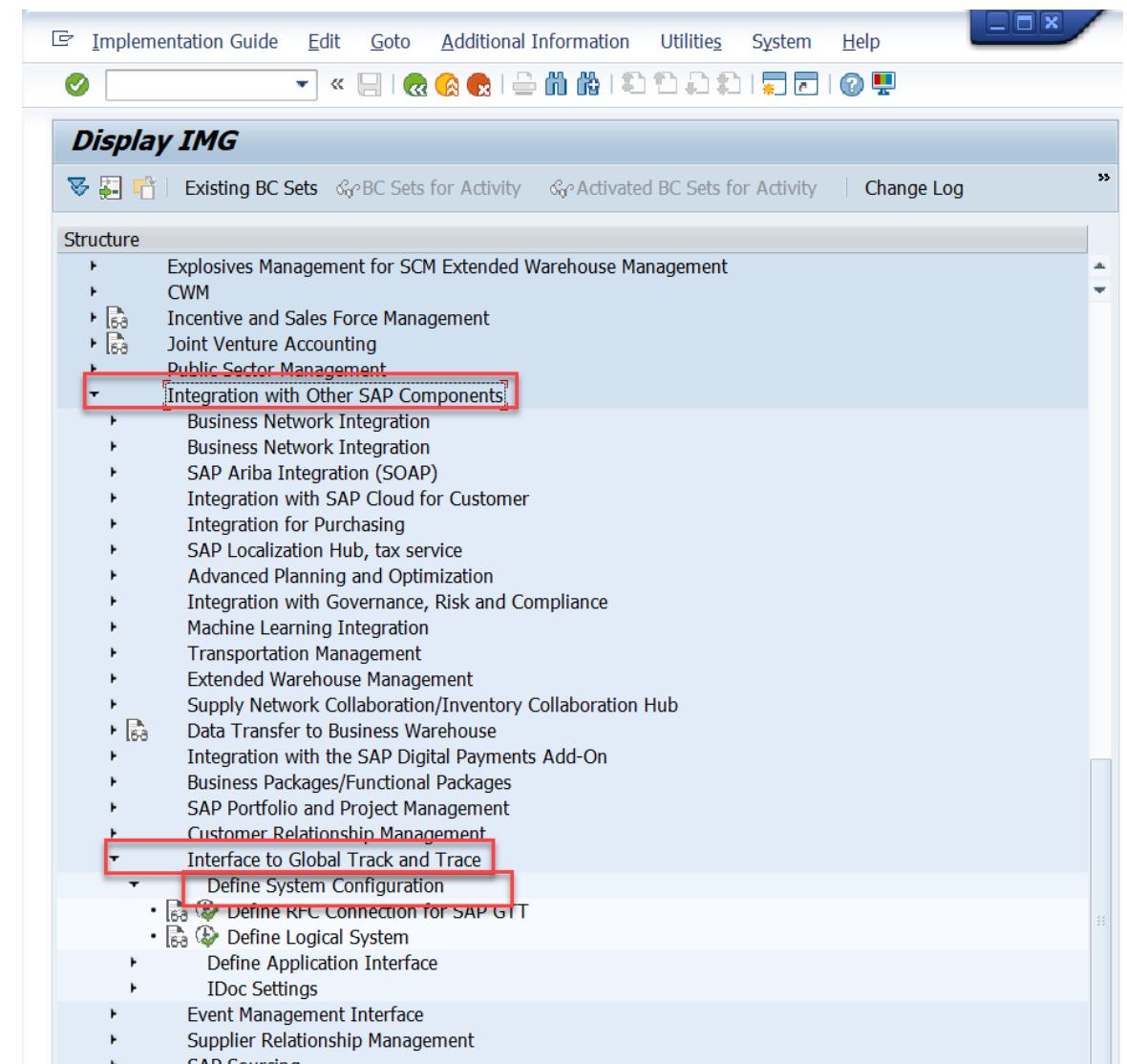
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

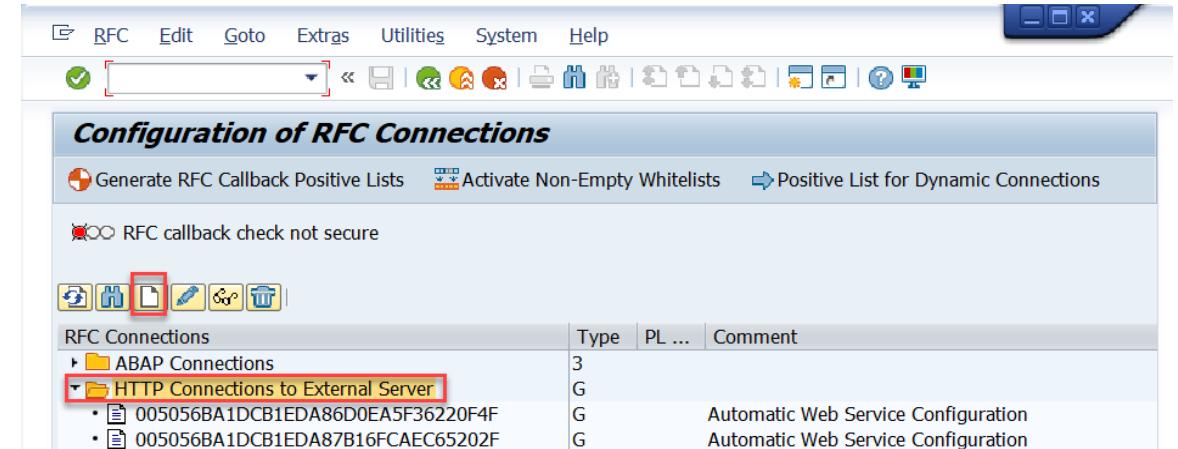
Define RFC Connection for SAP GTT



STEP 1: Define RFC Connection for GTT

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

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



STEP 1: Define RFC Connection for GTT

1-7: Enter a description

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

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

<https://sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com/>

Host: sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com

Port: 443

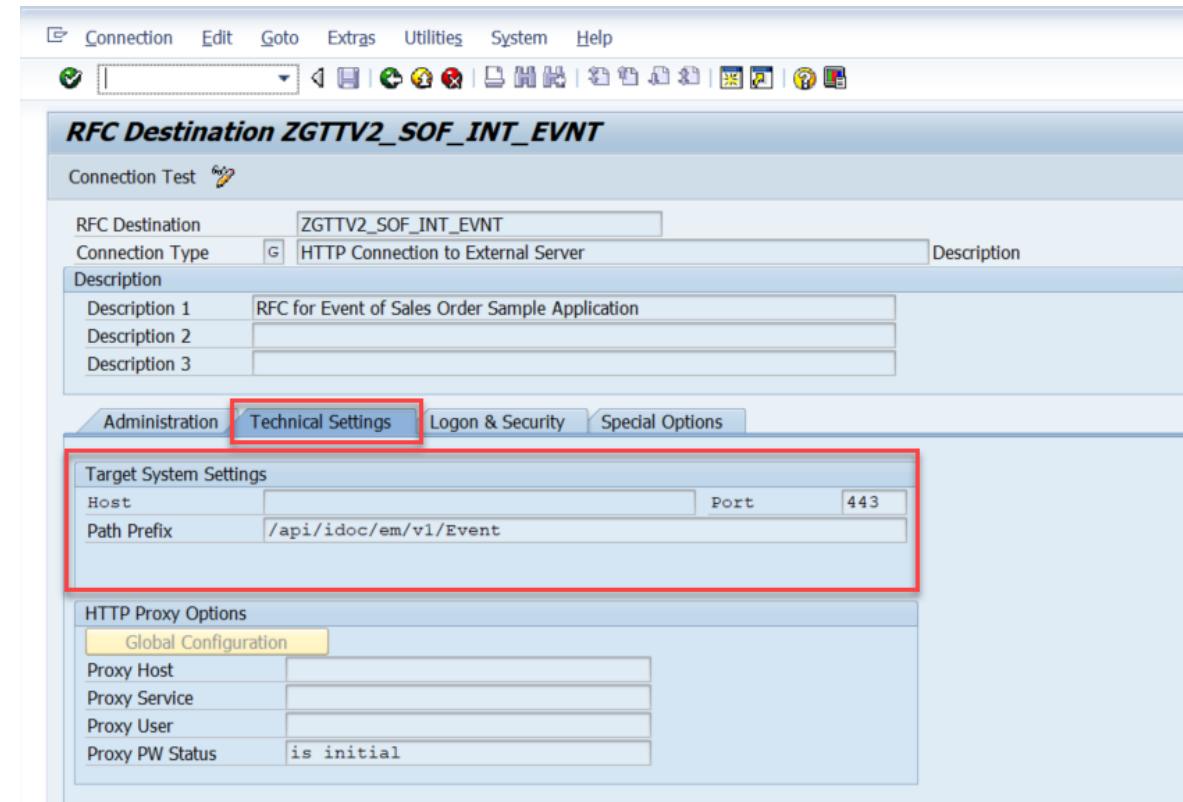
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

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	sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/Event	443
ZGTTV2_SOF_INT_TP	RFC for Tracked Process of Sales Order Sample Application	sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/TrackedProc	443

STEP 1: Define RFC Connection for GTT

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

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

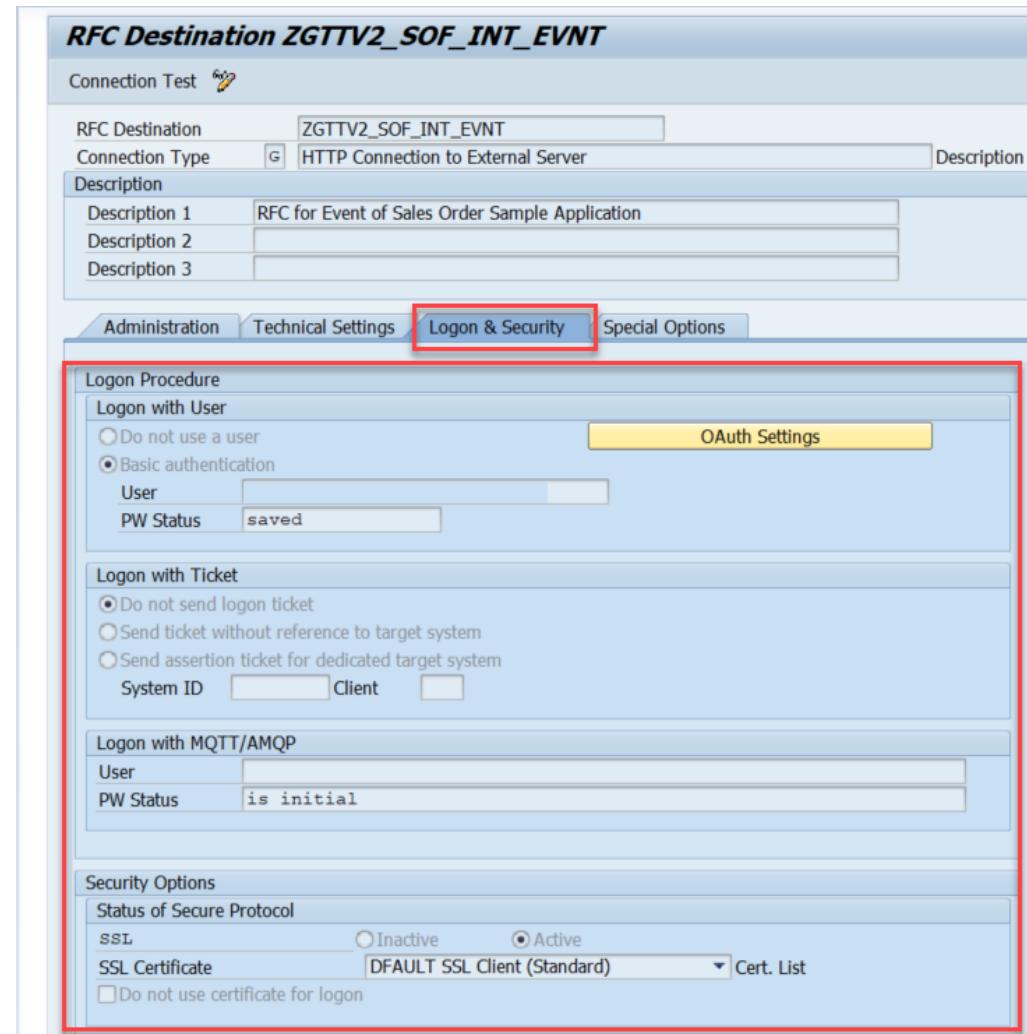
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

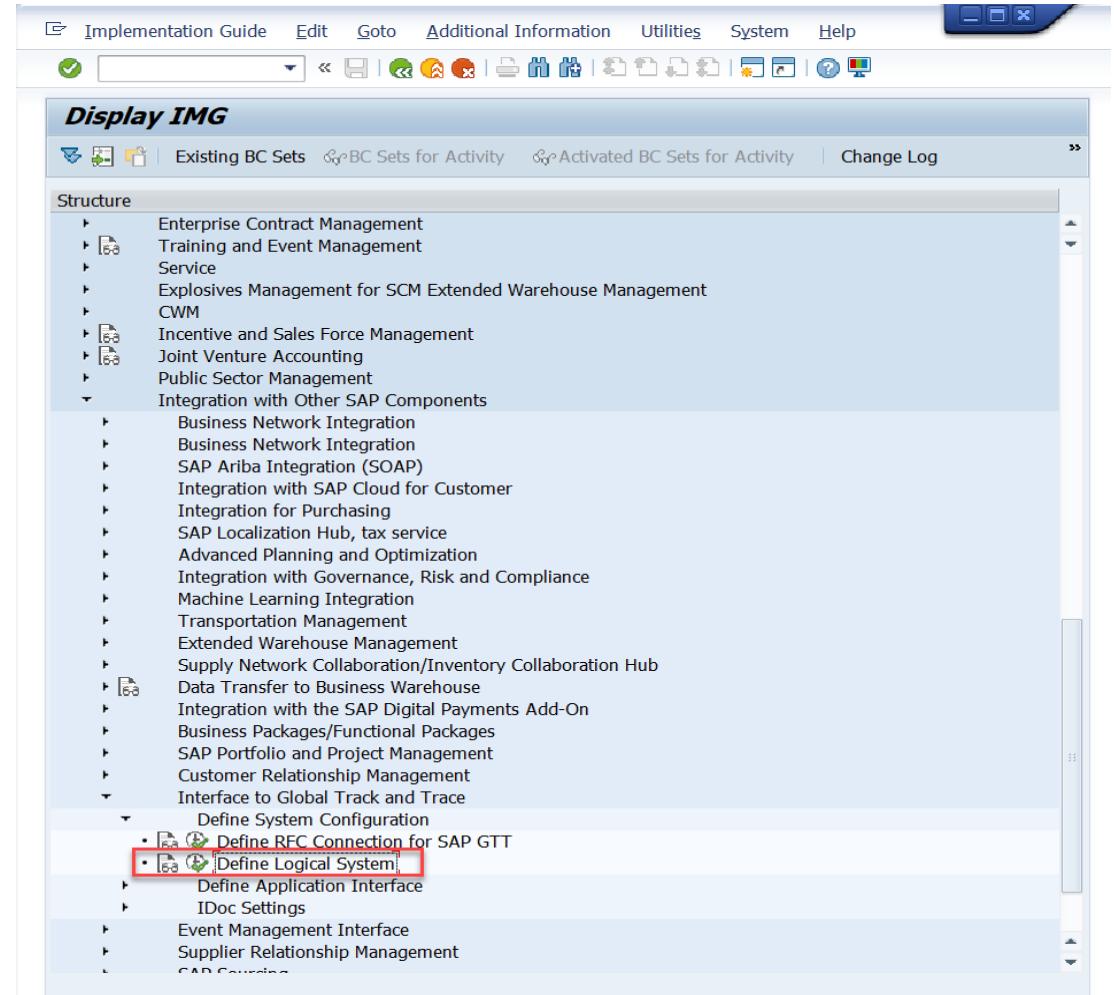
- one for event and
- the other for tracked process.



STEP 2: Define Logical System

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

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

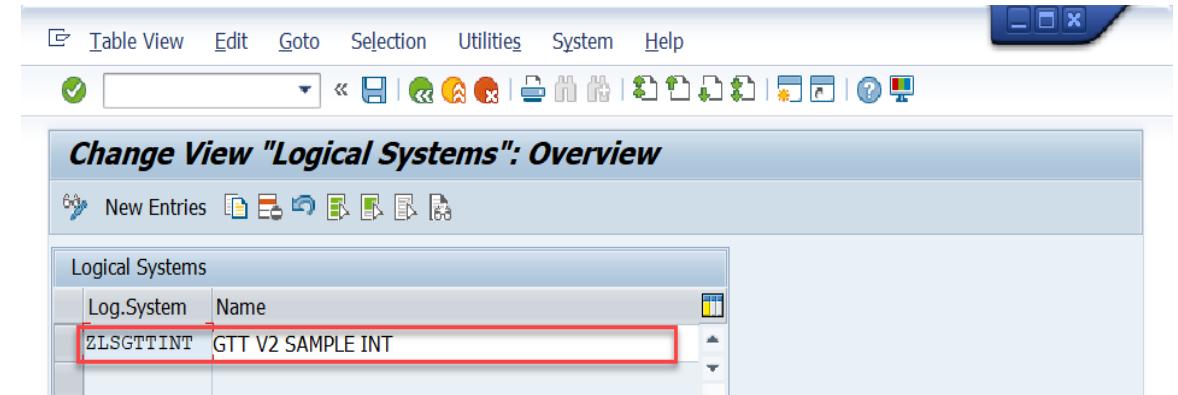


STEP 2: Define Logical System

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

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

2-4: Save the configuration

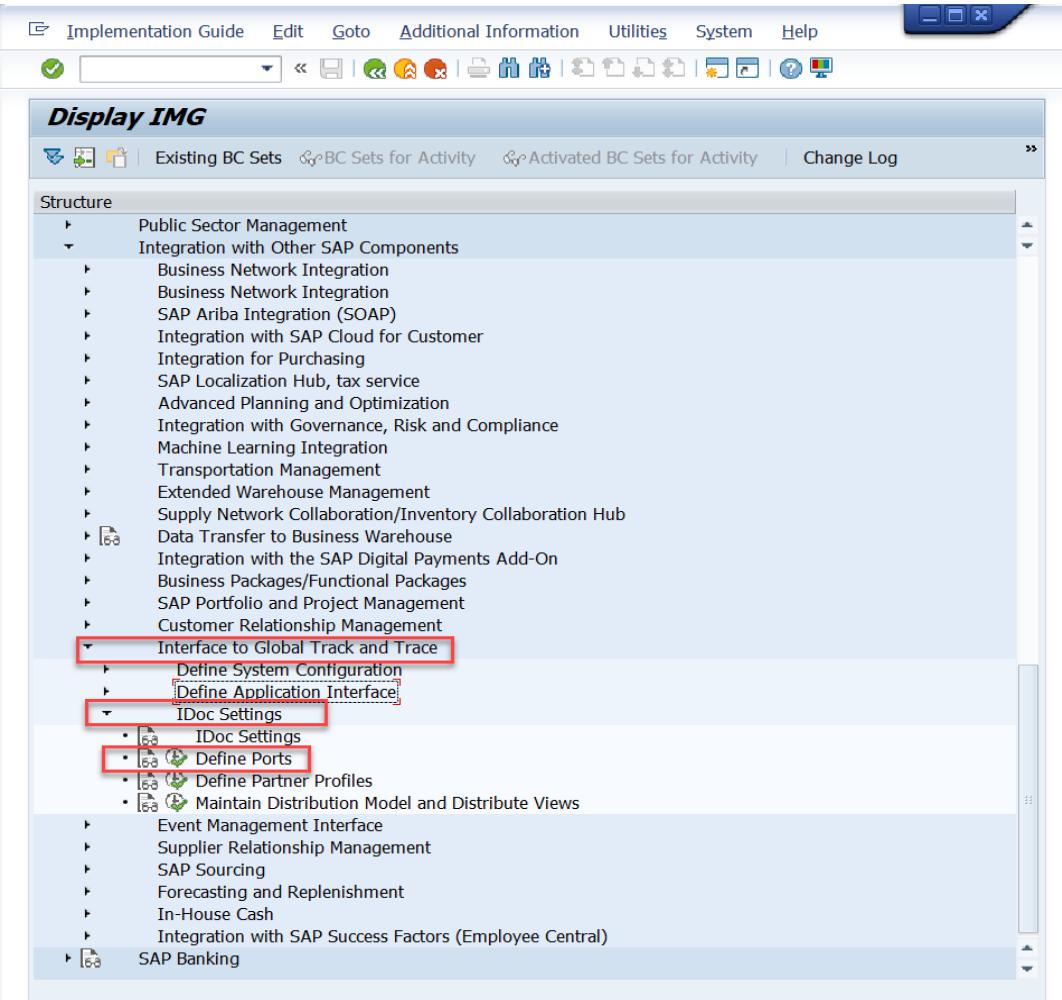


Log.System	Name
ZLSGTTINT	GTT V2 SAMPLE INT

STEP 3: Define Ports

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

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



STEP 3: Define Ports

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

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

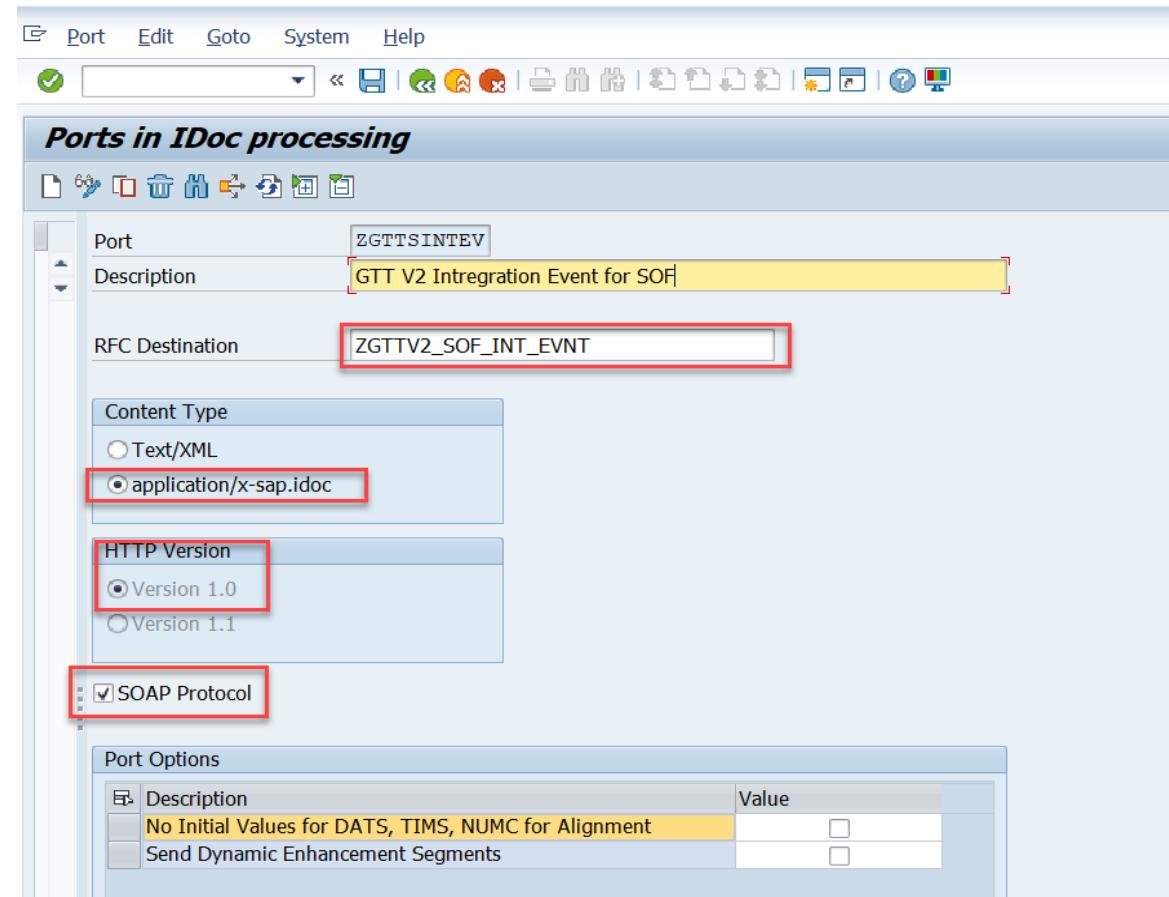
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

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

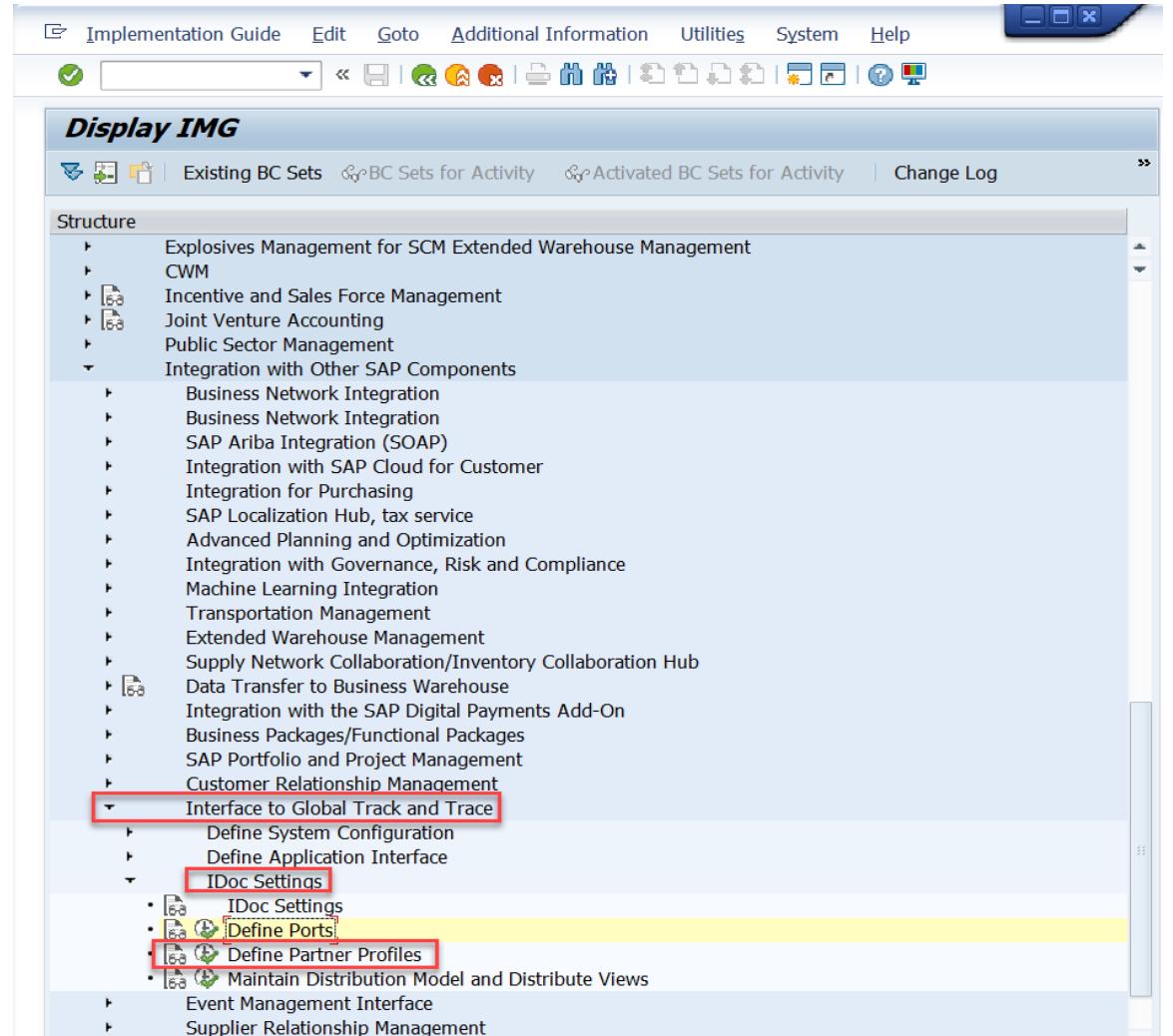


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSINTEV	GTT V2 Integration Event for SOF	ZGTTV2_SOF_INT_EVNT	application/x-sap.idoc	Version 1.0	Checked
ZGTTSINTTP	GTT V2 Integration Tracked Process for SOF	ZGTTV2_SOF_INT_TP	application/x-sap.idoc	Version 1.0	Checked

STEP 4: Define Partner Profiles

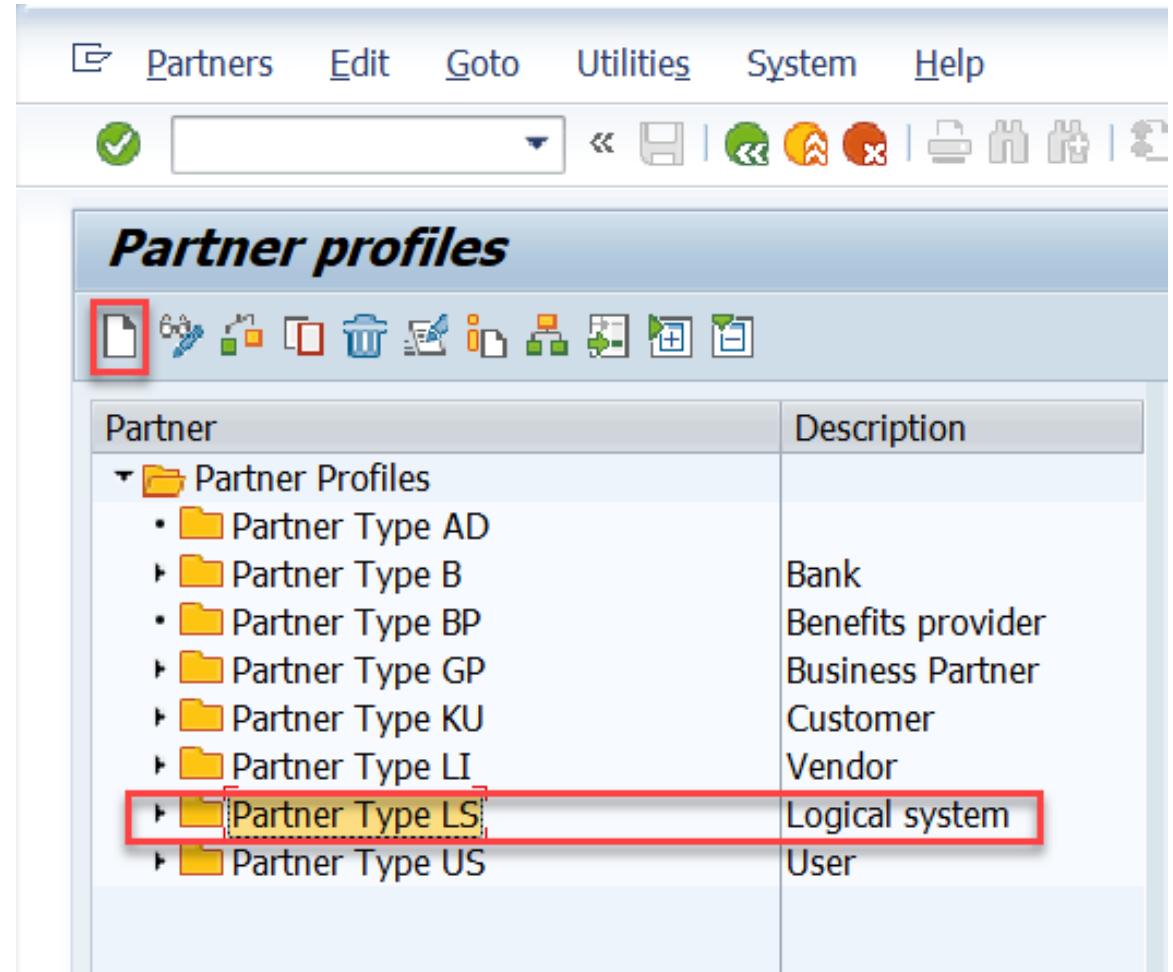
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components -> Interface to Global Track and Trace -> IDoc Settings**

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



STEP 4: Define Partner Profiles

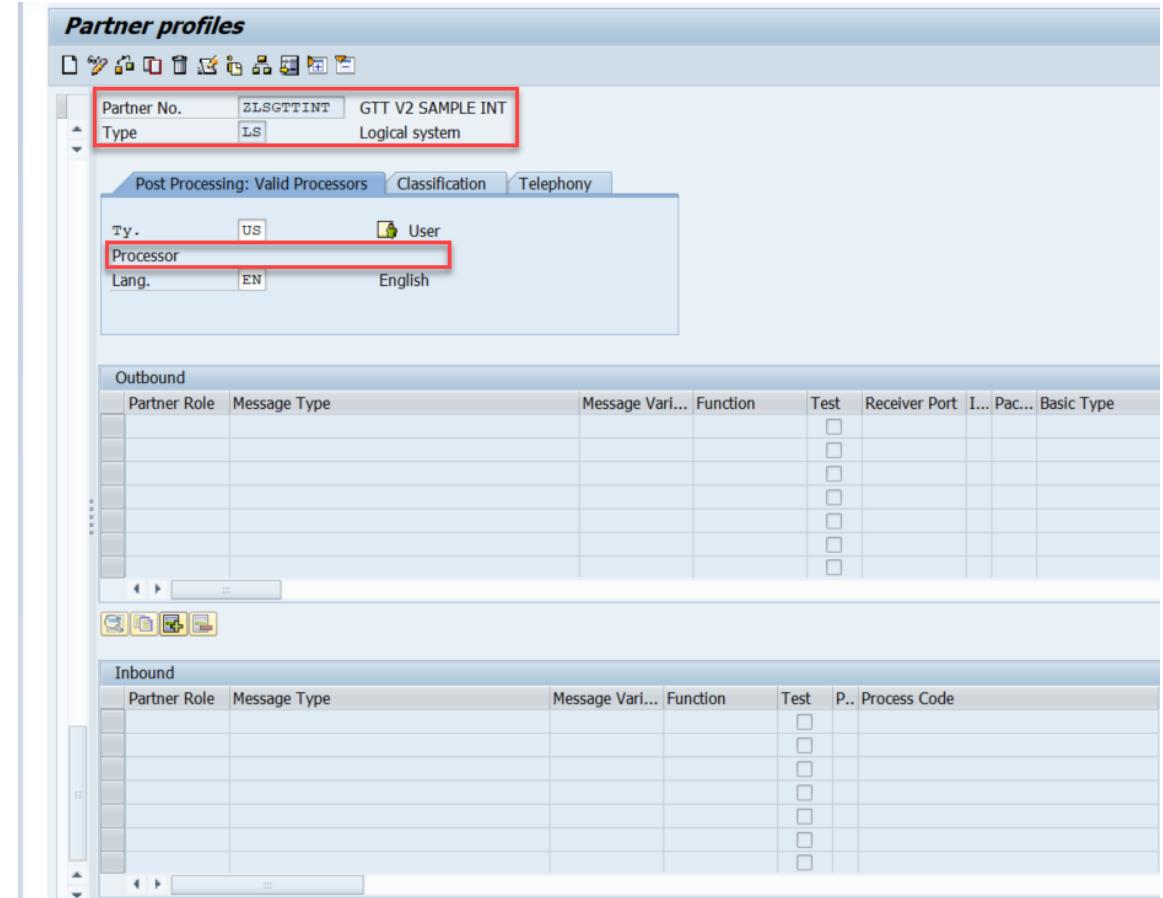
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

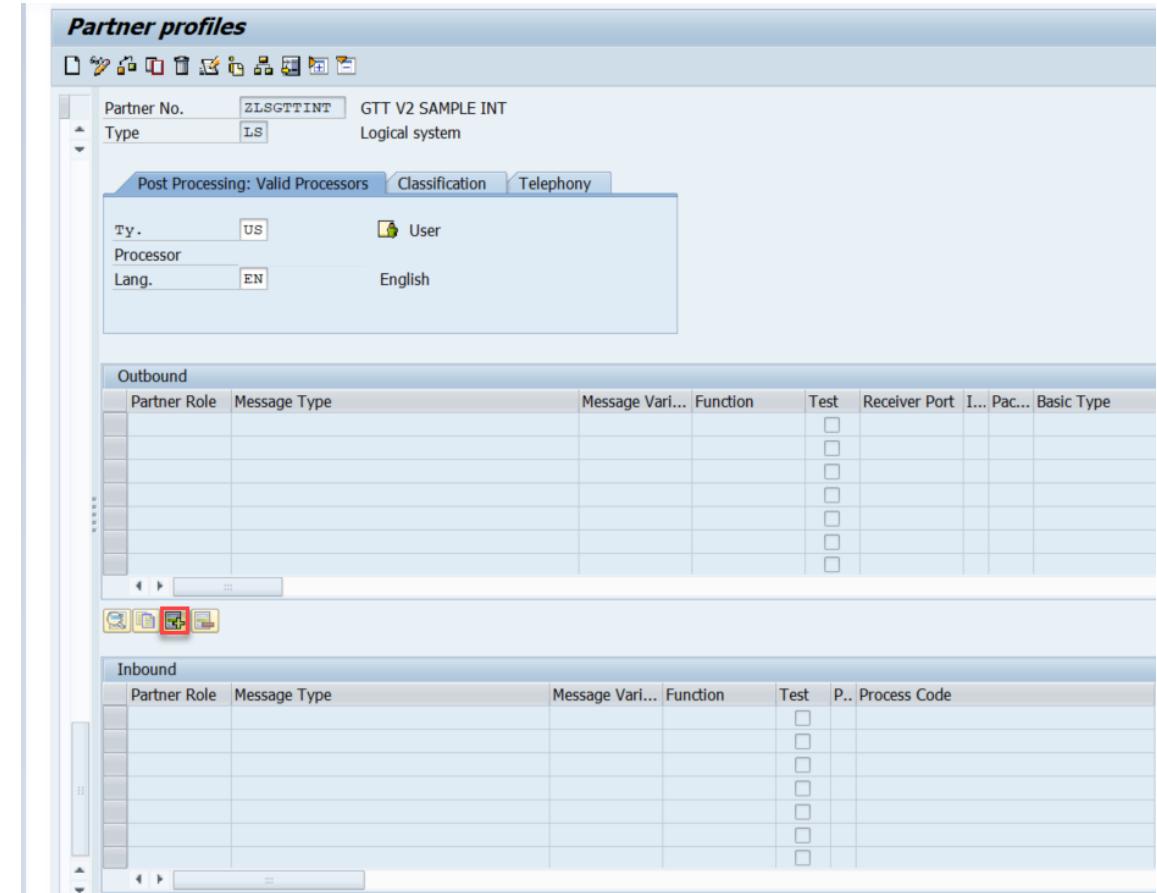
4-4: Fill in the **Partner No.** that you created in STEP 2

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



STEP 4: Define Partner Profiles

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



STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

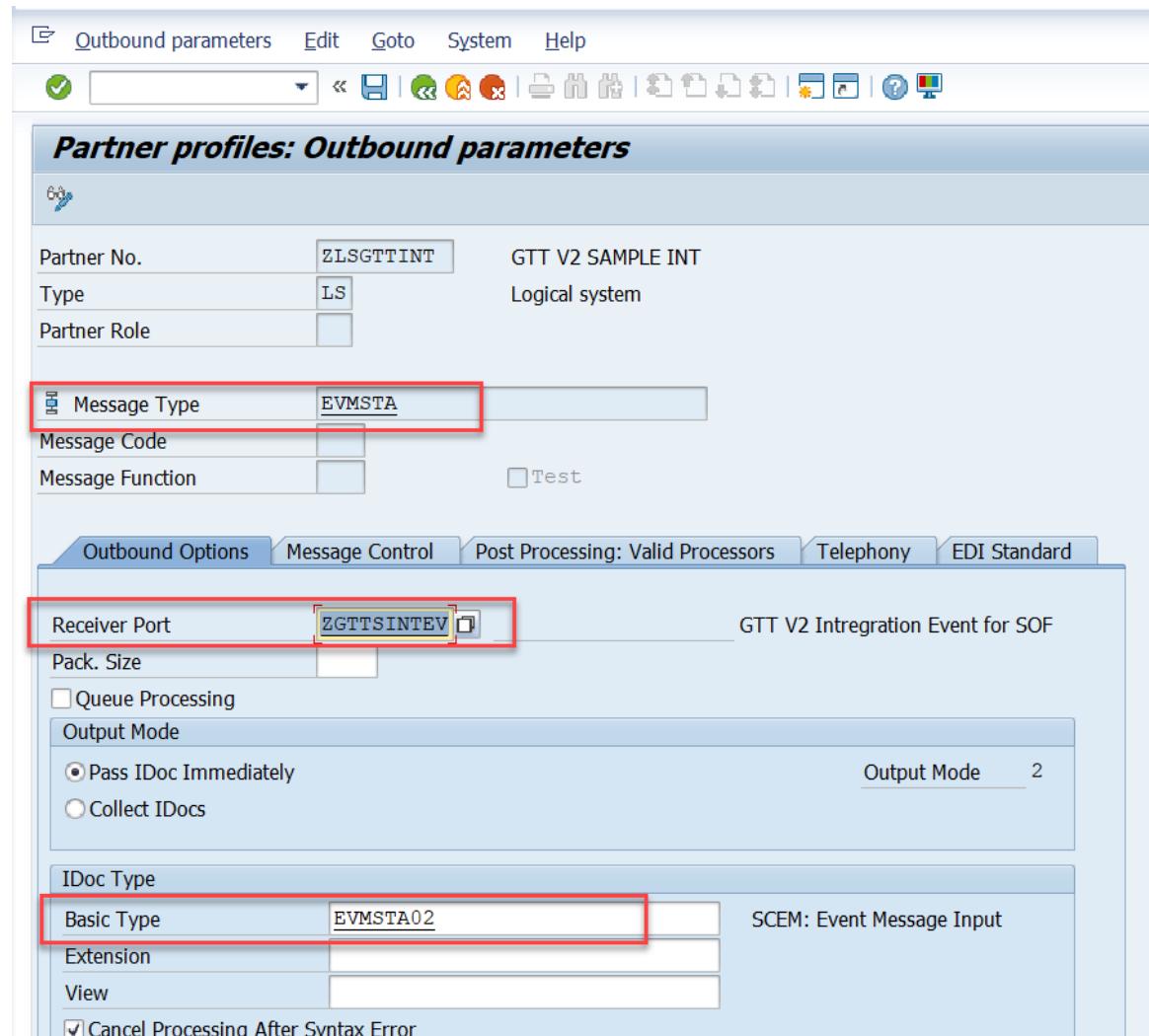
For the tracked Process:

Message Type: AOPOST

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

4-9: Save the configuration

Caution: In this step, you need to repeat steps 4-6 to 4-9 to add two outbound parameters, one for event and the other for tracked process.



STEP 4: Define Partner Profiles

4-10: Fill in the Message Type.

For the tracked Process:

Message Type: AOPOST

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

4-12: Save the configuration

Partner No.	Type	Outbound	Message Type	Receiver Port	IDoc Type
ZLSGTTINT	LS	Yes	AOPOST	ZGTTSINTTP	EHPOST01
ZLSGTTINT	LS	Yes	EVMSTA	ZGTTSINTEV	EVMSTA02

Partner profiles: Outbound parameters

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

Message Type AOPOST (highlighted with a red box)

Message Code
Message Function

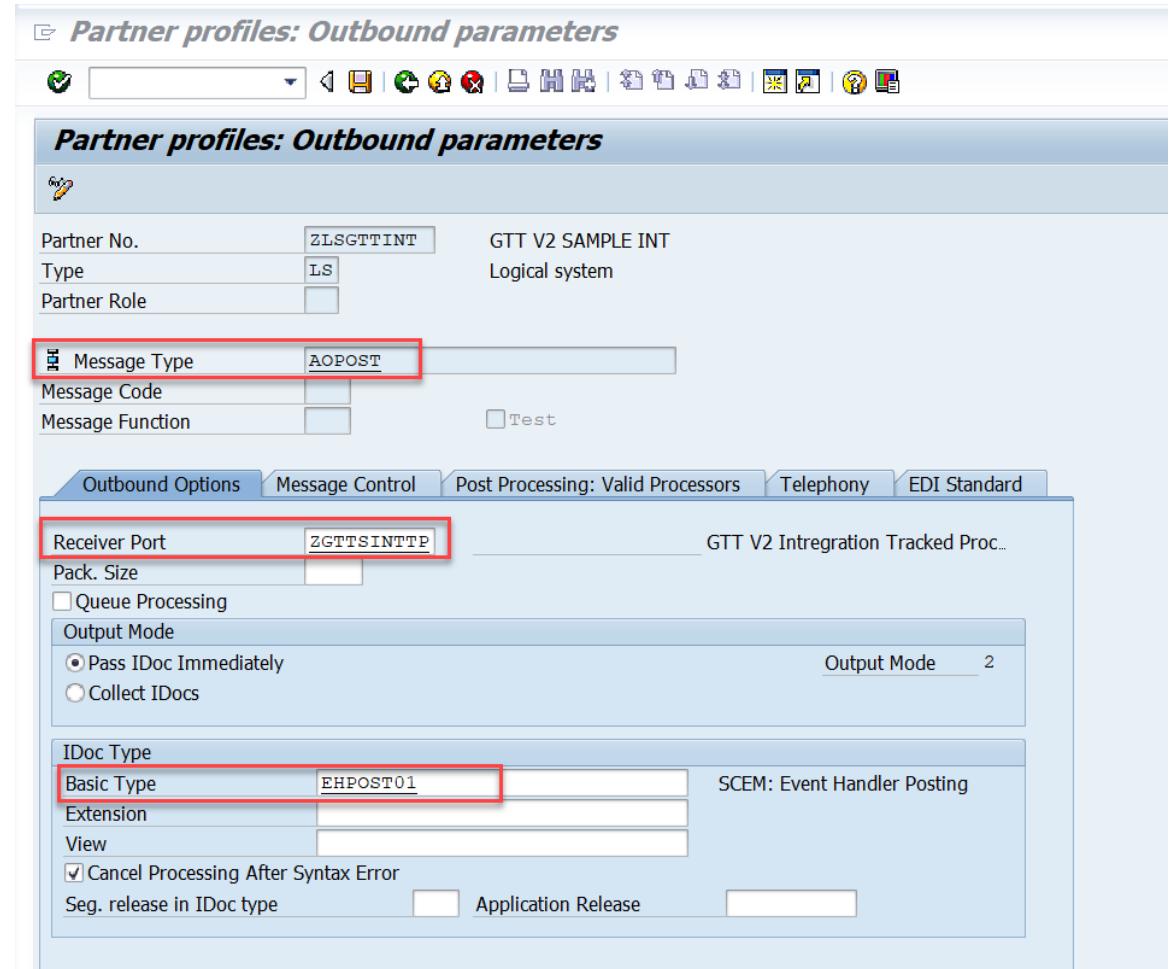
Test

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

Receiver Port ZGTTSINTTP (highlighted with a red box)
GTT V2 Intregation Tracked Proc...

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

IDoc Type Basic Type EHPOST01 (highlighted with a red box)
SCEM: Event Handler Posting
Extension
View
 Cancel Processing After Syntax Error
Seg. release in IDoc type Application Release



B) Configuration and Implementation

- Basic

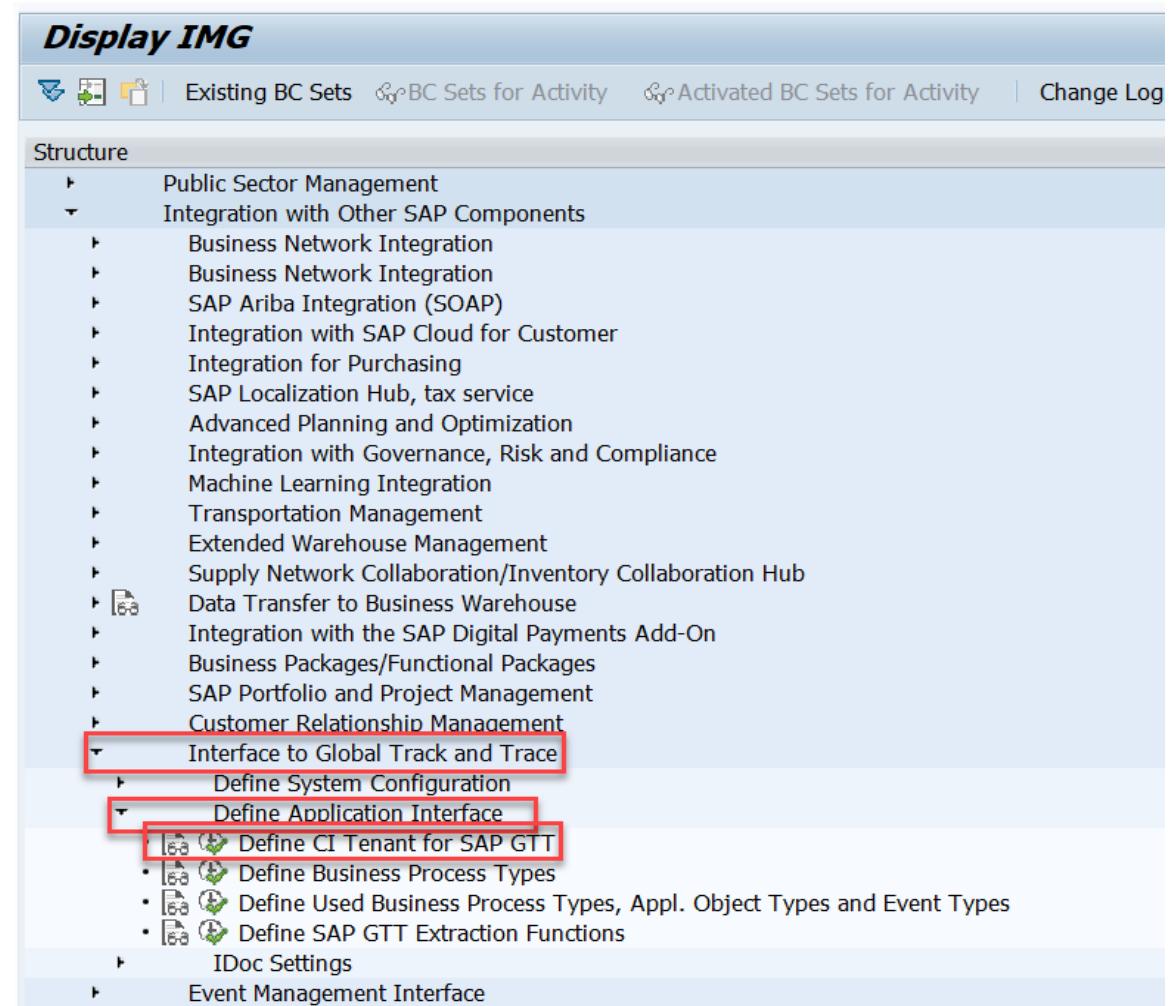
B2. Extractor Configuration



STEP 5: Define CI Tenant for GTT

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

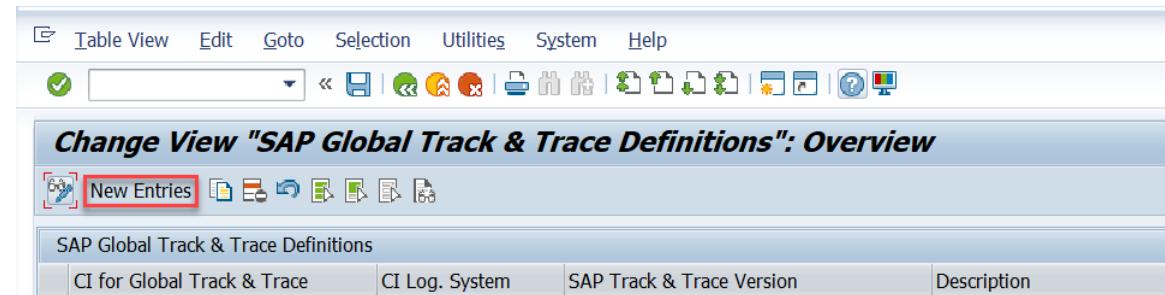
5-2: Choose activity
Define CI Tenant for SAP GTT



STEP 5: Define CI Tenant for GTT

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

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



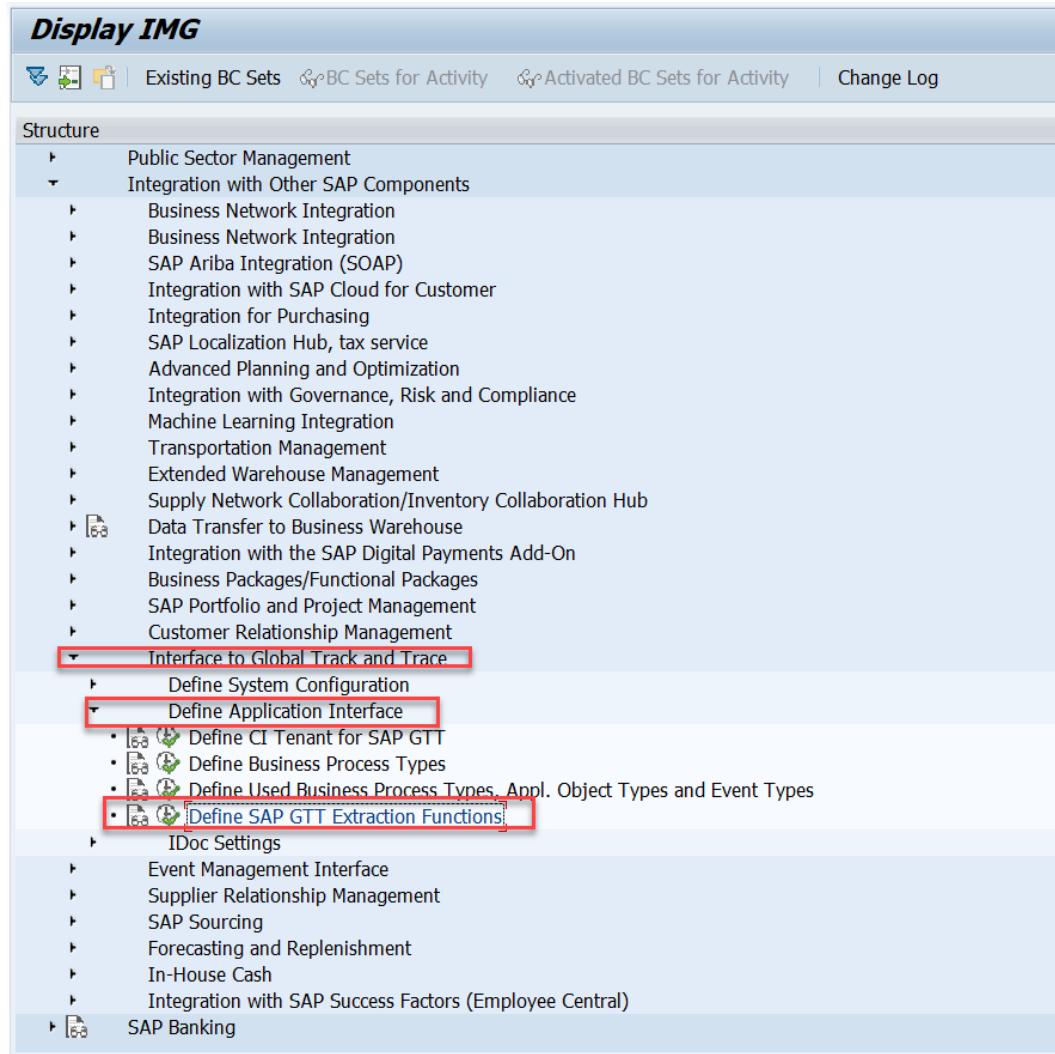
The screenshot shows the SAP GUI interface with the title bar "Display View 'SAP Global Track & Trace Definitions': Overview". Below the title bar is a toolbar with various icons. The main area displays a table titled "SAP Global Track & Trace Definitions" with four columns: CI for Global Track & Trace,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

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

STEP 6: Define GTT Extraction Functions

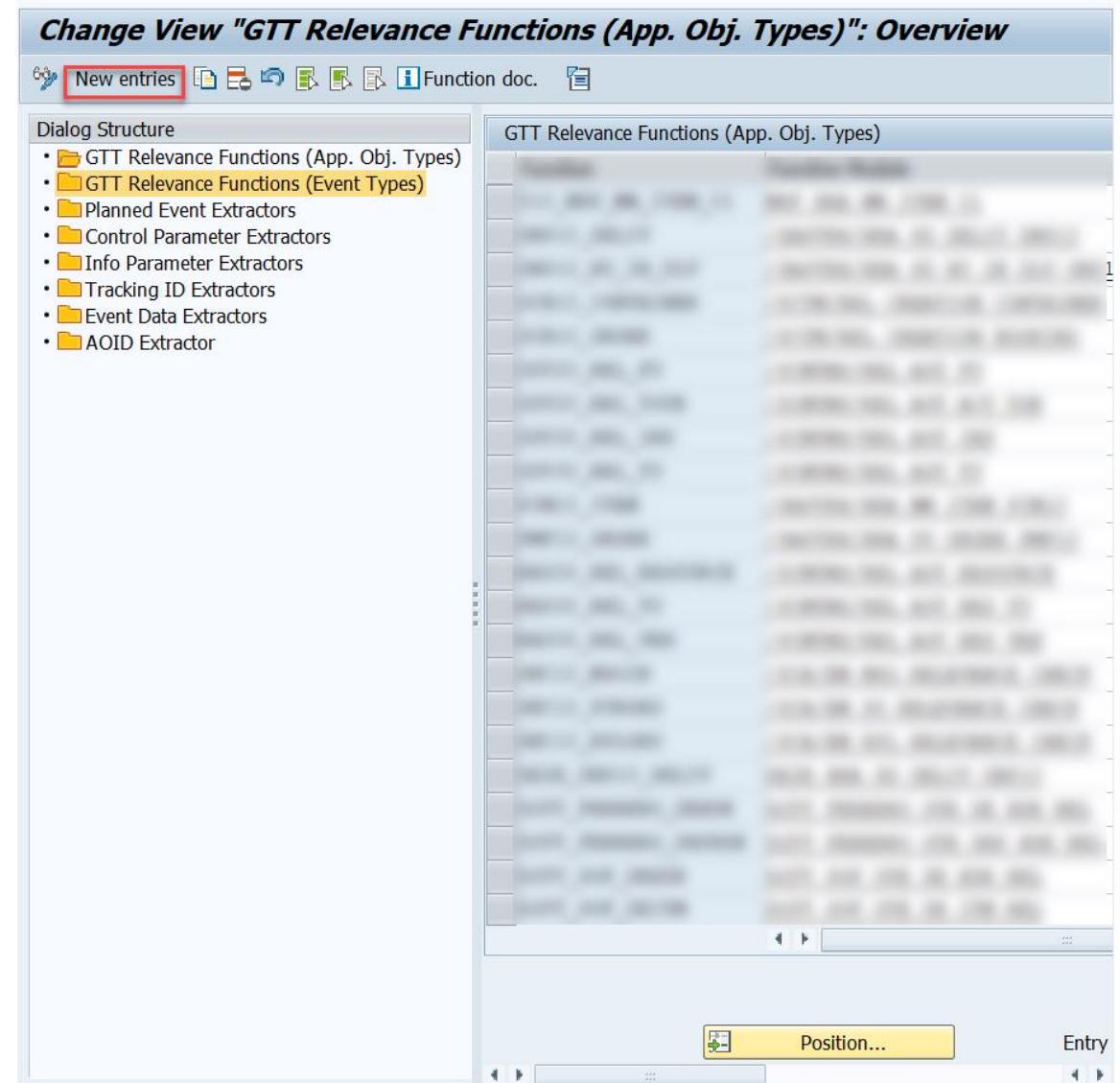
6-1: In **Display IMG** page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

6-2: Choose activity
Define SAP GTT Extraction Functions



STEP 6: Define GTT Extraction Functions

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



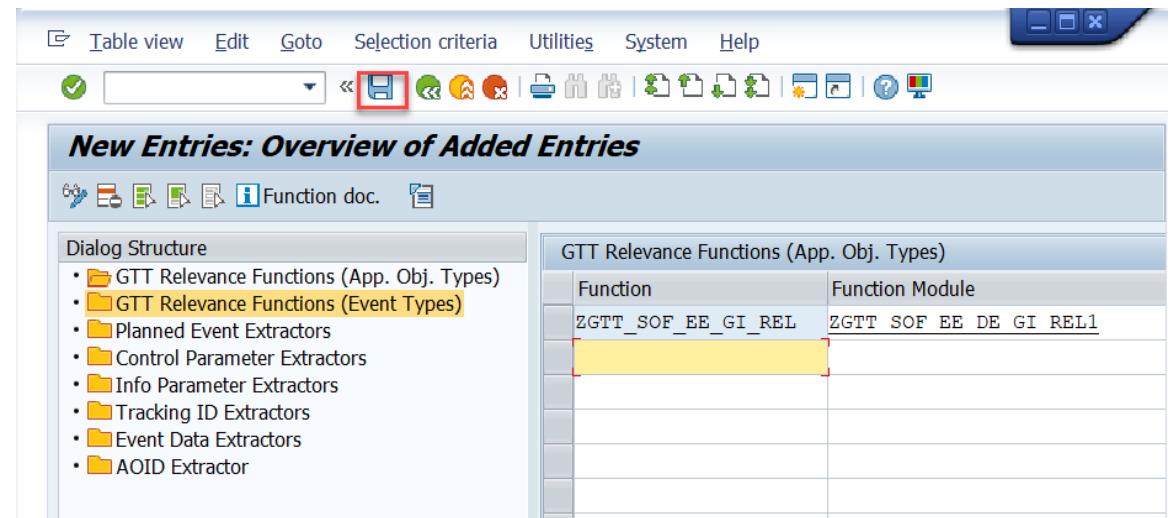
STEP 6: Define GTT Extraction Functions

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

New Entries: Overview of Added Entries	
GTT Relevance Functions (App. Obj. Types)	
Function	Function Module
ZGTT_SOF_EE_GI_REL	ZGTT_SOF_EE_DE_GI_REL1

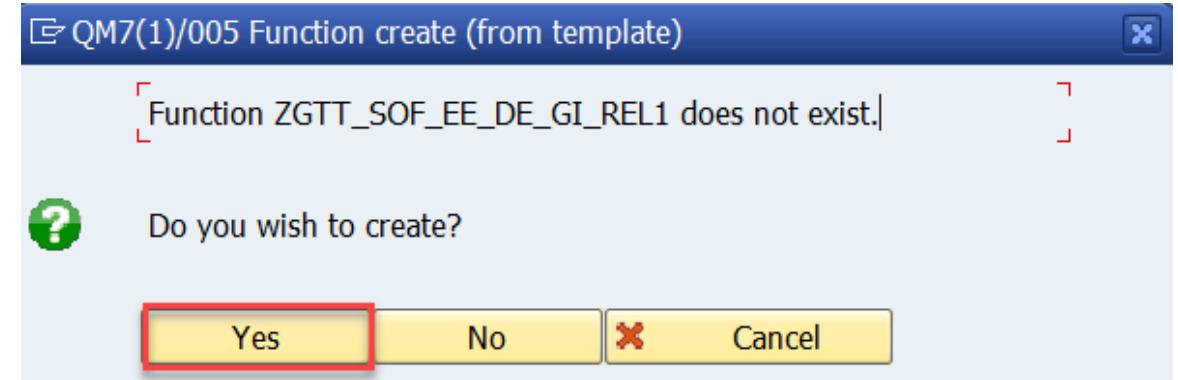
STEP 6: Define GTT Extraction Functions

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 following details:

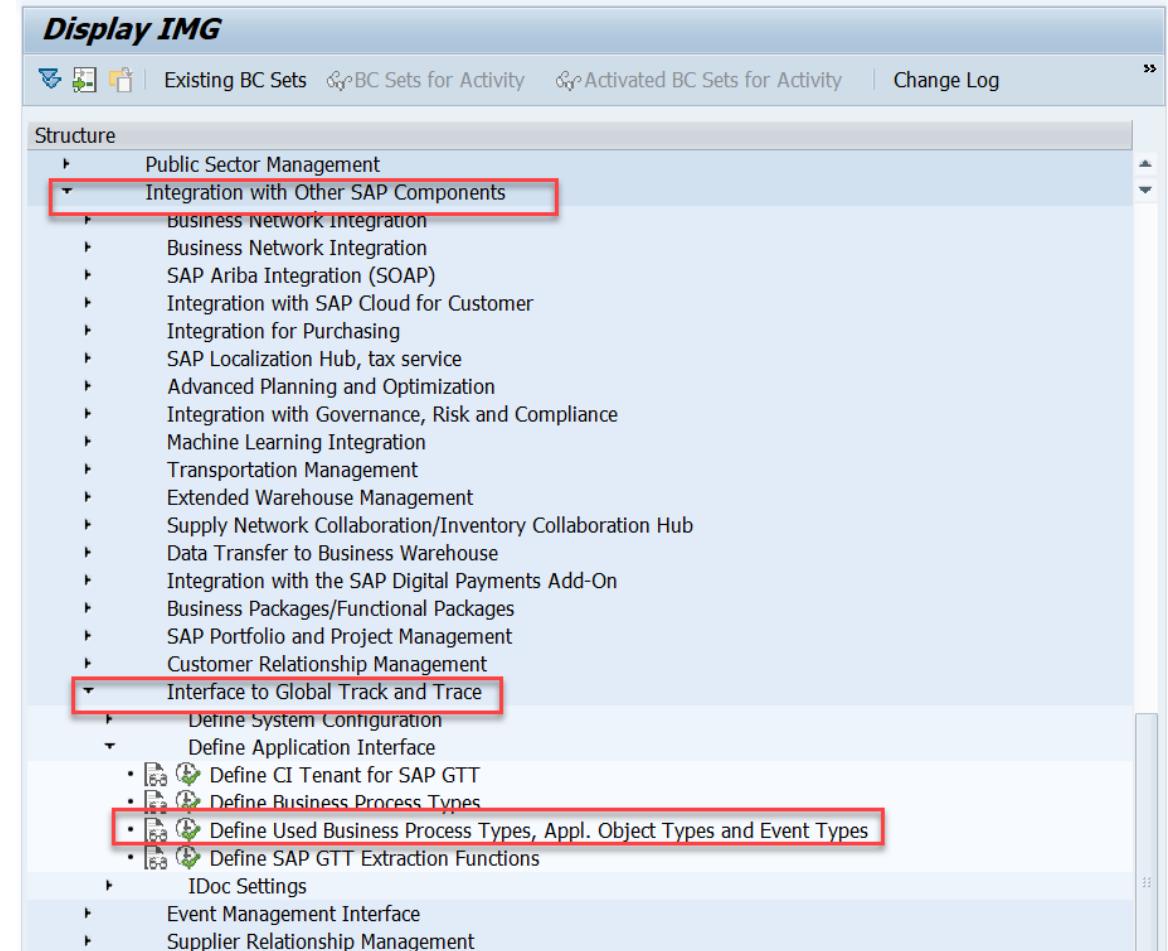
- Title Bar:** Function Builder: Display ZGTT_SOF_EE_DE_GI_REL1
- Function Module:** ZGTT_SOF_EE_DE_GI_REL1 (inactive)
- Attributes:** Import, Export, Changing, Tables, Exceptions, Source Code
- Repository Browser:** Function Group dropdown set to ZGTT_SOF, Object Name dropdown set to ZGTT_SOF.
- Function Modules:** A list of function modules under ZGTT_SOF, with ZGTT_SOF_EE_DE_GI_REL1 highlighted.
- Source Code:** ABAP code for the function module ZGTT_SOF_EE_DE_GI_REL1, showing imports, exports, parameters, tables, exceptions, and the main function block.
- Status:** Scope: FUNCTION ZGTT_SOF_EE_DE_GI_REL1, ABAP, Ln 13 Col 48.

```
1  FUNCTION ZGTT_SOF_EE_DE_GI_REL1.
2
3  *"** Local Interface:
4  *" IMPORTING
5  *"   REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM
6  *"   REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/AOTYPES
7  *"   REFERENCE(I_ALL_APFL_TABLES) TYPE TRXAS_TABCONTAINER
8  *"   REFERENCE(I_APPTYPE_TAB) TYPE TRXAS_APPTYPE_TABS_WA
9  *"   REFERENCE(I_APP_OBJECT) TYPE TRXAS_APPOBJ_CTAB_WA
10 *" EXPORTING
11 *"   VALUE(E_RESULT) LIKE SY-BINPT
12 *" TABLES
13 *"   C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL
14 *" EXCEPTIONS
15 *"   PARAMETER_ERROR
16 *"   RELEVANCE_DETERM_ERROR
17 *"   STOP_PROCESSING
18 *-
19 *-
20 * Top Include
21 * TYPE-POOLS:trxas.
22 *-
23
24
25
26 ENDFUNCTION.
```

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

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

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



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

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

In the following:

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

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

STEP 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			
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

GTT Relevant
 Stop ET Det.
 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
- Data Source for Events:**
 - Main Obj. Table: SHIPMENT_HEADER_NEW (highlighted with a red box)
 - Master Table: SHIPMENT_HEADER_OLD (highlighted with a red box)
- Reference Between Main and Master Table:**
 - First Field Reference from Main to Master Table
 - Second Field Reference from Main to Master Table

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)
- 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)

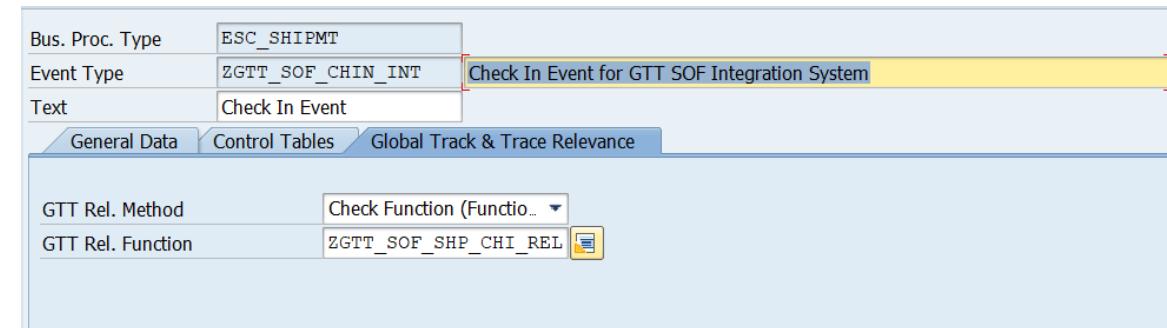
A red box highlights the "Event on Item level" note next to the Main Obj. Table entry.

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**.



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

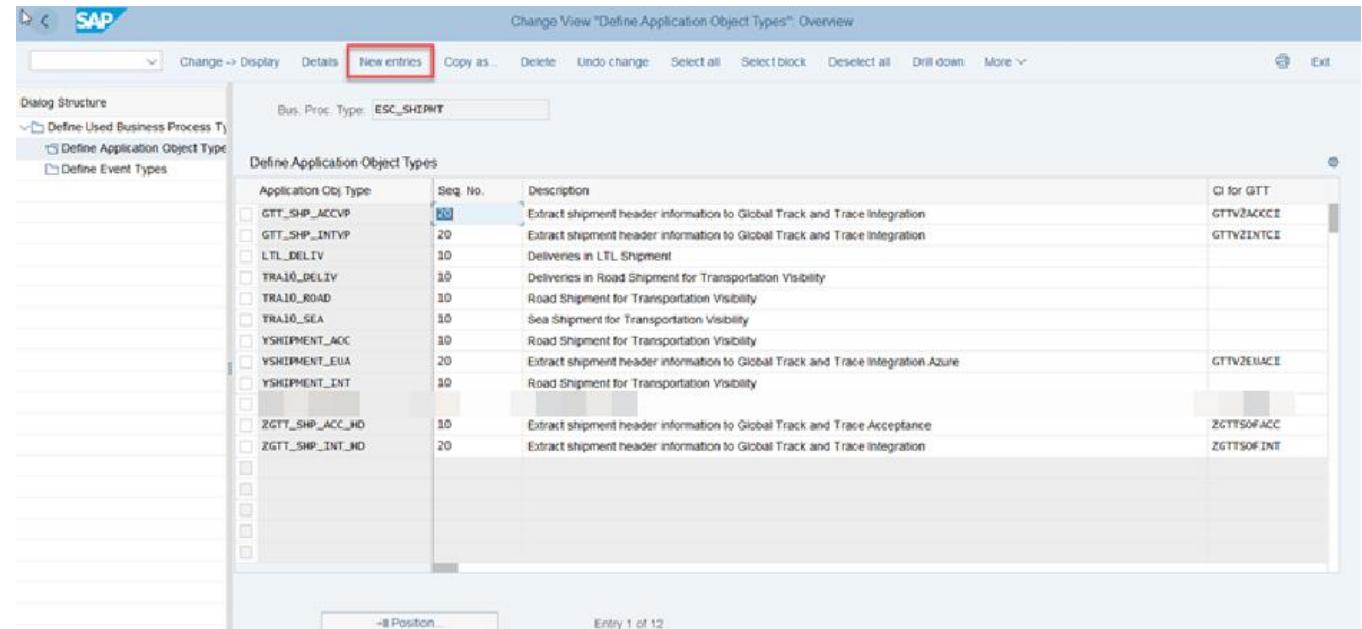
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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

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



The screenshot shows the SAP Fiori interface for defining application object types. The title bar reads "Change View 'Define Application Object Types'" Overview. The top navigation bar includes "Change -> Display", "Details", "New entries" (which is highlighted with a red box), "Copy as...", "Delete", "Undo change", "Select all", "Select block", "Deselect all", "Drill down", "More", and "Exit". The left sidebar shows a "Dialog Structure" tree with "Define Used Business Process Ty" (selected), "Define Application Object Type" (highlighted in blue), and "Define Event Types". The main area is titled "Define Application Object Types" and displays a table with the following data:

Application Obj. Type	Seq. No.	Description	QI for GTT
GTT_SHP_ACCVP	20	Extract shipment header information to Global Track and Trace Integration	GTTV2ACCCE
GTT_SHP_INTP	20	Extract shipment header information to Global Track and Trace Integration	GTTV2INTCE
LTL_DELIV	10	Deliveries in LTL Shipment	
TRA10_DELIV	20	Deliveries in Road Shipment for Transportation Visibility	
TRA10_ROAD	10	Road Shipment for Transportation Visibility	
TRA10_SEA	10	Sea Shipment for Transportation Visibility	
YSHIPMENT_AOC	20	Road Shipment for Transportation Visibility	
YSHIPMENT_EUA	20	Extract shipment header information to Global Track and Trace Integration Azure	GTTV2EUACE
YSHIPMENT_INT	20	Road Shipment for Transportation Visibility	
ZGTT_SHP_ACC_HD	20	Extract shipment header information to Global Track and Trace Acceptance	ZGTTSOFACC
ZGTT_SHP_INT_HD	20	Extract shipment header information to Global Track and Trace Integration	ZGTTSOFINIT

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 sections:

- 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 image contains two screenshots of SAP configuration interfaces. Both screenshots show the 'Control Tables' tab selected.

Business Process Type: ESC_SHIPMT

Appl. Obj. Type: ZGTT_SHP_INT_HD Extract shipment header information to Global Track and Trace Integration

Text: [Text input field]

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

Data Source for Created and Updated Objects

Main Obj. Table: SHIPMENT_HEADER_NEW **AOT on Header Level**

Master Table: [Text input field]

Data Source for Deleted Objects

Del.Obj. Table: SHIPMENT_HEADER_OLD

Reference Between Main and Master Table

First Field Reference from Main to Master Table

Business Process Type: ESC_DELIV

Appl. Obj. Type: ZGTT_DE_INT_ITEM Extract delivery order item information to Global Track and Trace Integration

Text: Delivery Item

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

Data Source for Created and Updated Objects

Main Obj. Table: DELIVERY_ITEM_NEW **AOT on Item Level**

Master Table: DELIVERY_HEADER_NEW

Data Source for Deleted Objects

Del.Obj. Table: DELIVERY_ITEM_OLD

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:** [Text input field]

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

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

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

The screenshot shows the SAP Fiori interface for defining business process types and application object types. The top section displays basic details: Bus. Proc. Type: ESC_DELIV, Appl. Obj. Type: ZGTT_DE_INT_ITEM, and Text: Delivery Item. Below this, tabs for General Data, Control Tables, Object Identification (which is selected), Global Track & Trace Relevance, and Parameter Setup are visible. The Object Identification tab contains sections for Method for determination of AOID and Application Object ID Source. The AOID Method dropdown is set to 'Determine from Field'. Under Application Object ID Source, two fields are defined: First Field to Build Appl. Obj. ID (Cntrl Tab. Type: 1 Main Object Table, AO ID Field: VBELN) and Second Field to Build Appl. Obj. ID (Cntrl Tab. Type: 1 Main Object Table, AO ID Field: POSNR). The Determine AOID By Function section contains an input field for the AOID Function. The entire 'Application Object ID Source' section is highlighted with a red box.

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

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

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

The screenshot shows a configuration interface for a business process. At the top, there are three input fields: 'Bus. Proc. Type' (ESC_DELIV), 'Appl. Obj. Type' (ZGTT_DE_INT_ITEM), and 'Text' (Delivery Item). Below these, a note says 'Extract delivery order item information to Global Track and Trace Integration'. A horizontal navigation bar at the bottom includes tabs for 'General Data', 'Control Tables', 'Object Identification', 'Global Track & Trace Relevance' (which is highlighted in blue), and 'Parameter Setup'. Under the 'Global Track & Trace Relevance' tab, there are two configuration fields: 'GTT Rel. Method' (set to 'A Check Function (Function Module)') and 'GTT Rel. Function' (containing the value 'ZGTT_SOF_DEITM'). The 'GTT Rel. Function' field is enclosed in a red rectangular box.

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

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

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

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

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

Click **Save**.

The screenshot shows the SAP Fiori interface for parameter setup. At the top, there are fields for Bus. Proc. Type (ESC_DELIV), Appl. Obj. Type (ZGTT_DE_INT_ITEM), 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 is set to 'A Determine by Function', and the Trk.ID Function is 'ZGTT_TID_DE_ITEM'. Under Parameter Setup, the Cntl Data Function is 'ZGTT_OTE_DE_ITEM', the Info Data Function is empty, and the Planned Event Function is 'ZGTT_EE_DE_ITM'. The entire TrkID Method section is highlighted with a red box.

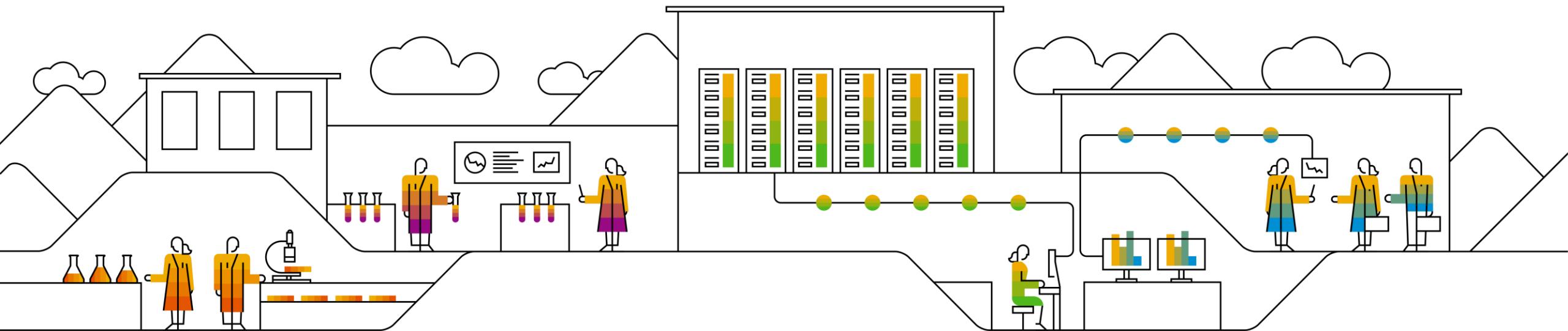
C) Download ABAP Code from GitHub

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

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

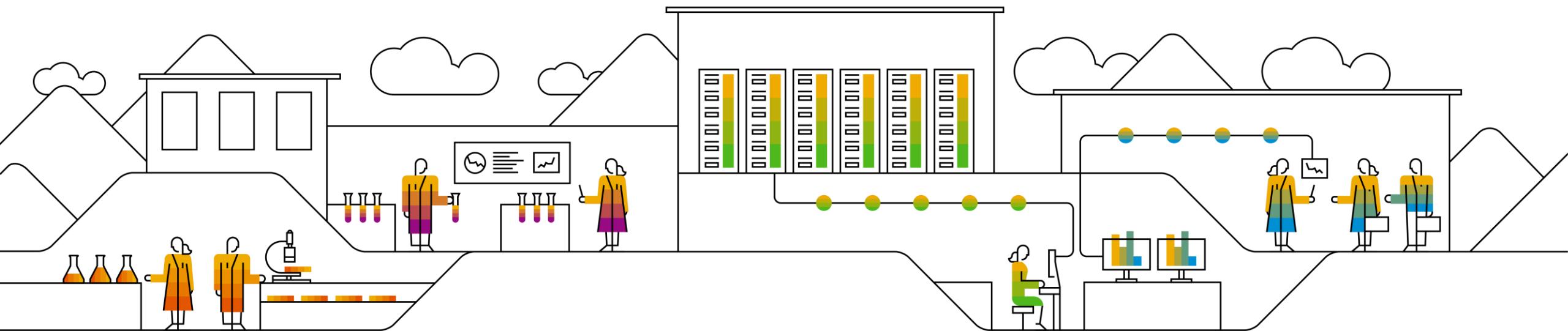
C3. Download Another ABAP Code from GitHub (Only for TPOF)

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



C) Download ABAP Code from GitHub

C1. Initial Download ABAP Code from GitHub (Onlyfor TSOF)



STEP 1: Install ABAPGit

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

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

 [Improve this page](#)

Summary #

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

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

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

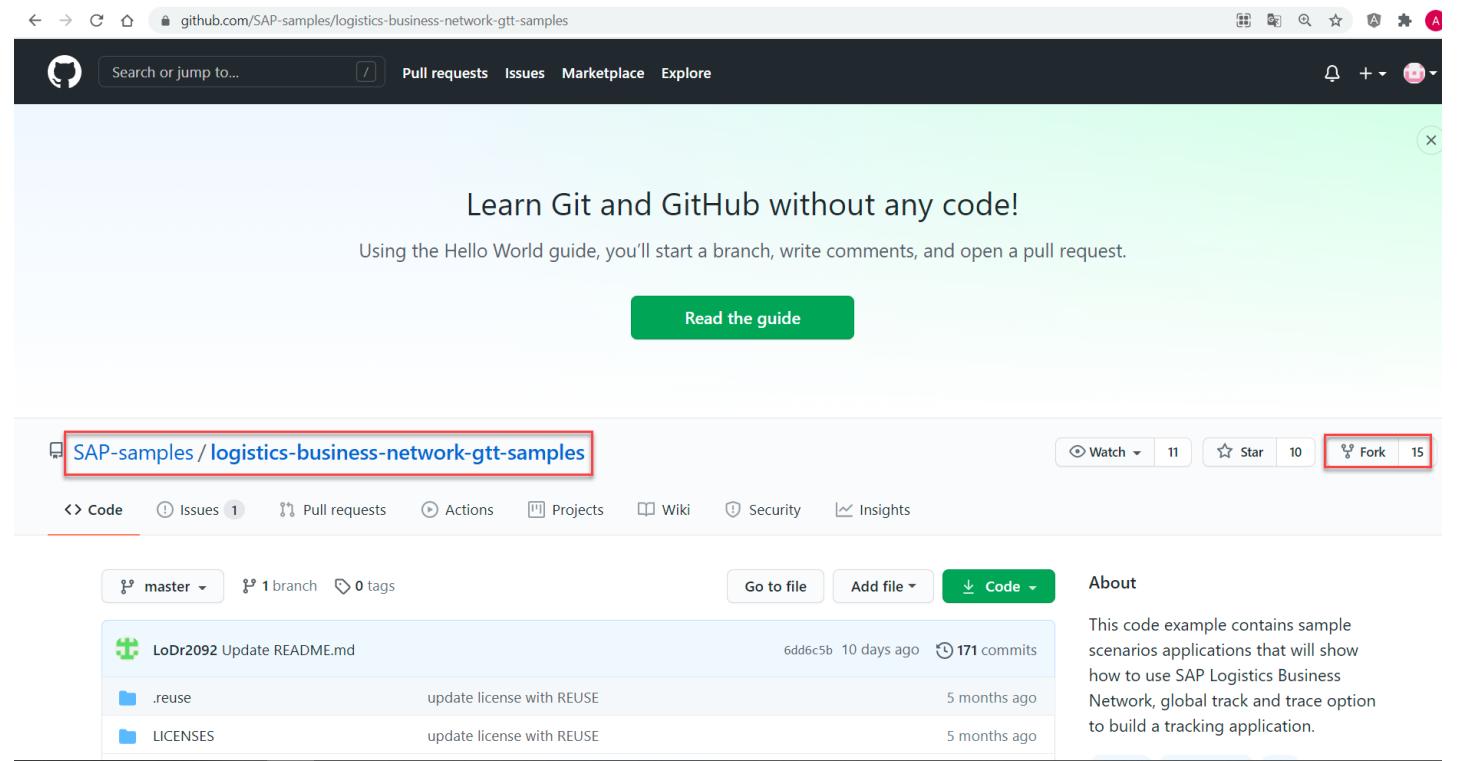
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

STEP 2: Fork Sample Code Repository

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

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



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

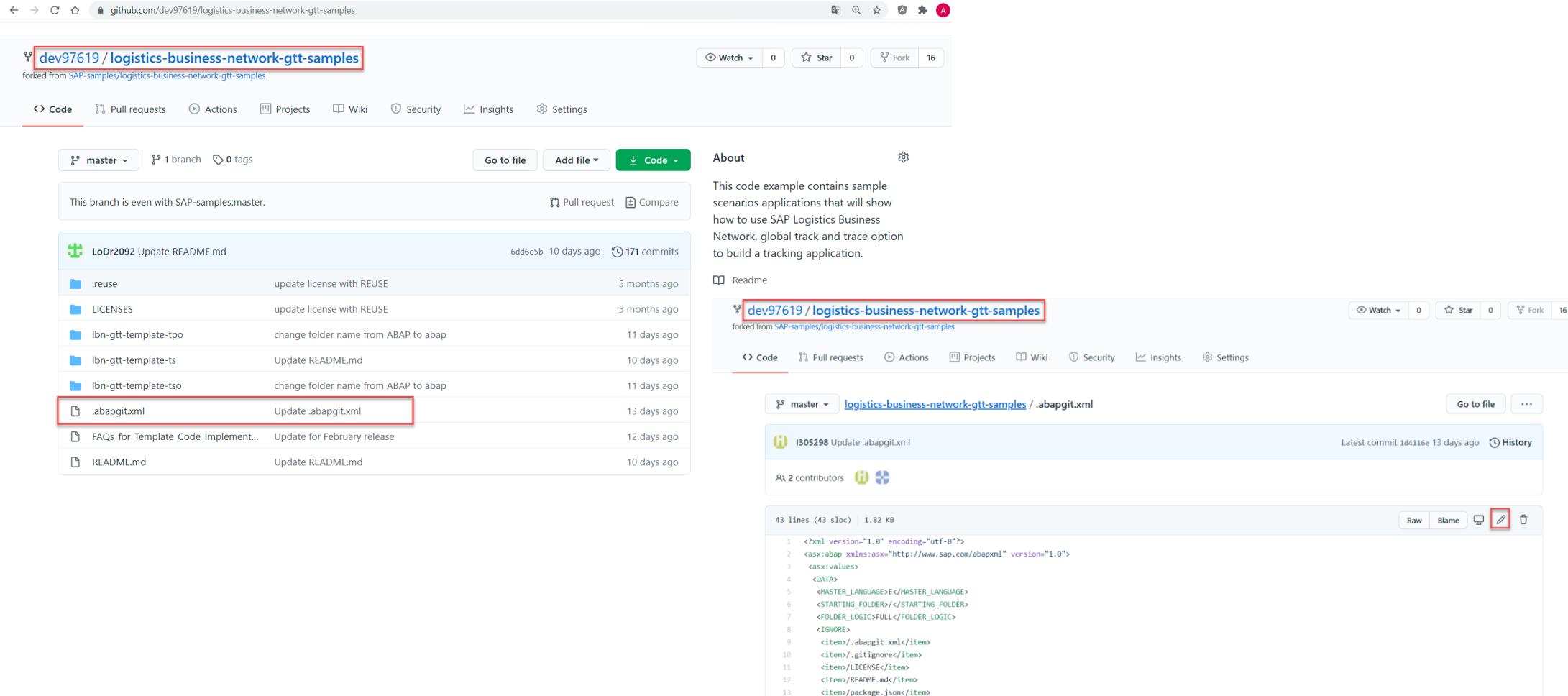
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one by LoDr2092 and another by the current user updating .abapgit.xml. The '.abapgit.xml' file is highlighted with a red box. The repository has 0 stars, 16 forks, and 171 commits. The 'About' section describes the code example as containing sample scenarios for SAP Logistics Business Network, global track and trace options. There are sections for 'Readme', 'Releases', and 'Packages', both of which are currently empty.

File	Description	Time
.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 'Code' tab is selected, showing the master branch 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 highlighted with a red box: 'I305298 Update .abapgit.xml' by 'LoDr2092' (6dd6c5b) 10 days ago. This commit is described as updating the configuration file. The right side of the screen shows the contents of the '.abapgit.xml' file, which is 43 lines long and 1.82 KB. The file content is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <sax:abap xmlns:sax="http://www.sap.com/abapxml" version="1.0">
3   <sax: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: Add the sentence of ‘<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>’ as below.

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 its XML content. A red box highlights the line '6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>'. To the right, a 'Commit changes' dialog is displayed. The 'Update .abapgit.xml' field contains the same highlighted line. The 'Commit directly to the master branch' radio button is selected. At the bottom, a large green 'Commit changes' button is highlighted with a red border.

```
<?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...

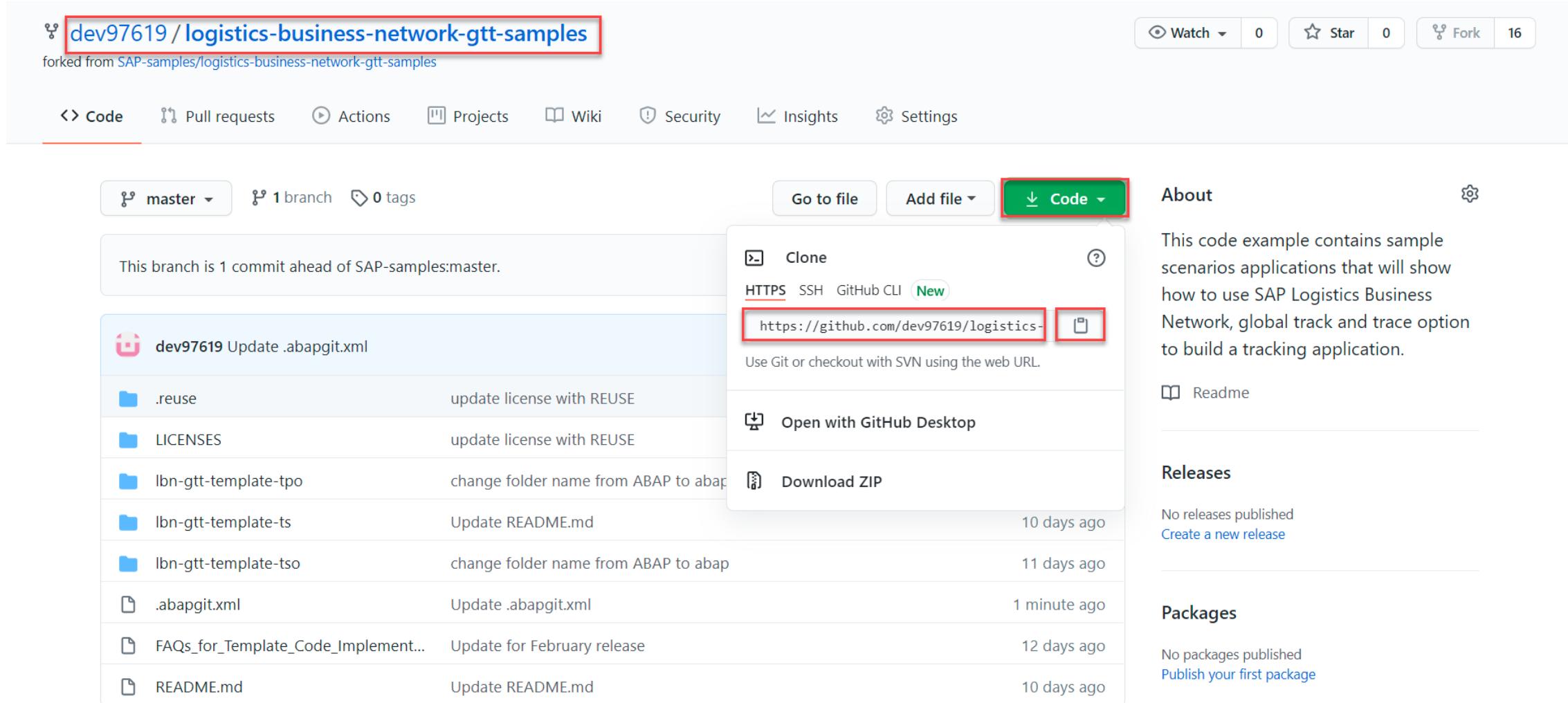
-o- Commit directly to the master branch.

! 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 0 stars and 16 forks. The 'Code' tab is selected. The repository has 1 branch and 0 tags. The master branch is ahead of SAP-samples:master by 1 commit. A dropdown menu is open over the 'Clone' link, which is highlighted with a red box. The URL 'https://github.com/dev97619/logistics...' is also highlighted with a red box. The 'Code' tab is also highlighted with a red box. The repository contains several files and folders, including '.reuse', 'LICENSES', 'lbn-gtt-template-tpo', 'lbn-gtt-template-ts', 'lbn-gtt-template-tso', '.abapgit.xml', 'FAQs_for_Template_Code_Implement...', and 'README.md'. The '.abapgit.xml' file was updated 1 minute ago. The 'About' section describes the repository as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options, and building a tracking application. It includes links for 'Readme', 'Releases', and 'Packages'.

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

lbn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

FAQs_for_Template_Code_Implement... Update for February release

README.md Update README.md

Go to file Add file Code

Clone HTTPS SSH GitHub CLI New

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

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published Create a new release

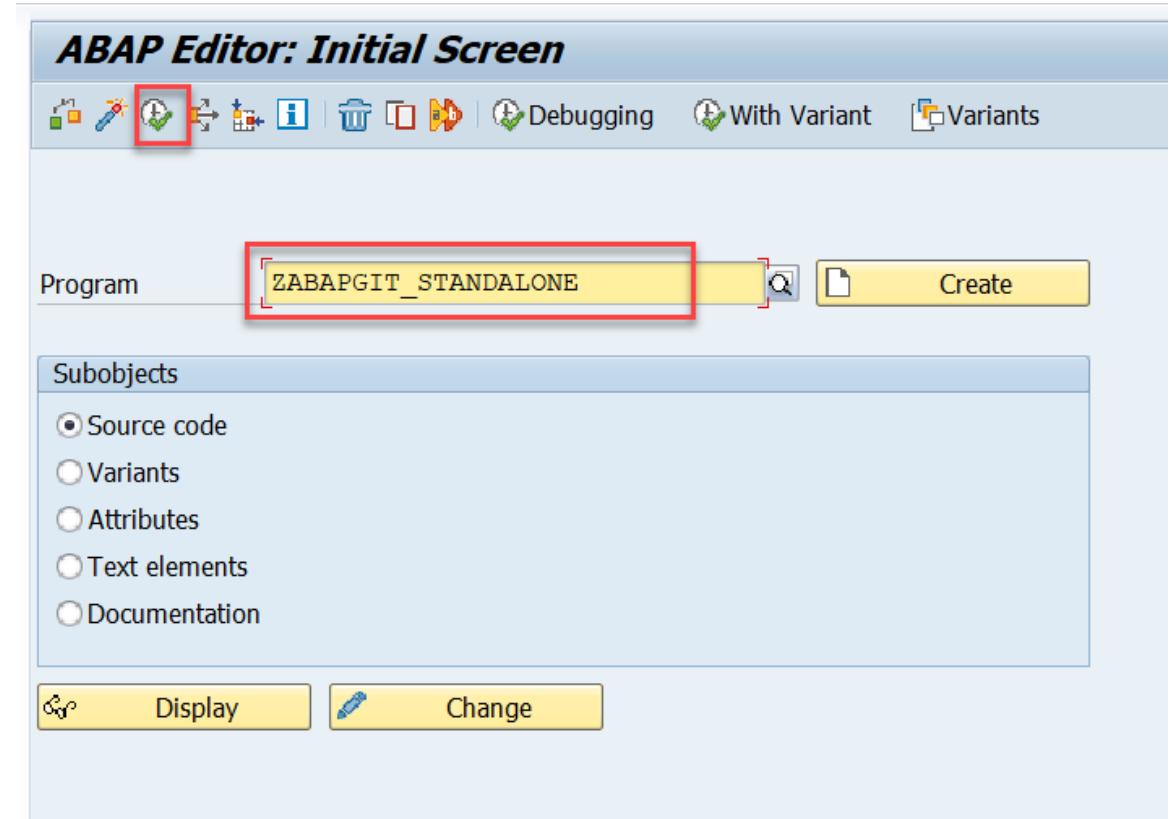
Packages

No packages published Publish your first package

STEP 4: Download ABAP Code from GitHub

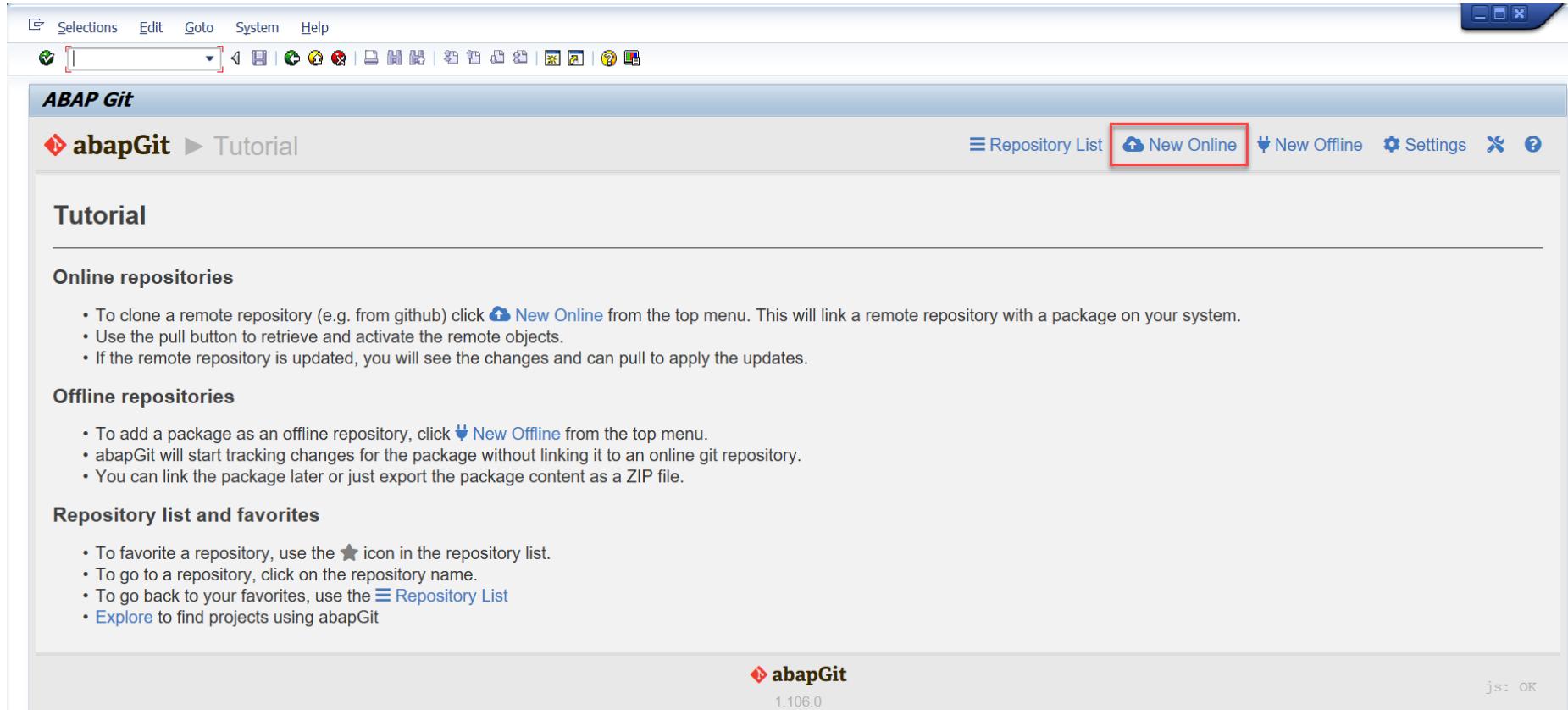
4-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

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

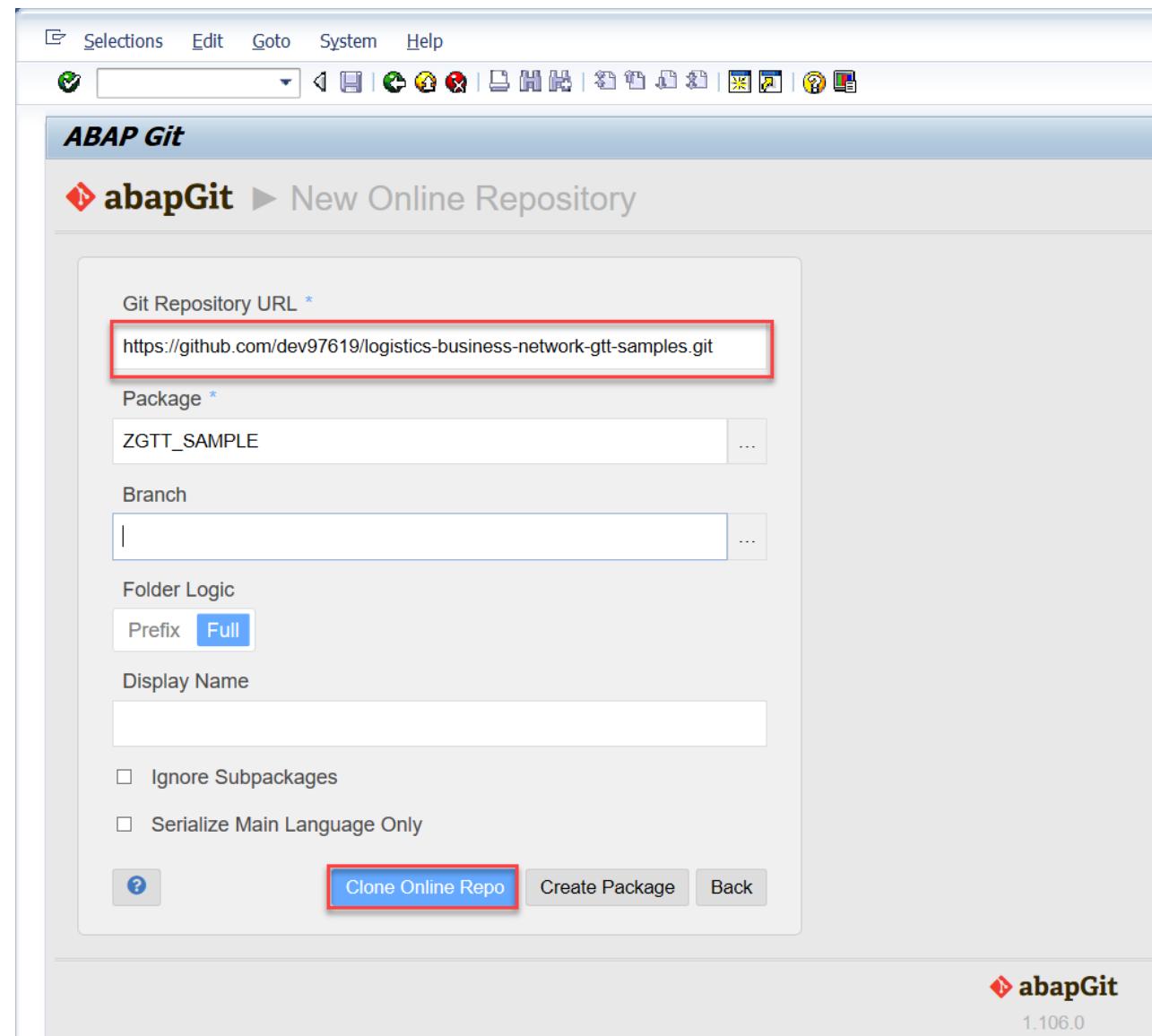
Caution:

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

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

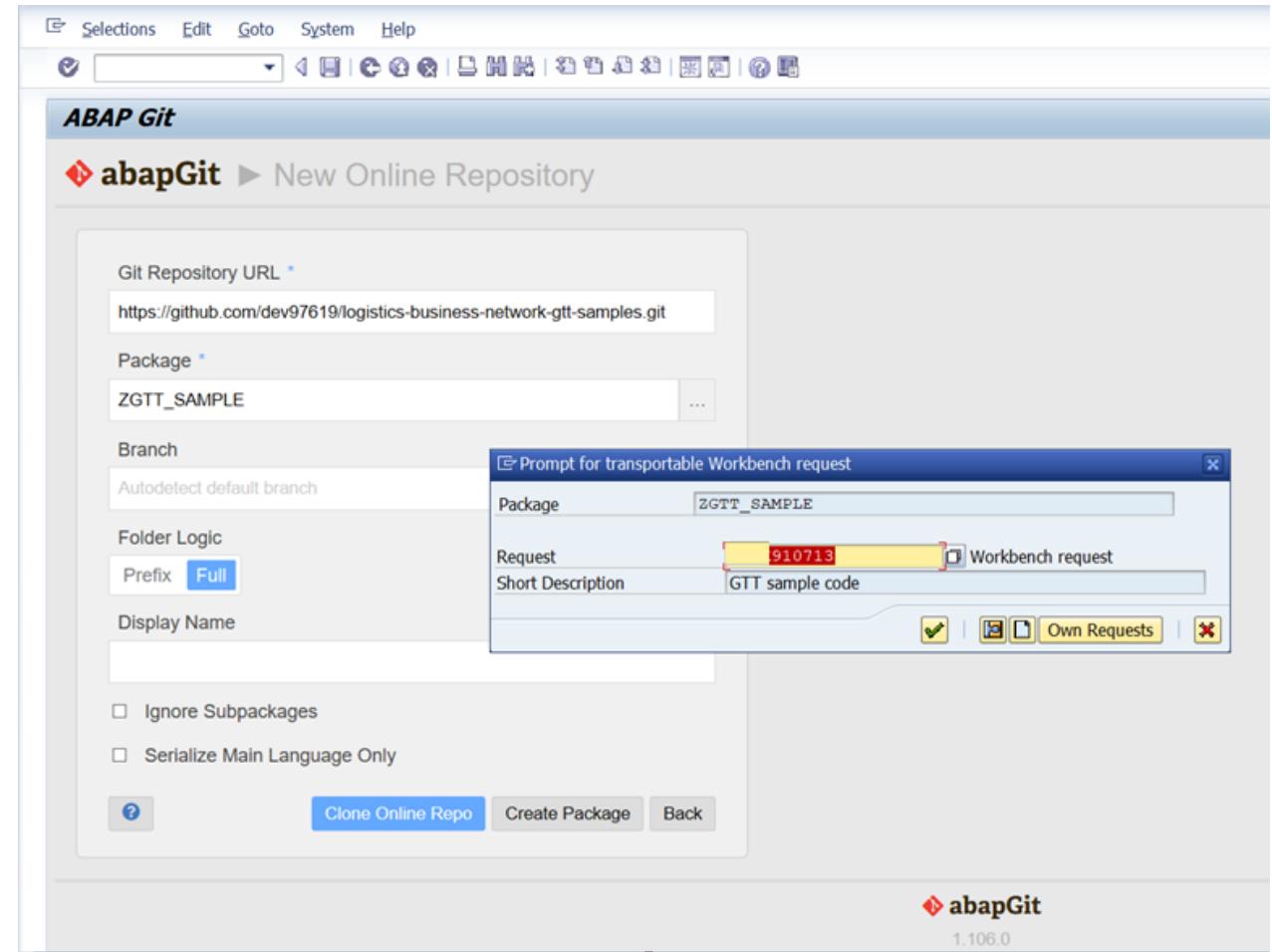
4-6: Set *Full* for **Folder Logic**

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



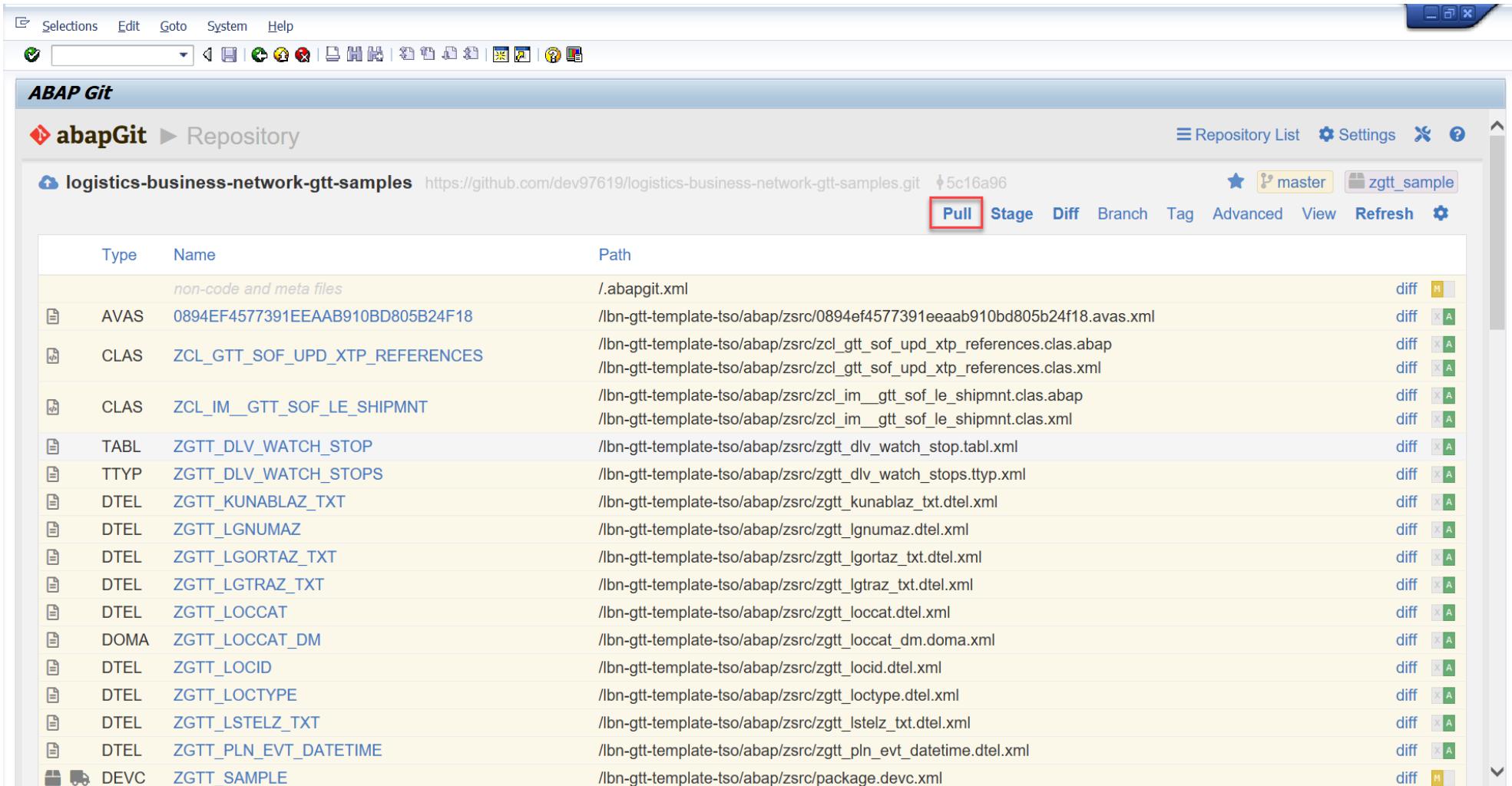
STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

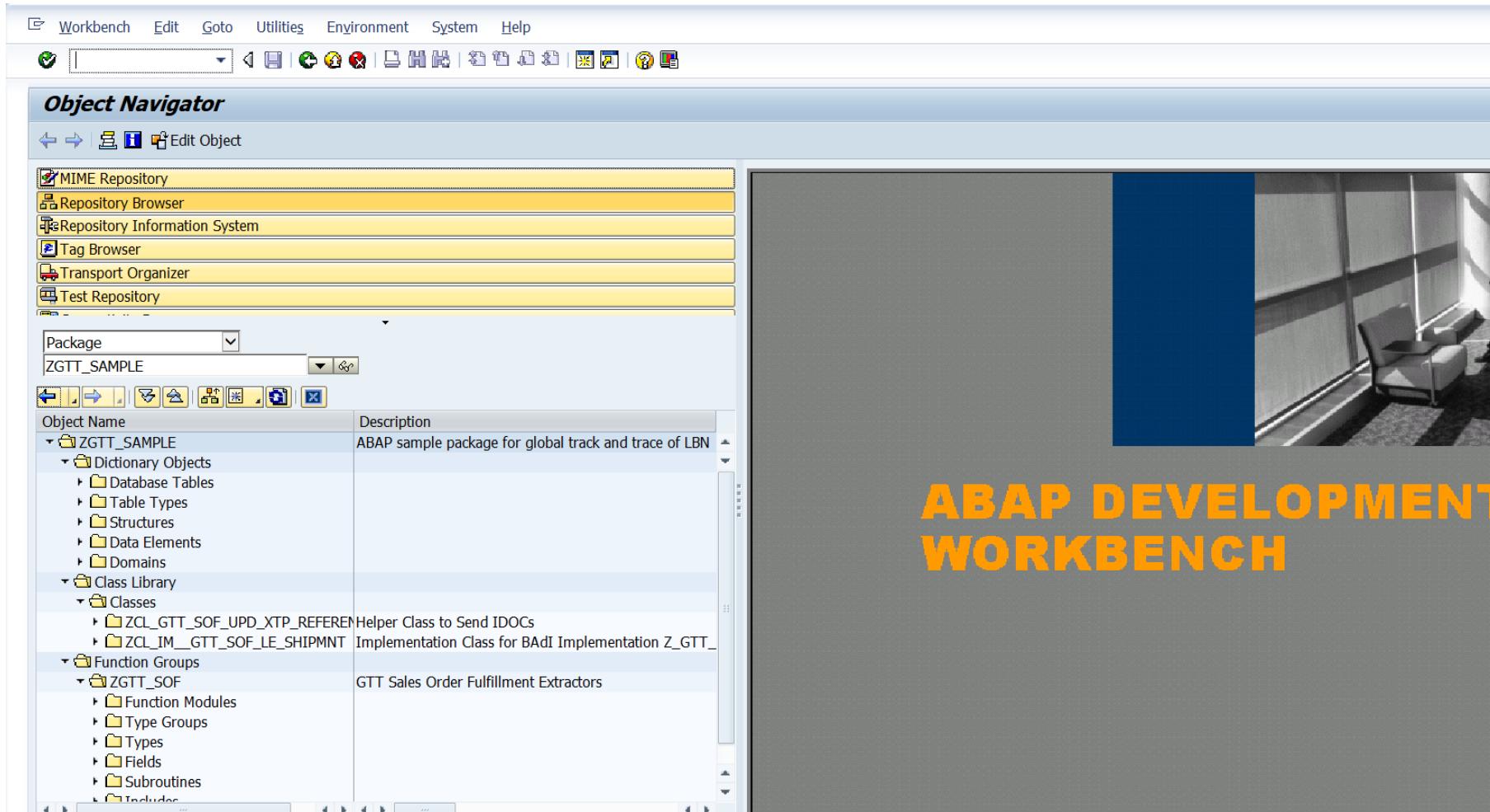


The screenshot shows the ABAP Git interface within an SAP application. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the repository 'abapGit > Repository'. The repository name 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 interface includes tabs for 'Repository List', 'Settings', and a refresh button. A prominent red box highlights the 'Pull' button in the top navigation bar. The main content is a table with columns 'Type', 'Name', 'Path', and 'diff'. The table lists numerous files and their paths, such as '.abapgit.xml', '0894EF4577391EEAAB910BD805B24F18', and 'ZGTT_SAMPLE'. Each row has a 'diff' link and a small icon.

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

STEP 4: Download ABAP Code from GitHub

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



C) Download ABAP Code from GitHub

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

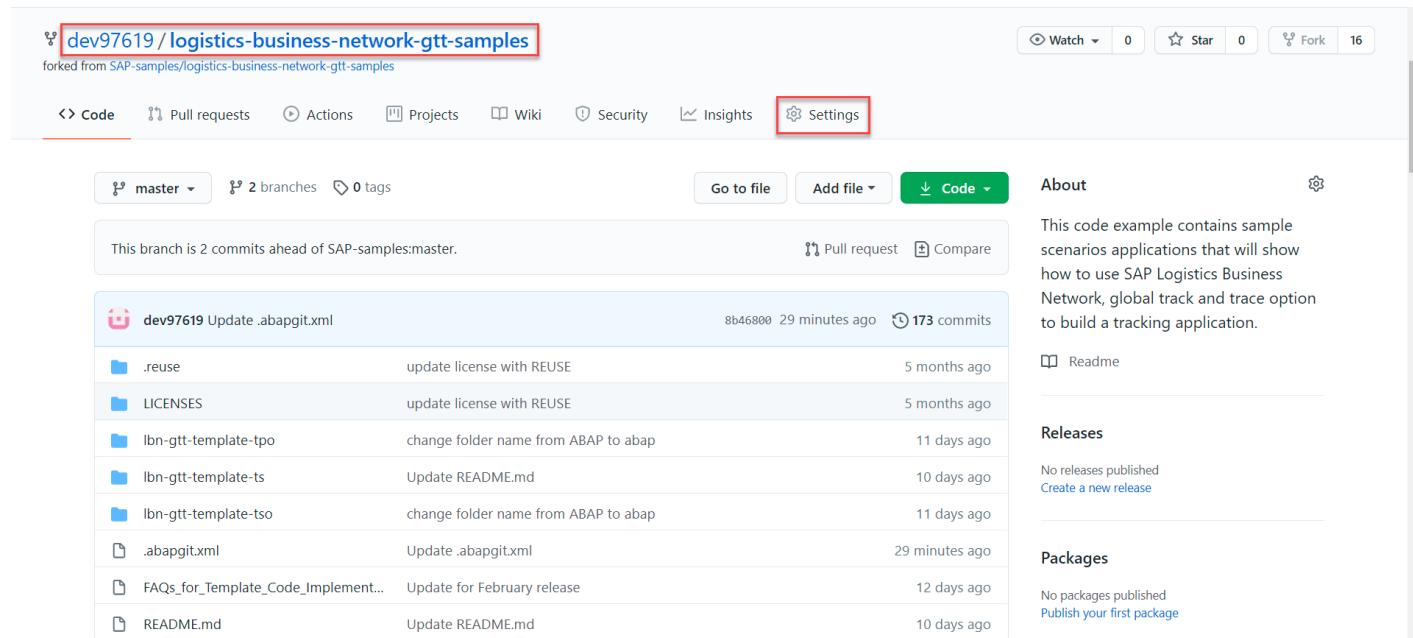


STEP 1: Delete the User's Account Repository

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

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

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



STEP 1: Delete the User's Account Repository

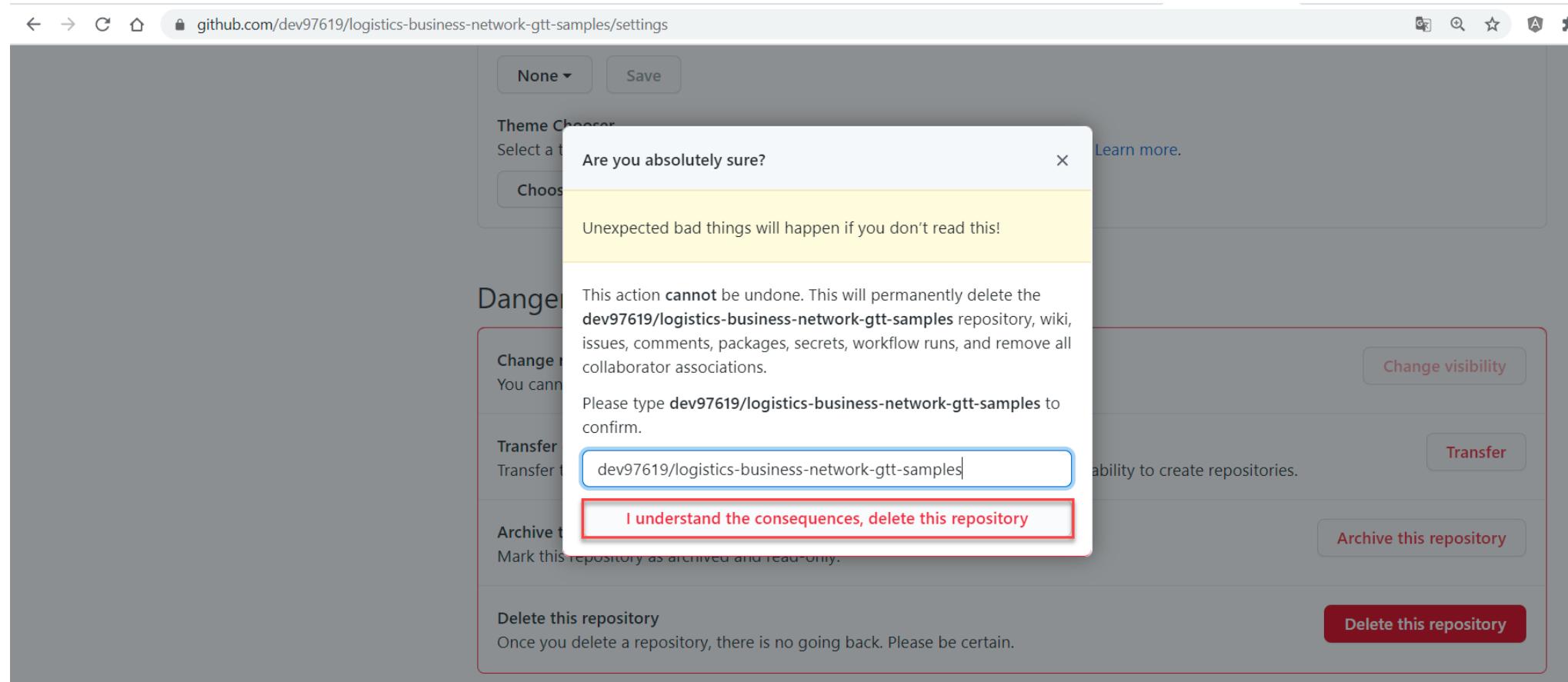
1-3: Scroll down and find the button “Delete this repository” and click it.

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

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

STEP 1: Delete the User's Account Repository

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



STEP 1: Delete the User's Account Repository

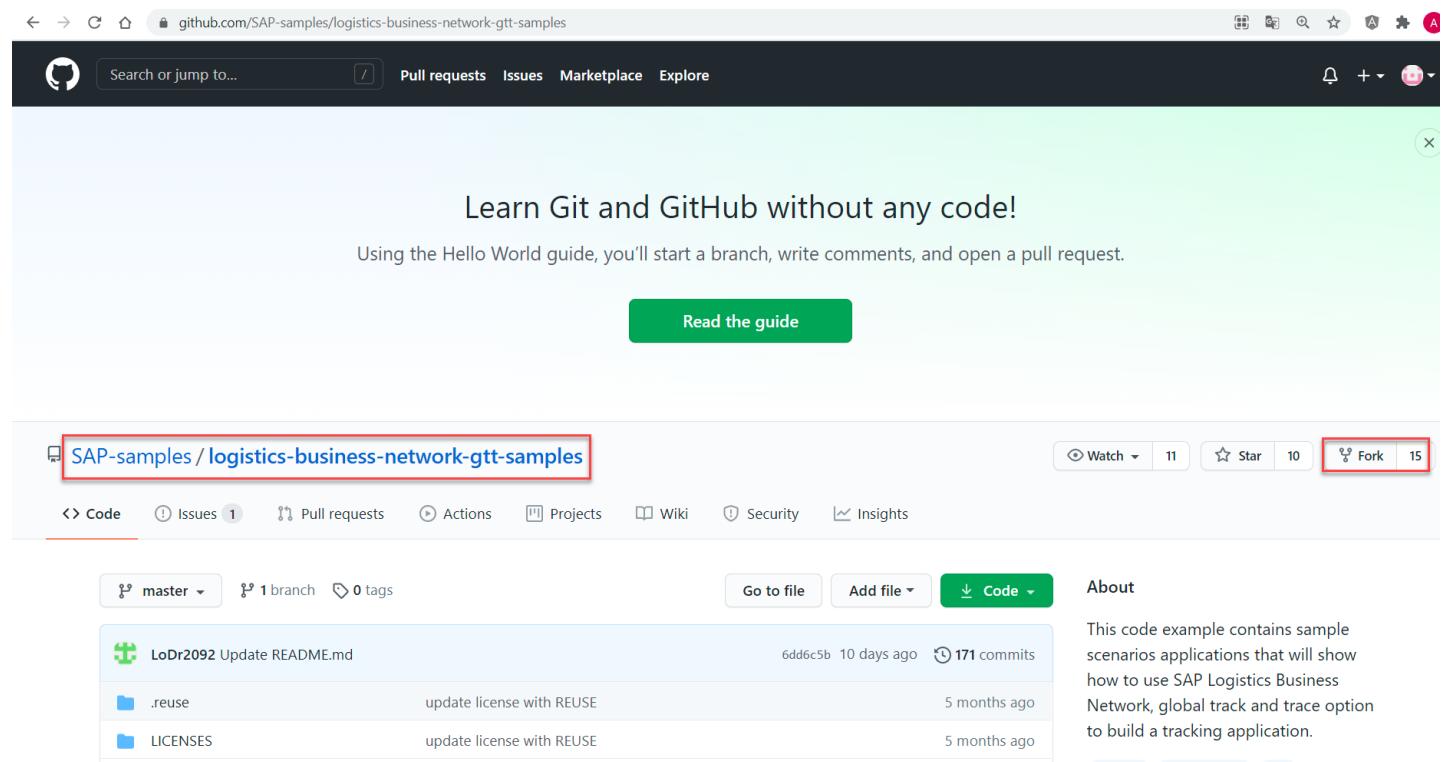
1-5: The user's account repository is deleted.

The screenshot shows the GitHub homepage. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. To the right of the search bar are notifications, a plus sign for creating new repositories, and a user profile icon. A prominent red rectangular box highlights a message in the main content area: "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." Below this message, there is a section titled "Create your first project" with a sub-section "Working with a team?". It includes buttons for "Create repository" (in green) and "Import repository". To the right, there is a large callout box with the heading "Learn Git and GitHub without any code!" and a description: "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." It features two buttons: "Read the guide" (in green) and "Start a project".

STEP 2: Fork Sample Code Repository

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

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



STEP 2: Fork Sample Code Repository

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

The screenshot shows a GitHub repository page. At the top, the URL is `github.com/dev97619/logistics-business-network-gtt-samples`. The repository name `dev97619 / logistics-business-network-gtt-samples` is highlighted with a red box. Below it, it says "forked from SAP-samples/logistics-business-network-gtt-samples". On the right, there are buttons for "Watch" (0), "Star" (0), and "Fork" (16). The main navigation bar includes "Code", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". The "Code" tab is selected and highlighted in green. Below the navigation, there are buttons for "master", "1 branch", "0 tags", "Go to file", "Add file", and "Code". A message says "This branch is even with SAP-samples:master." with buttons for "Pull request" and "Compare". The main content area displays a list of commits:

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

To the right of the commits, there is an "About" section with a detailed description of the code example, a "Readme" link, a "Releases" section (which is empty), and a "Packages" section (which is also empty).

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

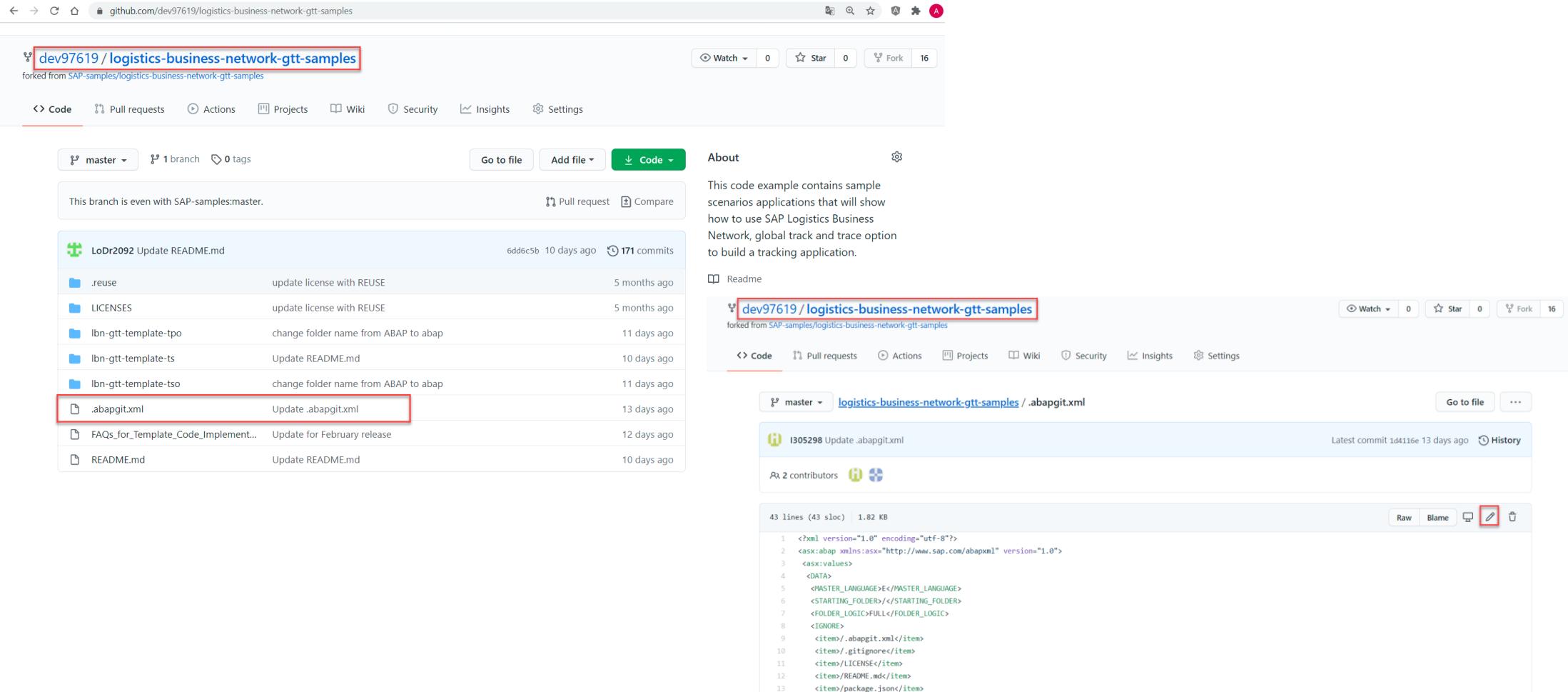
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The main content area displays a list of commits. A commit for '.abapgit.xml' is highlighted with a red box. The commit message is 'Update .abapgit.xml'. Other visible commits include 'Update README.md', 'update license with REUSE', and 'change folder name from ABAP to abap'. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages'.

Commit	Message	Date
LoDr2092 Update README.md	6dd6c5b 10 days ago	171 commits
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	11 days ago
Ibn-gtt-template-ts	Update README.md	10 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	11 days ago
.abapgit.xml	Update .abapgit.xml	13 days ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

STEP 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 0 stars, 16 forks, and 16 issues. The master branch has 1 branch and 0 tags. The README.md file states: "This branch is even with SAP-samples:master." There are 171 commits in the master branch. One commit, `I305298 Update .abapgit.xml`, is highlighted with a red box. The commit message is "Update .abapgit.xml". The .abapgit.xml file content is displayed below:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <sax:abap xmlns:sax="http://www.sap.com/abapxml" version="1.0">
3   <sax: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: Add the sentence of ‘<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>’ as below.

3-4: Commit change.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. On the left, the '.abapgit.xml' file is open, showing its XML content. A red box highlights the line '6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>'. On the right, a 'Commit changes' dialog is open, also with a red box highlighting the same line. The dialog includes fields for a commit message ('Update .abapgit.xml'), an optional description, and two radio button options for committing: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request.' Below the dialog are 'Commit changes' and 'Cancel' buttons.

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

Commit directly to the master branch.

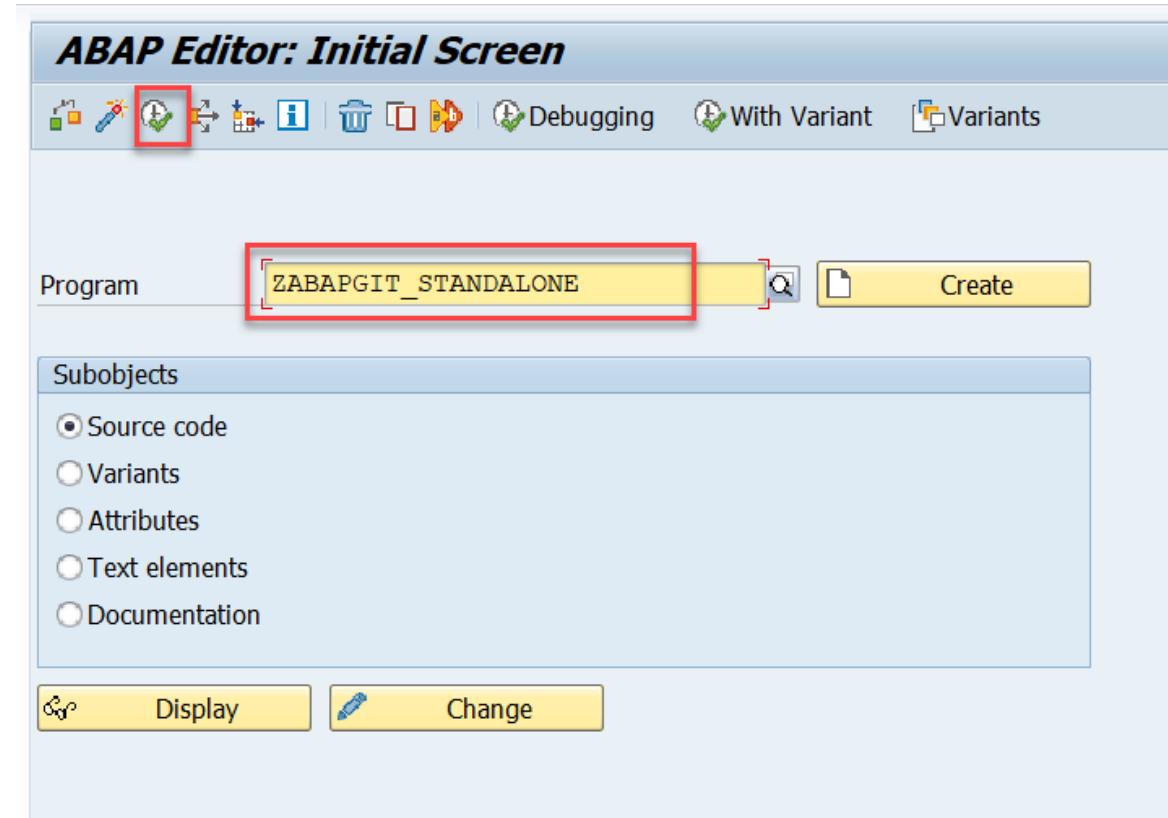
Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes Cancel

STEP 4: Update ABAP Code from GitHub

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

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



STEP 4: Update ABAP Code from GitHub

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



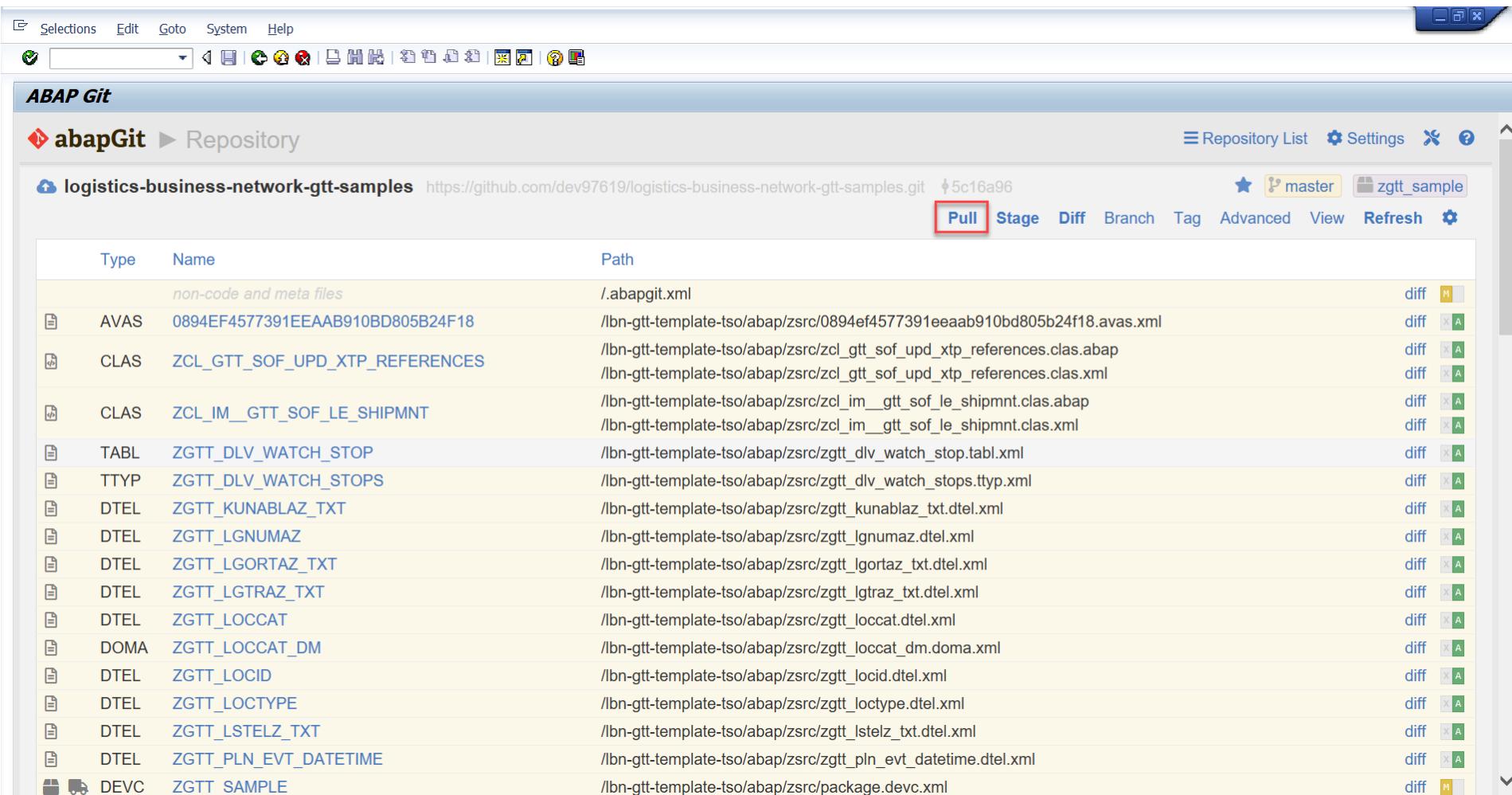
The screenshot shows the ABAP Git interface within a SAP application window. The title bar includes standard SAP menu items: Selections, Edit, Goto, System, Help. Below the menu is a toolbar with various icons. The main area is titled "ABAP Git" and displays a "Repository List". The list contains one entry:

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

At the bottom of the interface, there is a footer with the "abapGit" logo and version "1.106.0", and a status message "js: OK".

STEP 4: Update ABAP Code from GitHub

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

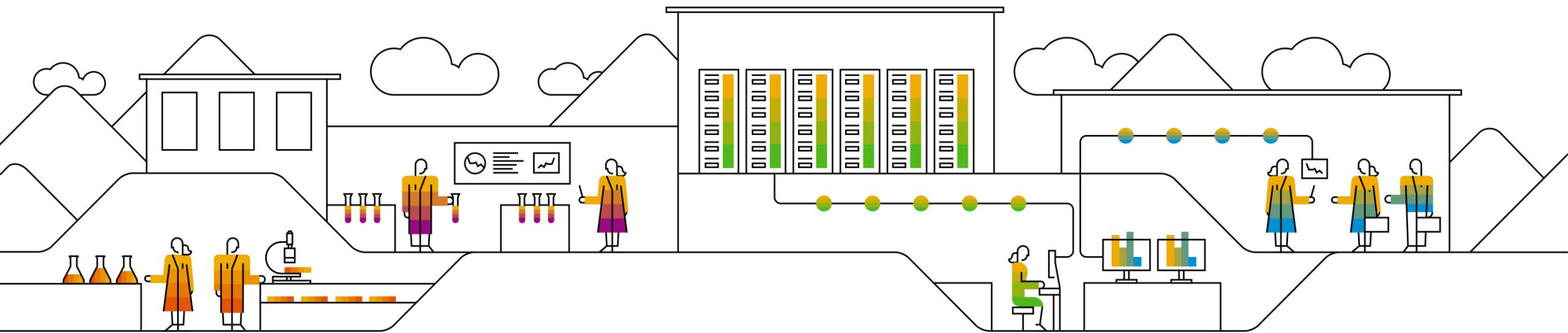


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git" and "abapGit Repository". Underneath, a repository card displays "logistics-business-network-gtt-samples" with its URL and a commit hash "5c16a96". To the right of the card are buttons for "master" and "zgtt_sample". A navigation bar below the card 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 "/.abapgit.xml", "/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml", and "/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap". Each row also has a "diff" button with a status indicator (M, A, or C).

Type	Name	Path	diff
	non-code and meta files	/	M
AVAS	0894EF4577391EEAAB910BD805B24F18	/.abapgit.xml /lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	A
CLAS	ZCL_GTT_SOF_UPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	A
CLAS	ZCL_IM_GTT_SOF_LE_SHIPMNT	/lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	A
TABL	ZGTT_DLV_WATCH_STOP	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	A
TTYP	ZGTT_DLV_WATCH_STOPS	/lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	A
DTEL	ZGTT_KUNABLAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_kunablaTxt.dtel.xml	A
DTEL	ZGTT_LGNUMAZ	/lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml	A
DTEL	ZGTT_LGORAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgorazTxt.dtel.xml	A
DTEL	ZGTT_LGTRAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgtrazTxt.dtel.xml	A
DTEL	ZGTT_LOCAT	/lbn-gtt-template-tso/abap/zsrc/zggt_locat.dtel.xml	A
DOMA	ZGTT_LOCAT_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_lstelzTxt.dtel.xml	A
DTEL	ZGTT_PLN_EVT_DATETIME	/lbn-gtt-template-tso/abap/zsrc/zggt_plnEvtDatetime.dtel.xml	A
DEV	ZGTT_SAMPLE	/lbn-gtt-template-tso/abap/zsrc/package.devC.xml	M

C) Download ABAP Code from GitHub

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



STEP 1: Fork Sample Code Repository

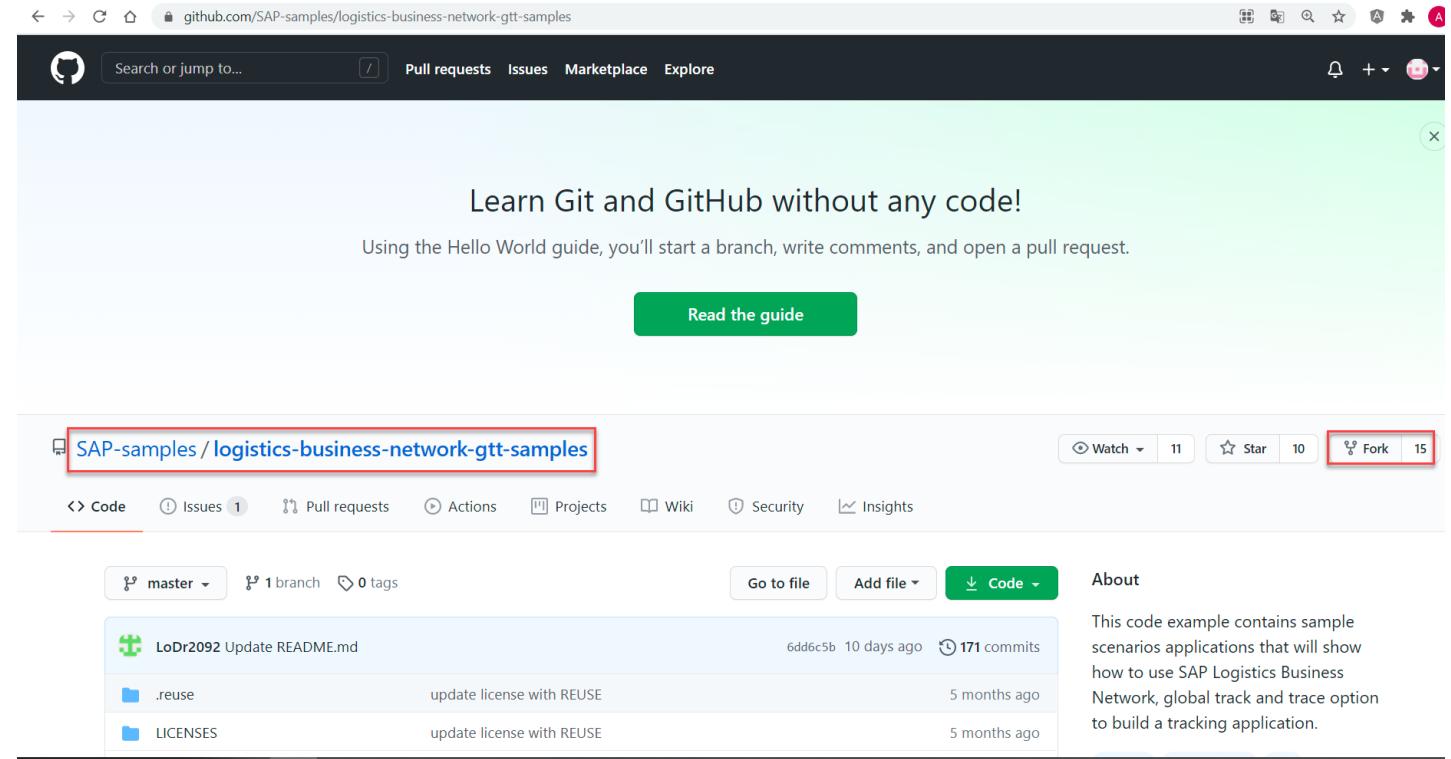
Prerequisite:

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

To install the TPOF do the following:

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

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



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

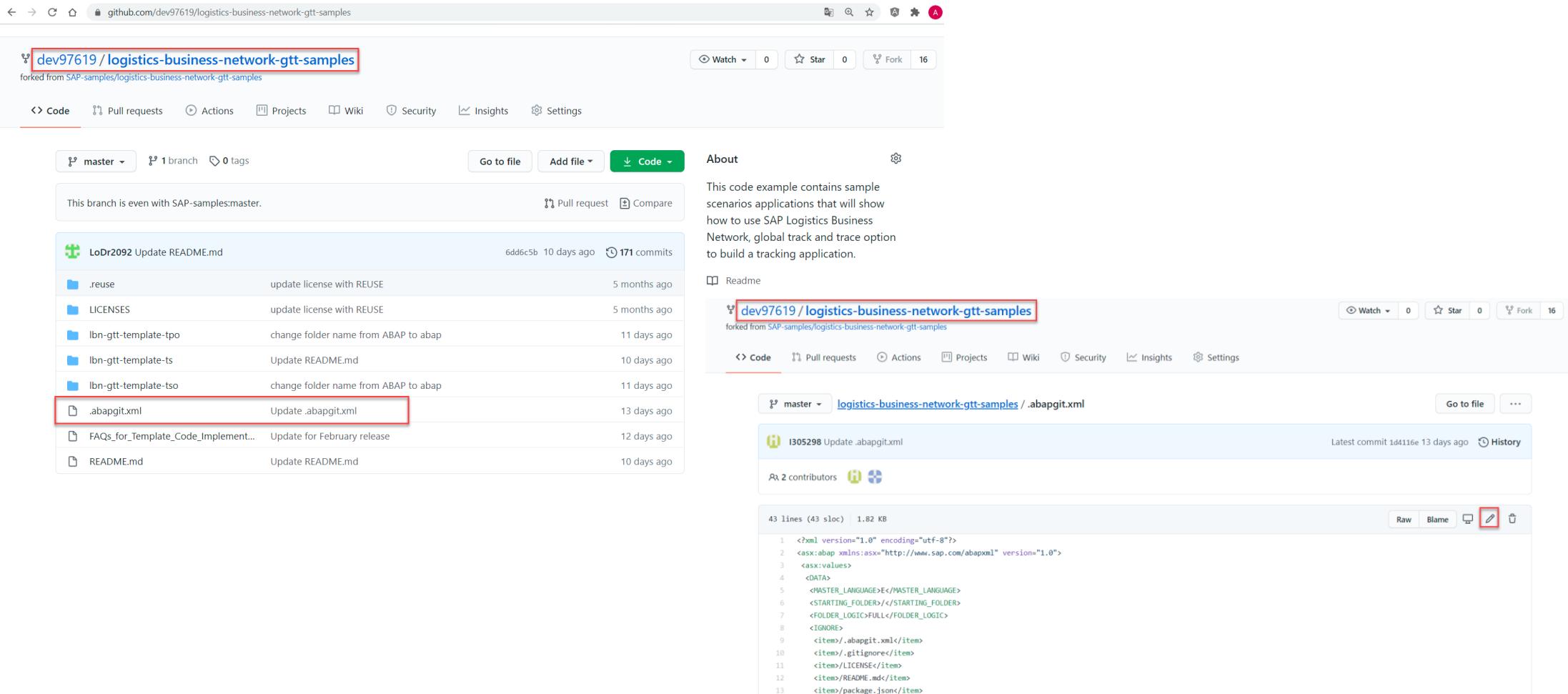
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one by LoDr2092 and another by the current user updating .abapgit.xml. The '.abapgit.xml' file is highlighted with a red box. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages', all of which show no activity.

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 2: Change Configuration File ‘.abapgit.xml’

2-2: 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 'Code' tab is selected, showing the master branch with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. Below is a list of commits:

- LoDr2092 Update README.md (6dd6c5b, 10 days ago, 171 commits)
- .reuse update license with REUSE (5 months ago)
- LICENSES update license with REUSE (5 months ago)
- lbn-gtt-template-tpo change folder name from ABAP to abap (11 days ago)
- lbn-gtt-template-ts Update README.md (10 days ago)
- lbn-gtt-template-tso change folder name from ABAP to abap (11 days ago)
- .abapgit.xml Update .abapgit.xml** (13 days ago) - This commit is highlighted with a red box.
- FAQs_for_Template_Code_Implement... Update for February release (12 days ago)
- README.md Update README.md (10 days ago)

The commit 'Update .abapgit.xml' is detailed in the right panel:

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

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

1305298 Update .abapgit.xml (Latest commit 1d4116e 13 days ago) History

2 contributors

43 lines (43 sloc) | 1.82 KB

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <sax:abap xmlns:sax="http://www.sap.com/abapxml" version="1.0">
3   <sax: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: Add the sentence of ‘<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/ </STARTING_FOLDER>’ as below.

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 includes the line '<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>' which is highlighted with a red box. The commit message input field contains 'Update .abapgit.xml'.

Code

Watch 0

Star 0

Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

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

Edit file Preview changes

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

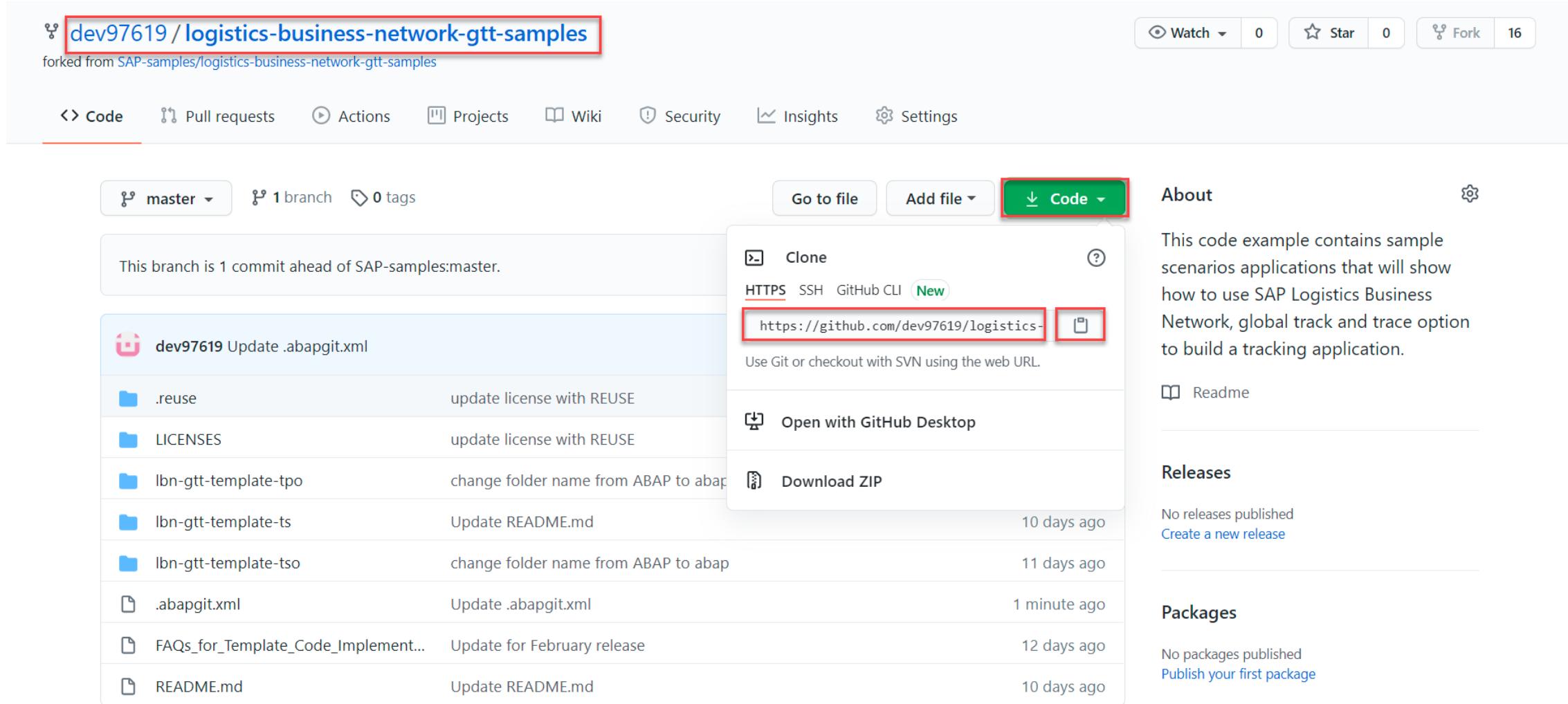
Commit directly to the master branch.

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 0 stars and 16 forks. The 'Code' tab is selected. The repository has 1 branch and 0 tags. The 'About' section describes it as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options. The 'Clone' section shows the HTTPS URL: <https://github.com/dev97619/logistics-business-network-gtt-samples>. The repository contains several files and folders, including '.reuse', 'LICENSES', 'lbn-gtt-template-tpo', 'lbn-gtt-template-ts', 'lbn-gtt-template-tso', '.abapgit.xml', 'FAQs_for_Template_Code_Implementation...', and 'README.md'. The '.abapgit.xml' file was updated 1 minute ago.

dev97619 / **logistics-business-network-gtt-samples**

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

Watch 0 ⌂ Star 0 ⌂ Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

lbn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

FAQs_for_Template_Code_Implementation... Update for February release

README.md Update README.md

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI New

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

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published Create a new release

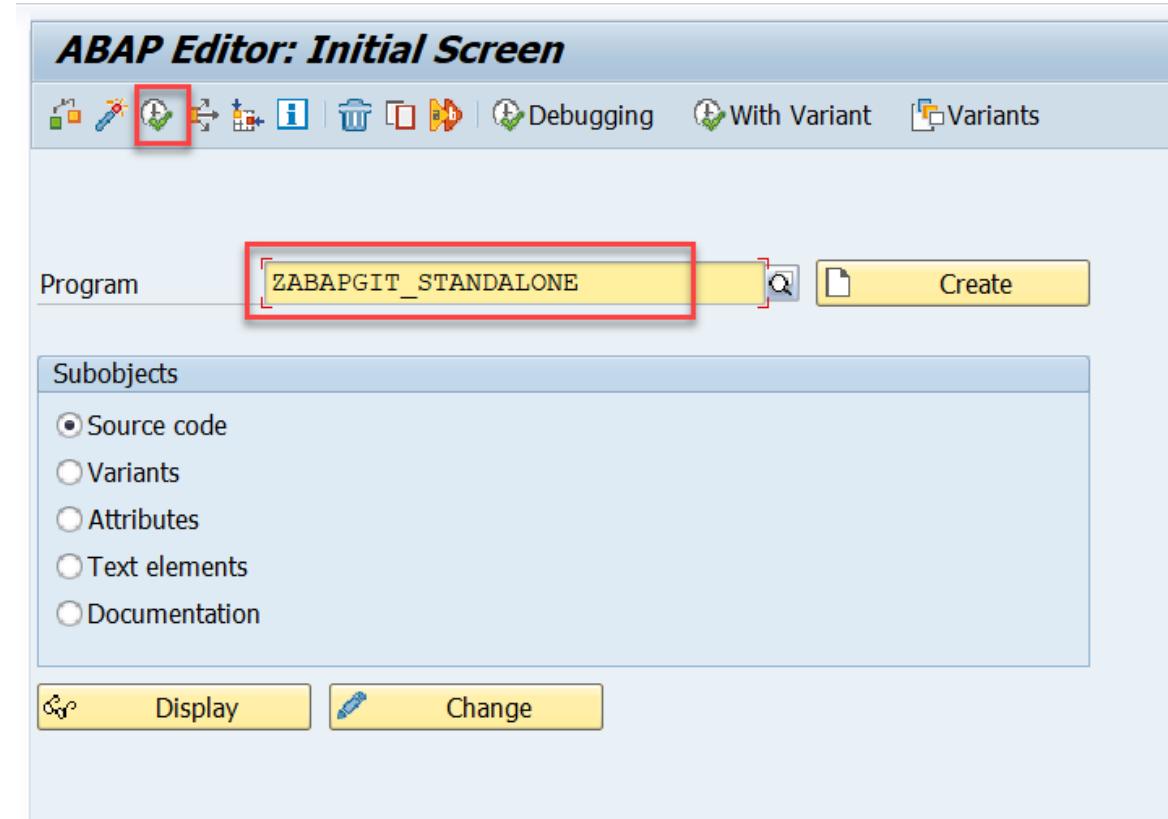
Packages

No packages published Publish your first package

STEP 3: Remove TSOF Repository in ABAPGit

3-1: Enter T-code SE38 and fill in the report name **ZABAPGIT_STANDALONE**.

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



STEP 3: Remove TSOF Repository in ABAPGit

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

The screenshot shows the ABAPGit interface. At the top, there is a toolbar with various icons and a menu bar with Selections, Edit, Goto, System, and Help. Below the toolbar is a navigation bar with a search field, a refresh icon, and several other icons. The main title is "ABAP Git". The sub-title is "abapGit ► Repository List". On the right side of the sub-title are buttons for New Online, New Offline, Settings, and Help. Below the sub-title is a filter bar with a "Filter:" input field, a "Only Favorites" checkbox, and a "Detail" checkbox. The main area is a table with the following columns: Name, Url, Package, Branch, and Action. There is one row visible:

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

At the bottom center is the "abapGit" logo with the version "1.106.0". To the right of the logo is the text "js: OK".

STEP 3: Remove TSOF Repository in ABAPGit

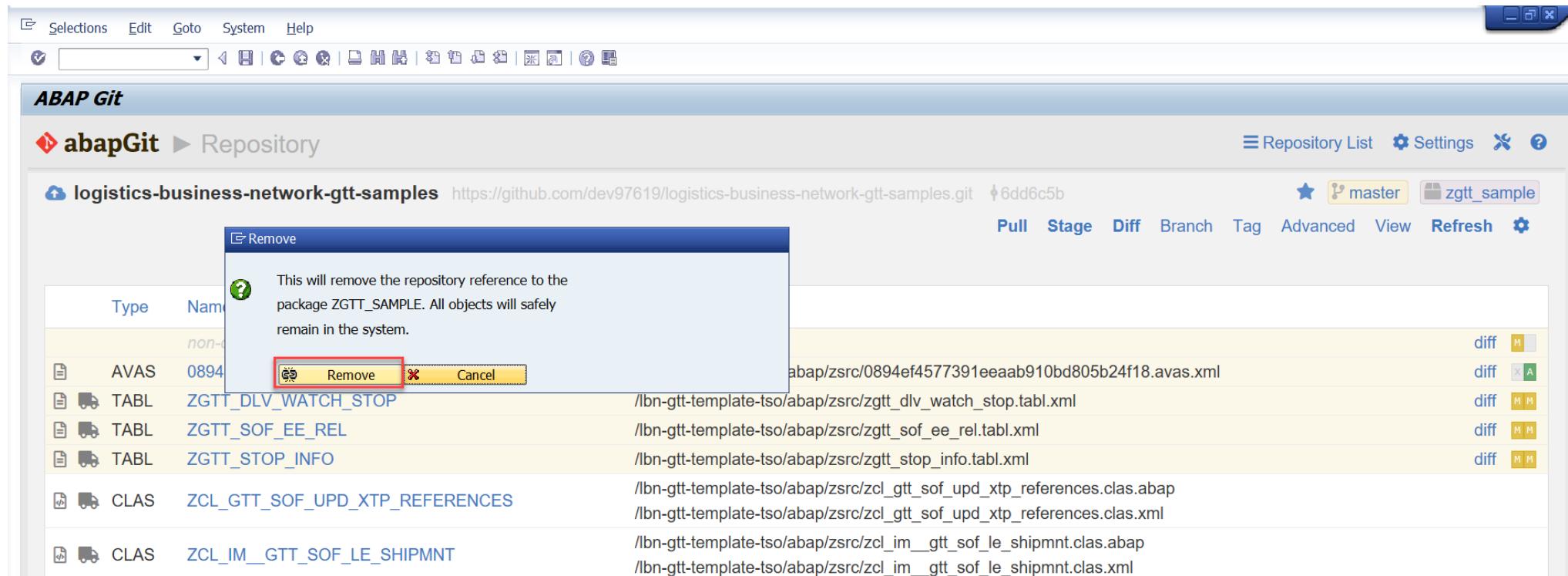
3-4: Under the “Advanced” menu, choose and click “Remove”.

The screenshot shows the ABAPGit interface for managing repositories. The repository listed is "logistics-business-network-gtt-samples" from GitHub. The "Advanced" menu is open, and the "Remove" option is highlighted with a red box. The menu also includes other options like "Reset Local (Force Pull)", "Checkout commit", "Background Mode", etc.

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

STEP 3: Remove TSOF Repository in ABAPGit

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

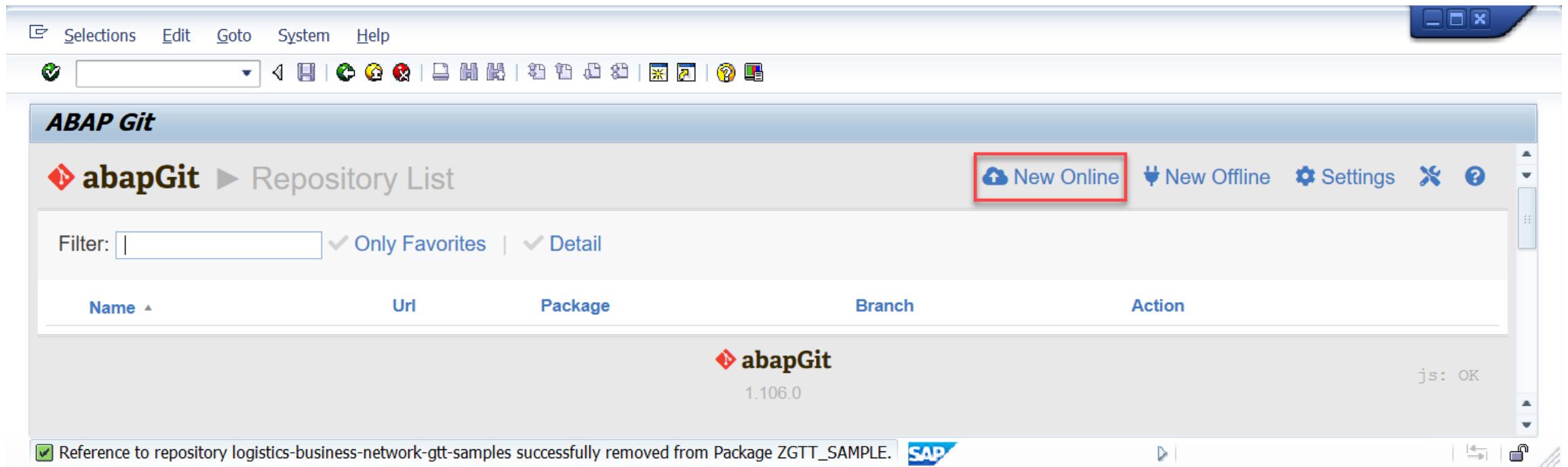


3-6: After repository removal you will see the following message:



STEP 4: Download TPOF Code from GitHub

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



STEP 4: Download TPOF Code from GitHub

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

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

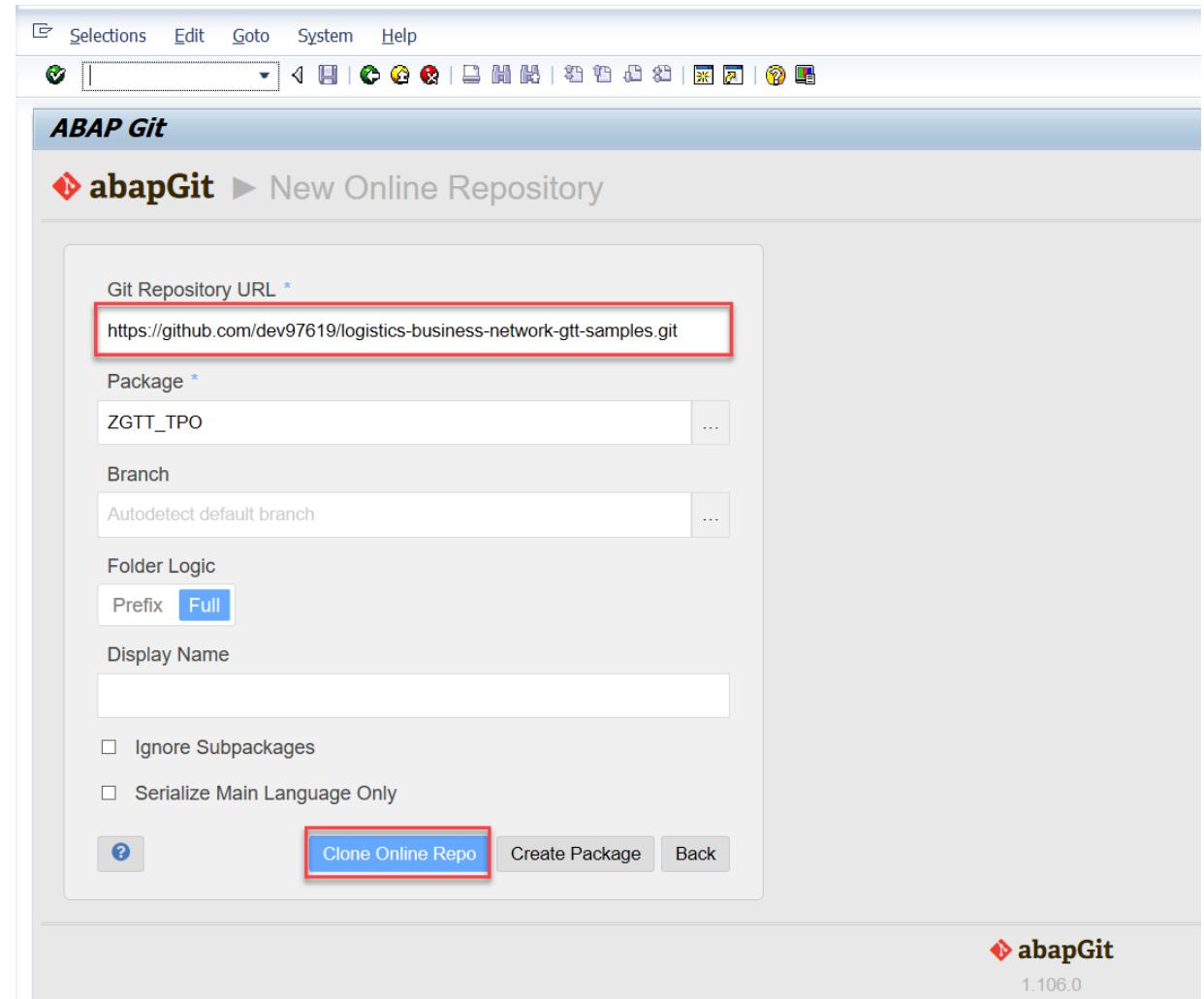
Caution:

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

4-3: Fill in the **Package** where you want to create the new ABAP code. If the package does not exist yet, click **Create package** to create it.

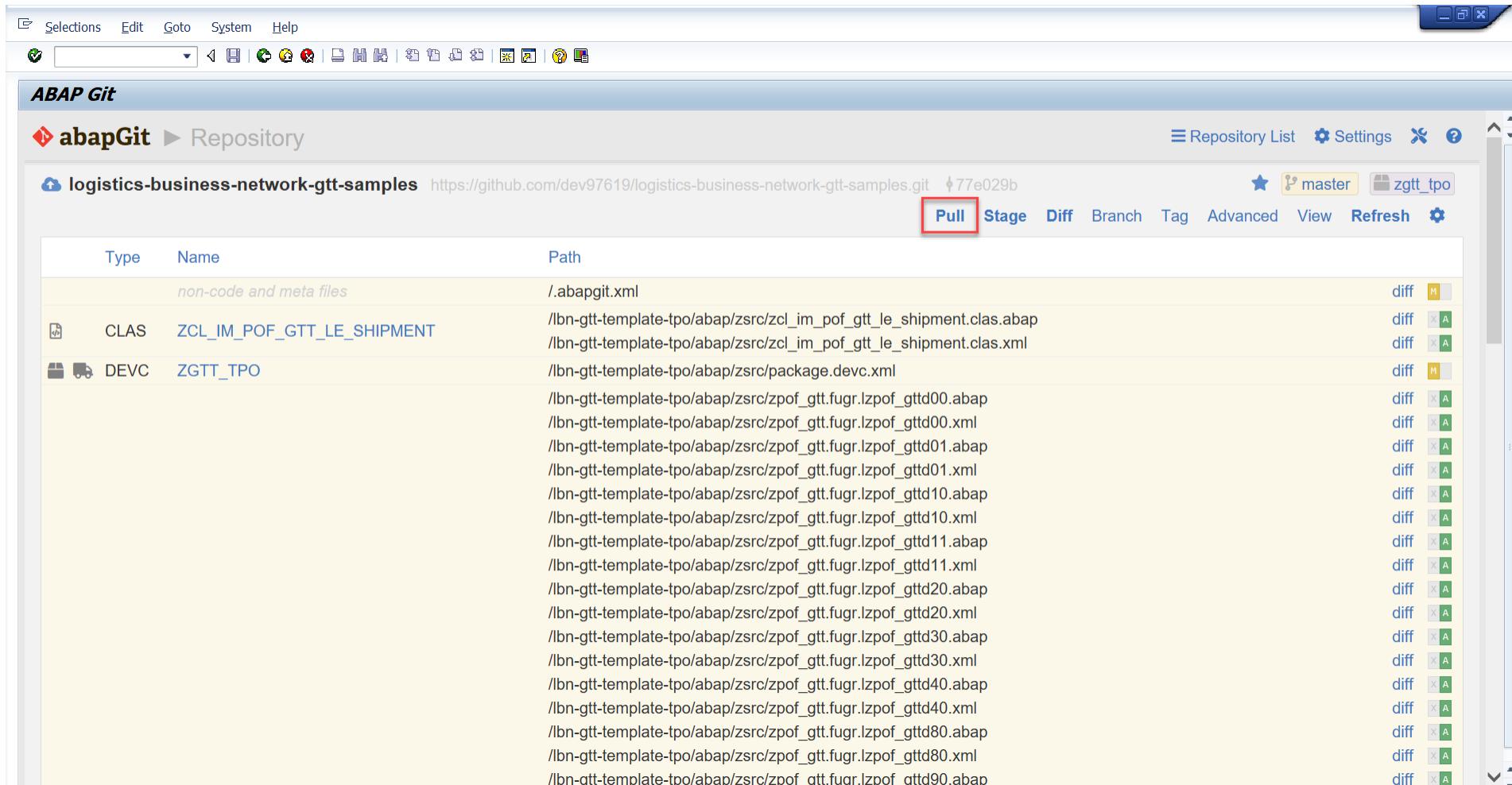
4-4: Set **Full** for **Folder Logic**

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



STEP 4: Download ABAP Code from GitHub

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



The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation "abapGit > Repository". The main area displays a table of files under the repository "logistics-business-network-gtt-samples". The table has columns for Type, Name, Path, and Diff. A red box highlights the "Pull" button in the toolbar above the table. The table shows several entries, including a "non-code and meta files" entry for ".abapgit.xml" and two entries under "CLAS" and "DEVC" types respectively.

Type	Name	Path	Diff
non-code and meta files			
		/.abapgit.xml	diff [H]
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap /lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	diff [A] diff [A]
DEVC	ZGTT_TPO	/lbn-gtt-template-tpo/abap/zsrc/package.devc.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd00.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd00.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd01.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd01.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd10.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd10.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd11.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd11.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd20.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd20.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd30.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd30.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd40.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd40.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd80.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd80.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd90.abap	diff [A] diff [A]

C) Download ABAP Code from GitHub

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



STEP 1: Install ABAPGit

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

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

 [Improve this page](#)

Summary #

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

- The standalone version is targeted at users. It 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

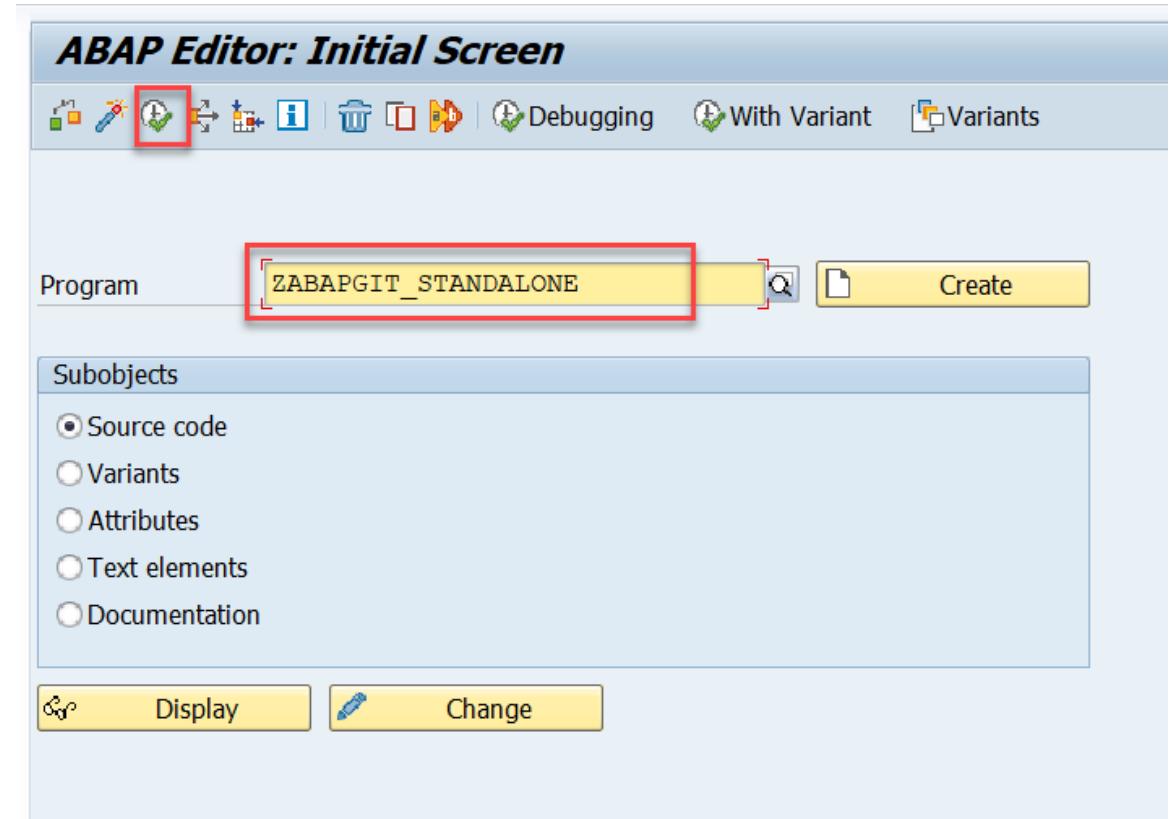
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

STEP 2: Download ABAP Code

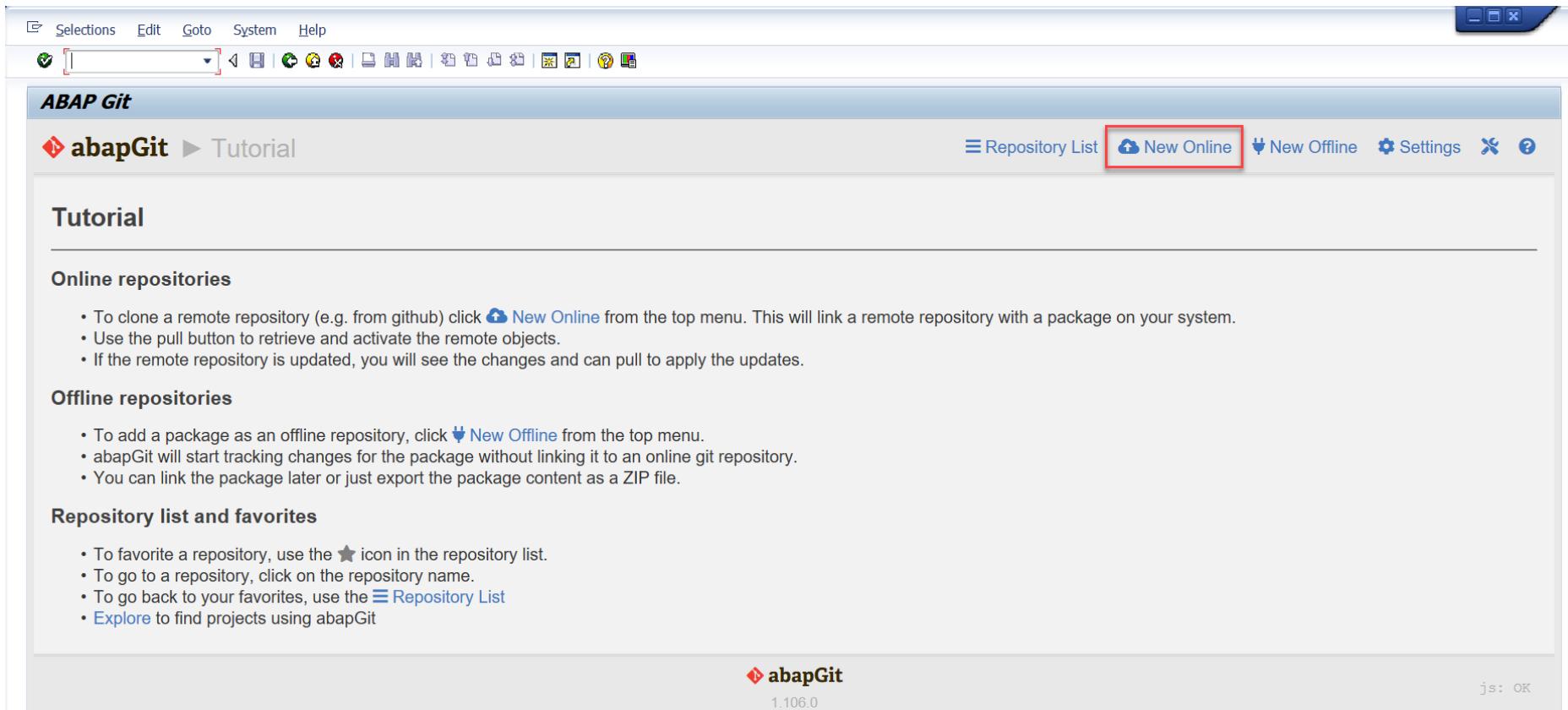
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 2: Download ABAP Code

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



STEP 2: Download ABAP Code

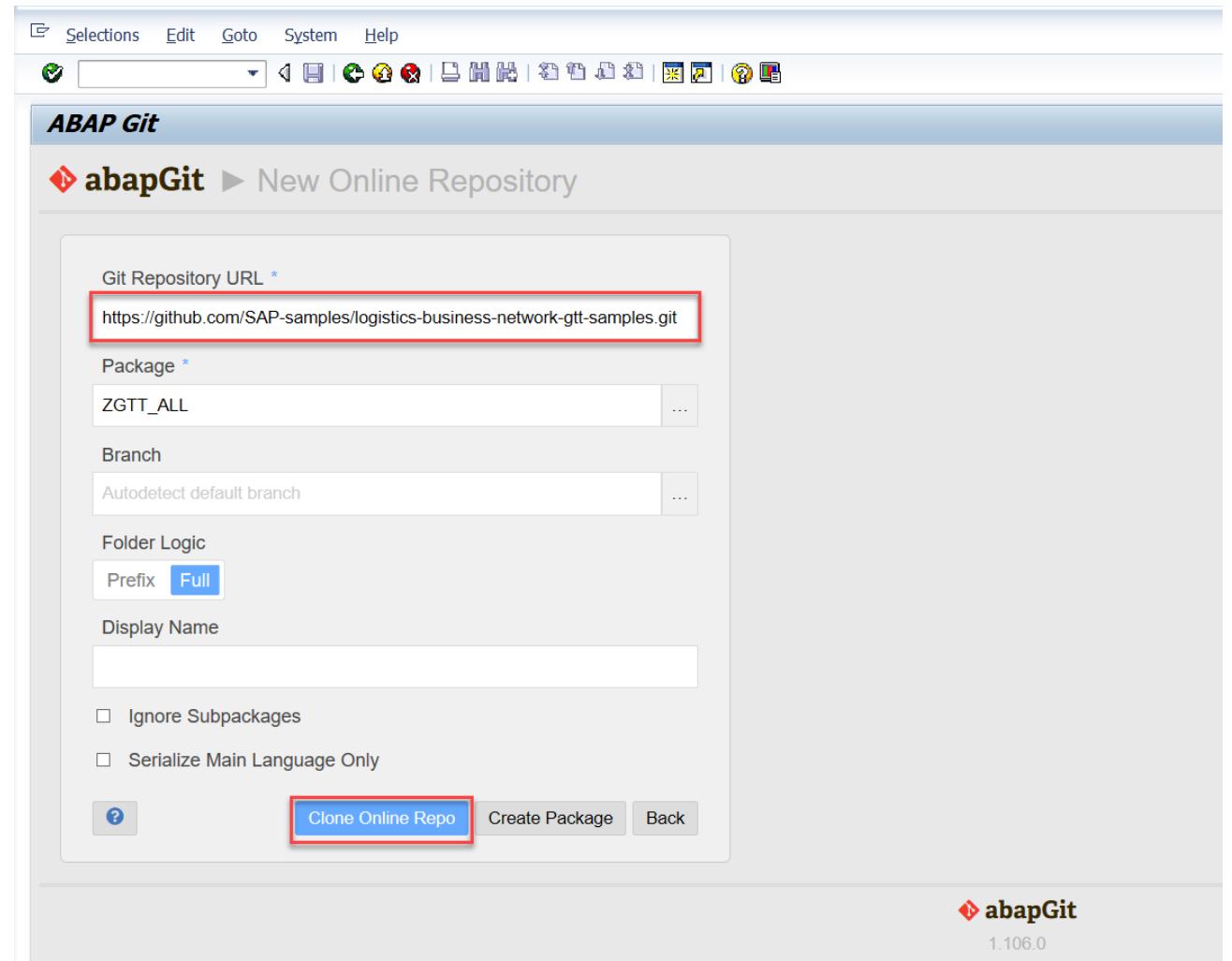
2-4: Fill in the **Git Repository URL**:

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

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

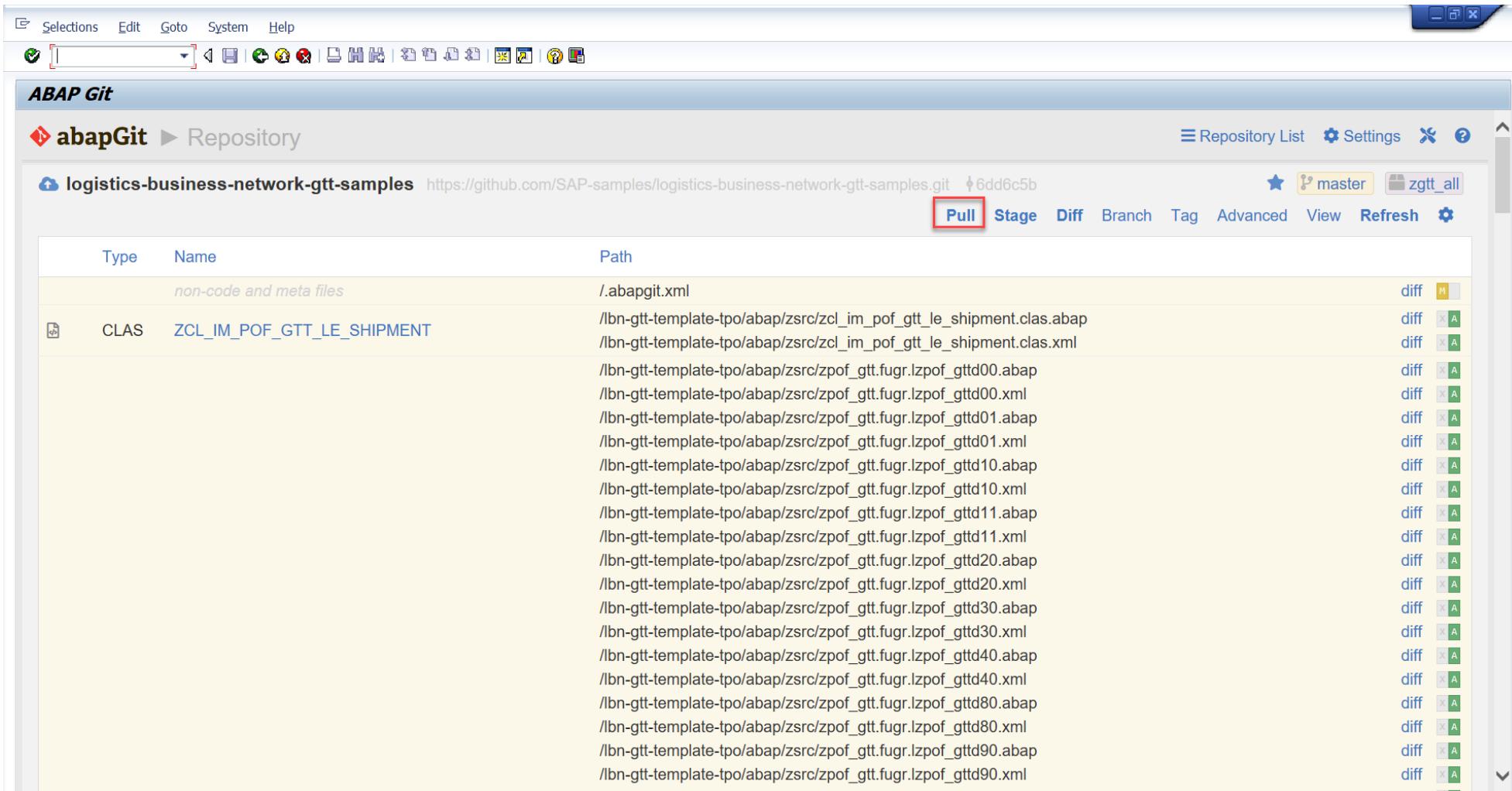
2-6: Set *Full* for **Folder Logic**

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



STEP 2: Download ABAP Code

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

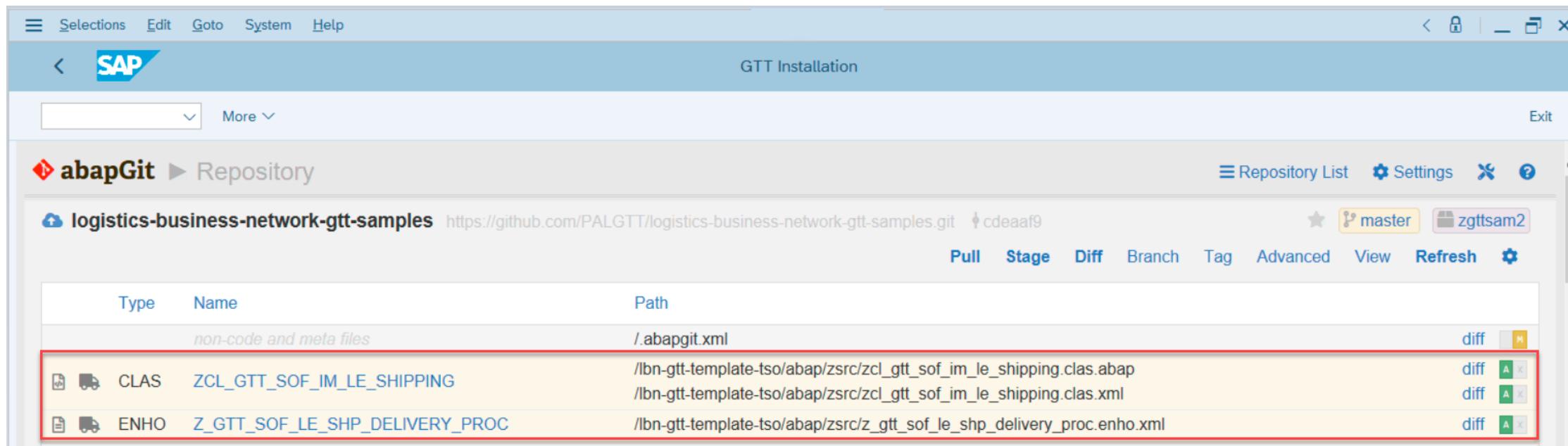


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the 'Repository' view for 'logistics-business-network-gtt-samples' at <https://github.com/SAP-samples/logistics-business-network-gtt-samples.git>. The repository is currently at commit `6dd6c5b`. The 'master' branch is selected. A red box highlights the 'Pull' button in the top navigation bar. The table below lists files and their paths, with 'diff' and 'A' icons indicating changes.

Type	Name	Path	diff	A
non-code and meta files				
		./abapgit.xml	diff	M
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd00.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd00.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd01.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd01.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd10.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd10.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd11.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd11.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd20.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd20.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd30.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd30.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd40.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd40.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd80.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd80.xml	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd90.abap	diff	A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gttd90.xml	diff	A

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

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



The screenshot shows the SAP GTT Installation interface with a GitHub repository named 'logistics-business-network-gtt-samples'. The repository URL is <https://github.com/PALGTT/logistics-business-network-gtt-samples.git>. The commit hash is cdeaaaf9. The interface includes tabs for Pull, Stage, Diff, Branch, Tag, Advanced, View, Refresh, and Settings. A red box highlights two specific rows in the table below, which represent deleted objects.

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

This is because the enhancement implementation Z_GTT_SOFTWARE_SHIPPING_DELIVERY_PROC is already obsolete and removed from the GitHub, the report ZABAPGIT_STANDALONE cannot remove the object which was already deleted in GitHub.

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

Solution:

Option 1:

1-1) Deactivate the BADI implementation.

Option 2:

2-1) Delete the enhancement implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC

2-2) Delete the BADI implementation class ZCL_GTT_SOF_IM_LE_SHIPPING

Notes:

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

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

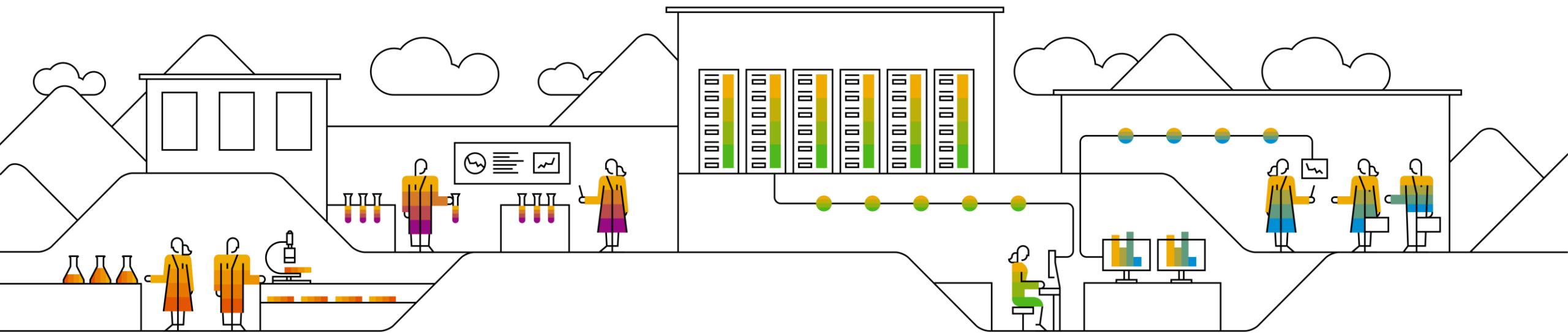
Known Issue: Remotely Deleted Object Cannot be Synchronized to the Local Object

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

The screenshot shows two SAP application windows. The top window is titled "BAdI Builder: Initial Screen for Implementations". It has tabs for "Check", "Delete implementation", "Copy implementation", "Rename implementation", "Application help", and "More". Below the tabs, there's a section for "Edit Implementation" with a radio button for "New BAdI" selected. The "Enhancement Implementation" field contains the value "Z_GTT_SOF_LE_SHP_DELIVERY_PROC", which is highlighted with a red box. The bottom window is titled "Enhancement Implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC Display". It also has tabs for "Properties", "History", "Technical Details", and "Implementation Elements", with "Implementation Elements" being the active tab. In the "Implementation Elements" tab, there's a table with one row. The row has columns for "BAdI Implementations" (containing "Z_GTT_SOF_IM_LE_SHIPPING") and "Description" (containing "Implementing Class"). To the right of the table, there are fields for "Badl Implementation" (set to "Z_GTT_SOF_IM_LE_SHIPPING"), "Description" (set to "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list"), and "Runtime Behavior". Under "Runtime Behavior", there's a checkbox for "Implementation is active" which is unchecked, and a note below it stating "The implementation will not be called".

D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

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

The image displays two side-by-side screenshots of the SAP GTT (Global Trace and Trace) interface, specifically the "Define Application Object Types" screen.

Screenshot 1 (Left): Define Application Object Types - Details

- Header:** Display View "Define Application Object Types": Details
- Toolbar:** Display > Change, Previous entry, Next entry, Other entry..., Drill down, More ▾
- Dialog Structure:** Define Used Business Process, Define Application Object Type (selected), Define Event Types
- Form Fields:**
 - Bus. Proc. Type: ESC_SORDER
 - Appl. Obj. Type: ZGTT_SO_INT_HD (highlighted with a red box)
 - Text: Sales Order Header
- Buttons:** General Data, Control Tables, Object Identification (selected), Global Track & Trace Relevance, Parameter Setup
- Method for determination of AOID:** AOID Method: Determine from Field
- Application Object ID Source:**
 - First Field to Build Appl. Obj. ID: Cntrl Tab. Type: 1 Main Object Table, AO ID Field: VBELN
 - Second Field to Build Appl. Obj. ID: Cntrl Tab. Type: (empty), AO ID Field: (empty)
- Determine AOID By Function:**

Screenshot 2 (Right): IDOC Integration

- Header:** sof Active, Sales Order Fulfillment, Namespace: com.bnrgtsamples.gtt.app.sof, Correlation Level: 5
- Toolbar:** Edit, Draft View ▾
- Tracked Process:** SalesOrder
- IDOC Integration:** Tracked Process Mapping, ERP Object Type: Others, Application Object Type: ZGTT_SO_INT_HD (highlighted with a red box)
- Tracked Process / Events (2):**

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

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

2: Maintain Tracking ID Type

In the AOT you maintained, make sure the name of Tracking ID Type is as same as the name defined in the corresponding process type of the model in Manage Models app in GTT Version 2.

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

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

SAP AOT Screenshot: The "Display View 'Define Application Object Types': Details" screen for a process named "ESC_SORDER". Under "Parameter Setup", the "Tr.ID Code Set" field is highlighted with a red box and contains the value "SALES_ORDER".

Manage Models App Screenshot: The "Model Details" screen for a model named "SOF". Under the "Tracked Process" tab, the "Items (6)" section lists several entities. The "SalesOrder" entity has its "Tracking Id Type" field highlighted with a red box and set to "SALES_ORDER". A modal dialog titled "Edit Tracked Process" is open for the "SalesOrder" entity, showing the "Name" field set to "SalesOrder" and the "Tracking Id Type" field set to "SALES_ORDER", both of which are also highlighted with red boxes.

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

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

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

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.

Function	Function Module
510_WRF_MM_ITEM_01	WRF_XRA_MM_ITEM_01
OBP10_DELIV	/SAPTRX/XRA_SD_DELIV_OBP10
OBP10_HU_IN_DLV	/SAPTRX/XRA_SD_HU_IN_DLV_OBP10
OCB10_CONTAINER	/SCTM/REL_CREATION_CONTAINER
OCB10_ORDER	/SCTM/REL_CREATION_BOOKING
ODT20_REL_FU	/SCMTMS/REL_AOT_FU
ODT20_REL_TOUR	/SCMTMS/REL_AOT_ACT_TOR
ODT30_REL_INS	/SCMTMS/REL_AOT_INS
ODT30_REL_TU	/SCMTMS/REL_AOT_TU
PCM10_ITEM	/SAPTRX/XRA_MM_ITEM_PCM10
PMF10_ORDER	/SAPTRX/XRA_PP_ORDER_PMF10
RES30_REL_RESOURCE	/SCMTMS/REL_AOT_RESOURCE
RES30_REL_TU	/SCMTMS/REL_AOT_RES_TU
RES30_REL_VEH	/SCMTMS/REL_AOT_RES_VEH
SNC10_MSGIN	/SCA/EM_MSG_RELEVANCE_CHECK
SNC10_PURORD	/SCA/EM_PO_RELEVANCE_CHECK
SNC10_RPLORD	/SCA/EM_RPL_RELEVANCE_CHECK
ZE2E_OBP10_DELIV	ZE2E_XRA_SD_DELIV_OBP10
ZGTT_FERRERO_DEHDR	ZGTT_FERRERO_OTE_DE_HDR_REL
ZGTT_FERRERO_SHPHDR	ZGTT_FERRERO_OTE_SHP_HDR_REL

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:

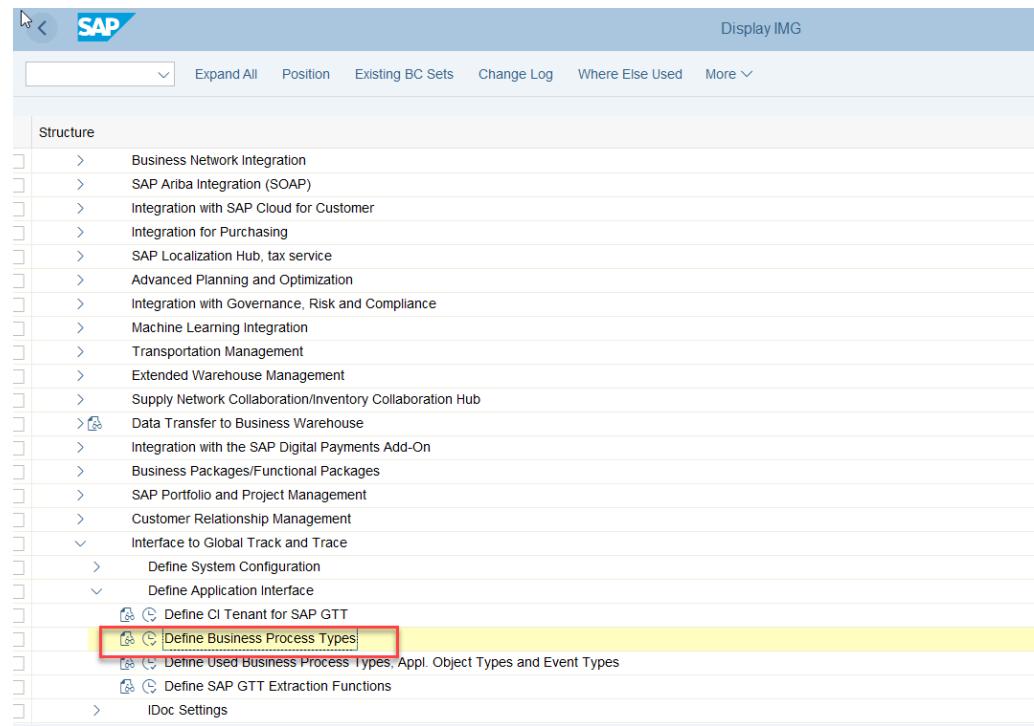
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_HD	Function for setup of control parameters of delivery header
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_ITEM	Function for setup of control parameters of delivery item
Control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

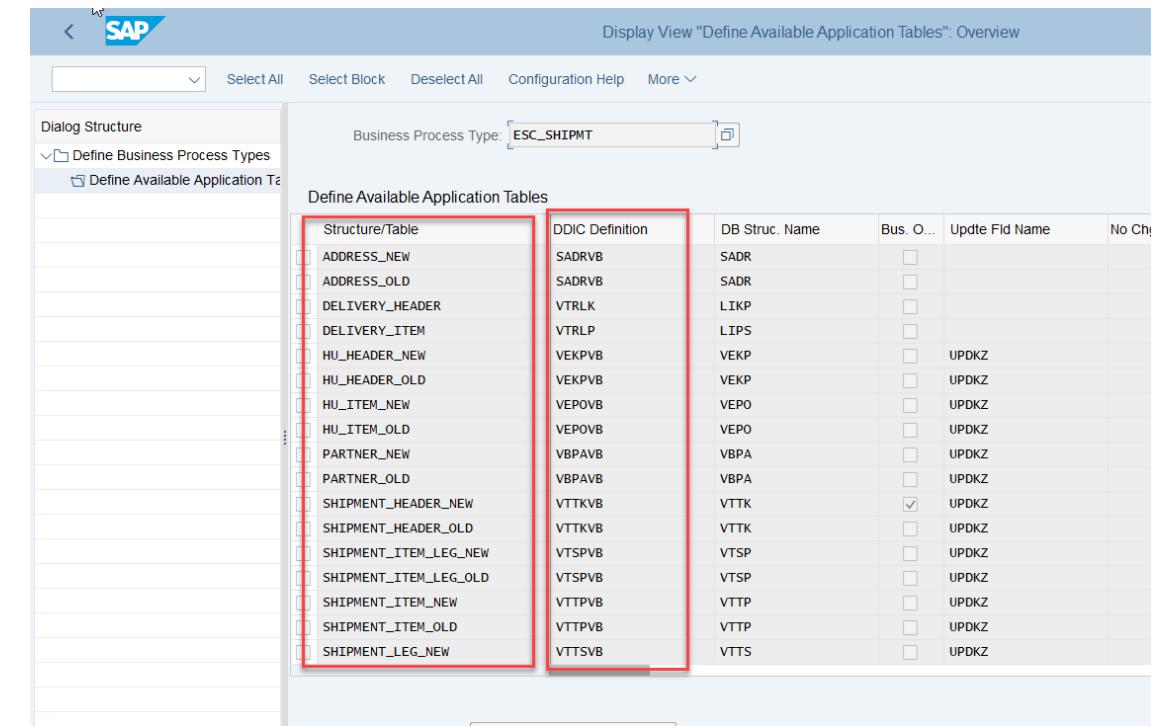
5-1: In **Display IMG** page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

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 navigation path is: Structure > Integration with Other SAP Components > Interface to Global Track and Trace > Define Application Interface. At the bottom of the list, there are two items highlighted with red boxes: "Define Business Process Types" and "Define Used Business Process Types, Appl. Object Types and Event Types". The "Define Business Process Types" item is the one currently selected, indicated by a yellow background.



The screenshot shows the SAP Display View "Define Available Application Tables" for the business process type "ESC_SHIPMT". The table lists various context tables and their corresponding DDIC definitions. The columns are: Structure/Table, DDIC Definition, DB Struc. Name, Bus. O..., Updt Fld Name, and No Chg. A red box highlights the first 18 rows of the table, which correspond to the context tables listed in the previous screenshot.

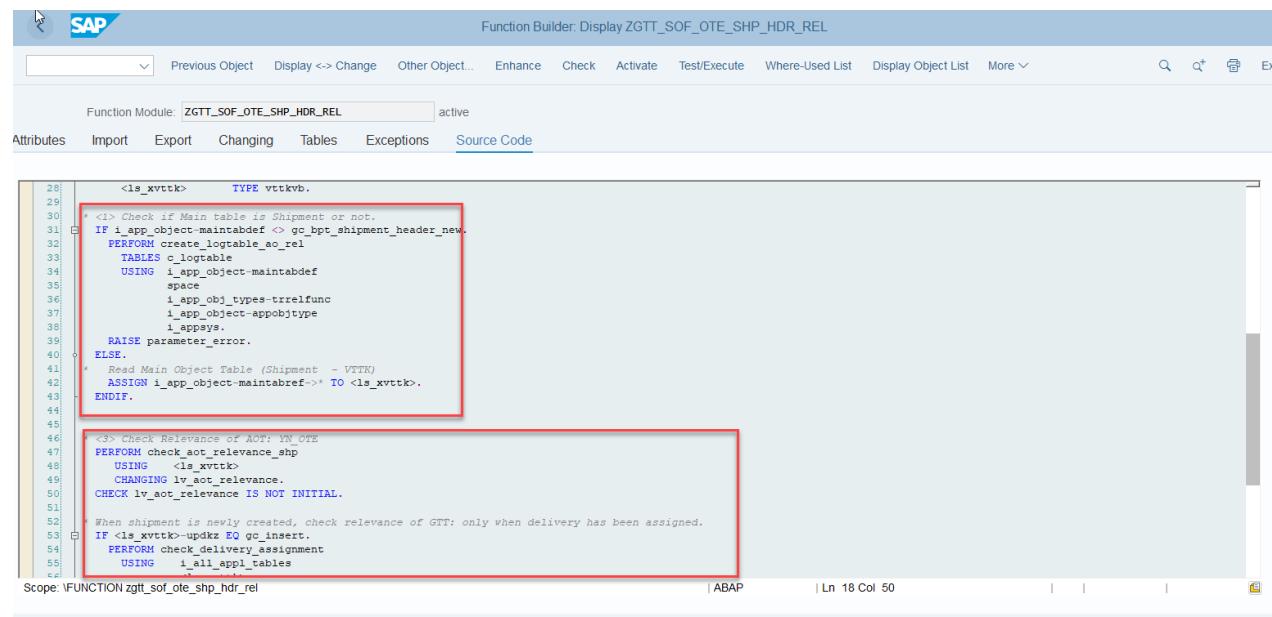
Structure/Table	DDIC Definition	DB Struc. Name	Bus. O...	Updt Fld Name	No Chg
ADDRESS_NEW	SADRVB	SADR			
ADDRESS_OLD	SADRVB	SADR			
DELIVERY_HEADER	VTRLK	LIKP			
DELIVERY_ITEM	VTRLP	LIPS			
HU_HEADER_NEW	VEKPVB	VEKP			UPDKZ
HU_HEADER_OLD	VEKPVB	VEKP			UPDKZ
HU_ITEM_NEW	VEPOVB	VEPO			UPDKZ
HU_ITEM_OLD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

6: Coding Tips in the GTT relevance function modules

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

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

See sample code of function: *ZGTT_SOF_OTE_SHP_HDR_REL*



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOF_OTE_SHP_HDR_REL". The "Source Code" tab is selected. The code is written in ABAP and performs the following steps:

- It checks if the main table is a Shipment (VTTK) using the condition `IF i_app_object-maintabdef <> go_bpt_shipment_header_new.`
- If it is a Shipment, it performs a log table creation with the table `c_logtable` and space `space`, using object types `trelfunc` and `appobjtype`. It also handles `l_appls` and `l_appsys`.
- If it is not a Shipment, it reads the Main Object Table (Shipment - VTTK) and assigns the object reference to `<ls_xvttk>`.
- It then checks the relevance of ACT1 IN OTE using the `check_act_relevance_shp` program unit.
- Finally, it updates the delivery assignment table `gc_insert` and performs a delivery assignment check.

The code is annotated with several red boxes highlighting specific sections of logic:

- A red box surrounds the logic for checking if the main table is a Shipment and performing the log table creation.
- A red box surrounds the logic for reading the Main Object Table (Shipment - VTTK) and assigning the object reference.
- A red box surrounds the logic for checking the relevance of ACT1 IN OTE.
- A red box surrounds the final update and delivery assignment check logic.

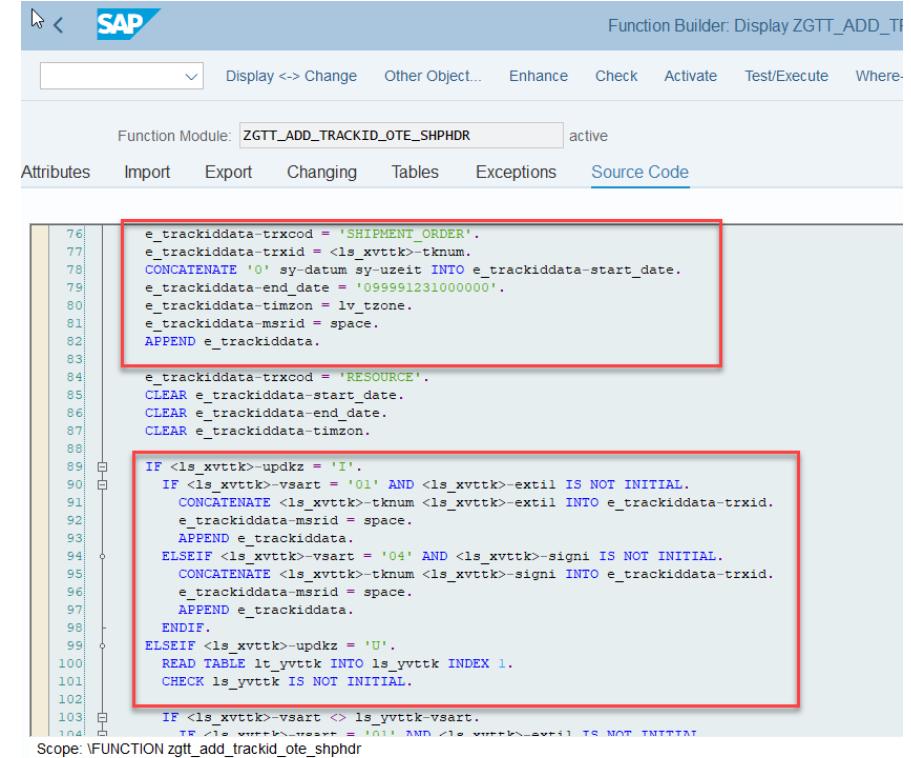
7: Coding Tips in the Tracking ID function modules

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

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

See sample code of function:

ZGTT_ADD_TRACKID_OTE_SHPHDR



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_ADD_TRACKID_OTE_SHPHDR". The "Source Code" tab is selected. The code is written in ABAP and handles the creation of tracking IDs based on different update types (I, O1, O4, U) and specific conditions. The code uses concatenation and APPEND statements to build the tracking ID structure. A red box highlights the main processing loop for update types O1, O4, and U, where it checks if the tracking ID exists and then appends it to the tracking data table.

```
76 e_trackiddata-trxcod = 'SHIPMENT_ORDER'.
77 e_trackiddata-trxid = <ls_xvttk>-tknum.
78 CONCATENATE '01' sy-datum sy-uzeit INTO e_trackiddata-start_date.
79 e_trackiddata-end_date = '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>-updkz = 'I'.
90   IF <ls_xvttk>-vsart = '01' AND <ls_xvttk>-extil IS NOT INITIAL.
91     CONCATENATE <ls_xvttk>-tknum <ls_xvttk>-extil INTO e_trackiddata-trxid.
92     e_trackiddata-msrid = space.
93     APPEND e_trackiddata.
94   ELSEIF <ls_xvttk>-vsart = '04' AND <ls_xvttk>-signi IS NOT INITIAL.
95     CONCATENATE <ls_xvttk>-tknum <ls_xvttk>-signi INTO e_trackiddata-trxid.
96     e_trackiddata-msrid = space.
97     APPEND e_trackiddata.
98   ENDIF.
99 ELSEIF <ls_xvttk>-updkz = 'U'.
100   READ TABLE lt_yvttk INTO ls_yvttk INDEX 1.
101   CHECK ls_yvttk IS NOT INITIAL.
102
103 IF <ls_xvttk>-vsart <> ls_yvttk-vsart.
104   IF <ls_xvttk>-vsart = '01' AND <ls_xvttk>-extil IS NOT INITIAL
105
Scope: FUNCTION zgtt_add_trackid_ote_shphdr
```

8: Coding Tips in the Control Parameter function modules

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

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

See sample code of function: *ZGTT_SOF_OTE_SHP_HD*

8: Coding Tips in the Control Parameter Function Modules

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

The screenshot shows the SAP Manage Models app interface. At the top, there's a header with the SAP logo, a 'Model Details' dropdown set to 'Internal - Test', and user icons for help and profile. Below the header, the model name 'sof' is shown with a status of 'Active'. A 'Sales Order Fulfillment' description is present. On the right, there are 'Edit' and 'Draft View' buttons.

The main navigation bar includes tabs for 'Tracked Process', 'Field Type Pool', 'Event Type Pool', 'Code List', 'IDOC Integration' (which is currently selected), 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below the navigation bar, a 'Tracked Process' dropdown is set to 'Shipment' and an 'Integration Switch' is turned 'ON'.

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

The 'Tracked Process / Events (26)' table lists various tracked processes and their corresponding IDOC segments and event codes. The table includes sections for 'Tracked Process', 'Event Types', and specific events like 'LoadingStart', 'POD', 'Departure', 'Arrival', and 'LoadingEnd'.

A red box highlights the 'User Model Fields' table, which maps application object fields to IDOC segments and fields. The table has columns for 'Field', 'IDOC Segment', and 'IDOC Field'. It includes entries for 'shipmentNo', 'serviceAgentLbId', 'transportationMode', 'dangerousGoods', 'forwardingAgentTrackingId', and two expanded sections: 'stops' and 'resourceTPs'.

User Model Fields		
Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCP	YN_SHP_NO
serviceAgentLbId	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 name "ZGTT_SOF_OTE_SHP_HD" is selected in the top bar. The "Source Code" tab is active. The code is as follows:

```
120 * Shipment number: VTTR-TKNUM
121 ls_control_data-paramname = gc_cp_yn_shp_no.
122 ls_control_data-value     = <ls_xvttk>-tknum.
123 APPEND ls_control_data TO e_control_data.
124
125 * Service Agent ERP ID
126 ls_control_data-paramname = gc_cp_yn_shp_sa_erp_id.
127 ls_control_data-value     = <ls_xvttk>-tdlnr.
128 APPEND ls_control_data TO e_control_data.
129
130 * Service Agent LBN ID
131 CLEAR lv_forward_agt.
132 CALL METHOD cl_site_bp_assignment=>select_bp_via_cvi_link
    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:

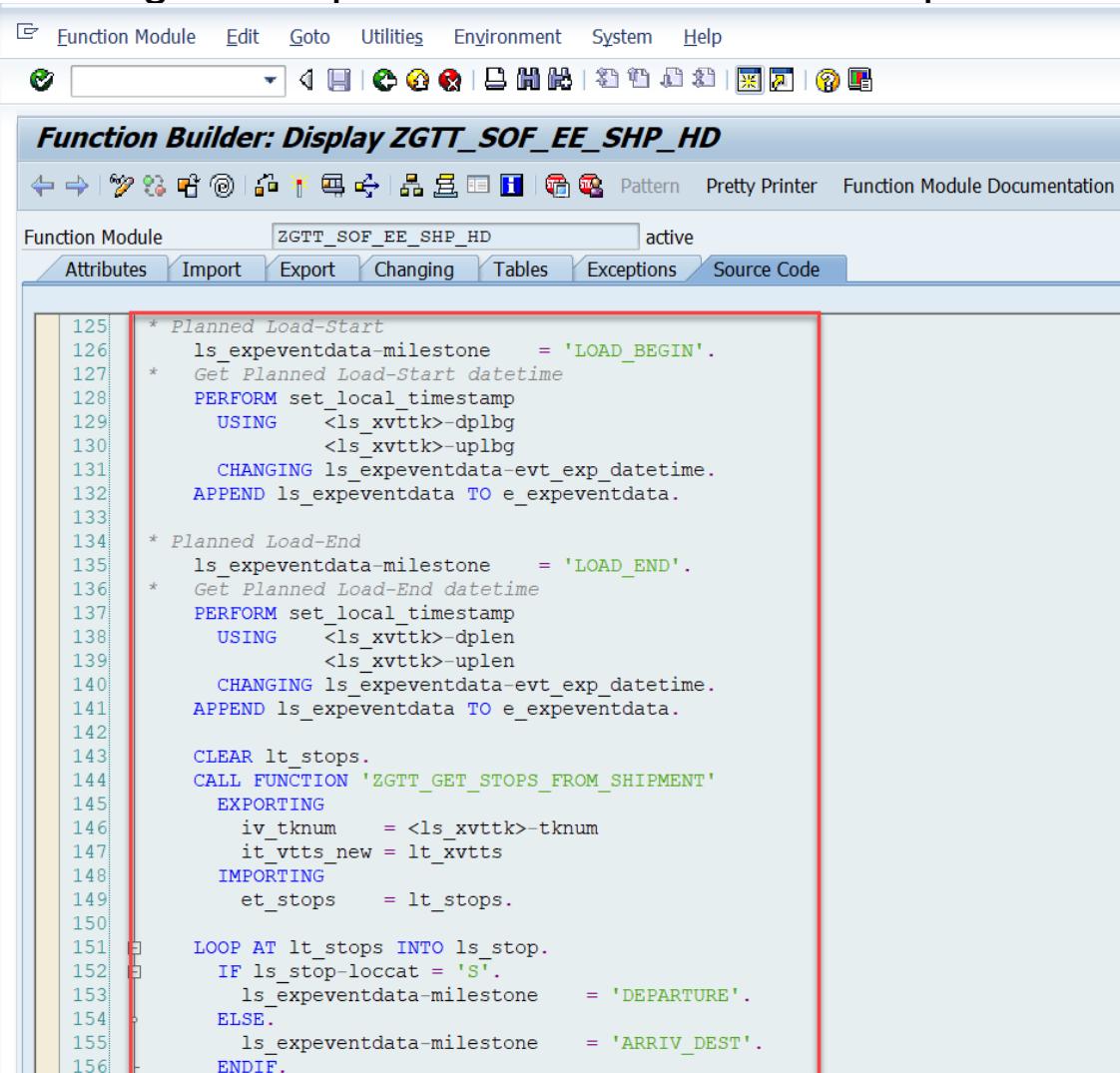
1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. As default except no change made on the model configuration, GTT version 2 asks for full transport for all the planned events, which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

See sample code of function: *ZGTT_SOF_EE_SHP_HD*

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

9: Coding Tips in the Planned Event Function Modules

Main logic of shipment Planned Events is implemented in function module ZGTT_SOF_EE_SHP_HD



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOF_EE_SHP_HD". The function module "ZGTT_SOF_EE_SHP_HD" is active. The "Source Code" tab is selected. The code is written in ABAP and handles the logic for planned events related to load starts and ends, and stops.

```
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 GTT via TP request, which leads to an error)
5. If the event has reference table information, fill in the table *CT_TRACKREFERENCES*.
6. The field *CT_TRACKINGHEADER-SRCCOD*, *SRCID*, *SRCTX* is used for event reason transport.
7. In *Manage Models* app, click tab *IDOC Integration* to map the user-defined parameter names and model field names.

See sample code of function: *ZGTT_SOF_EE_DE_PICKING*

10: Coding Tips in the Event Data Function Modules

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

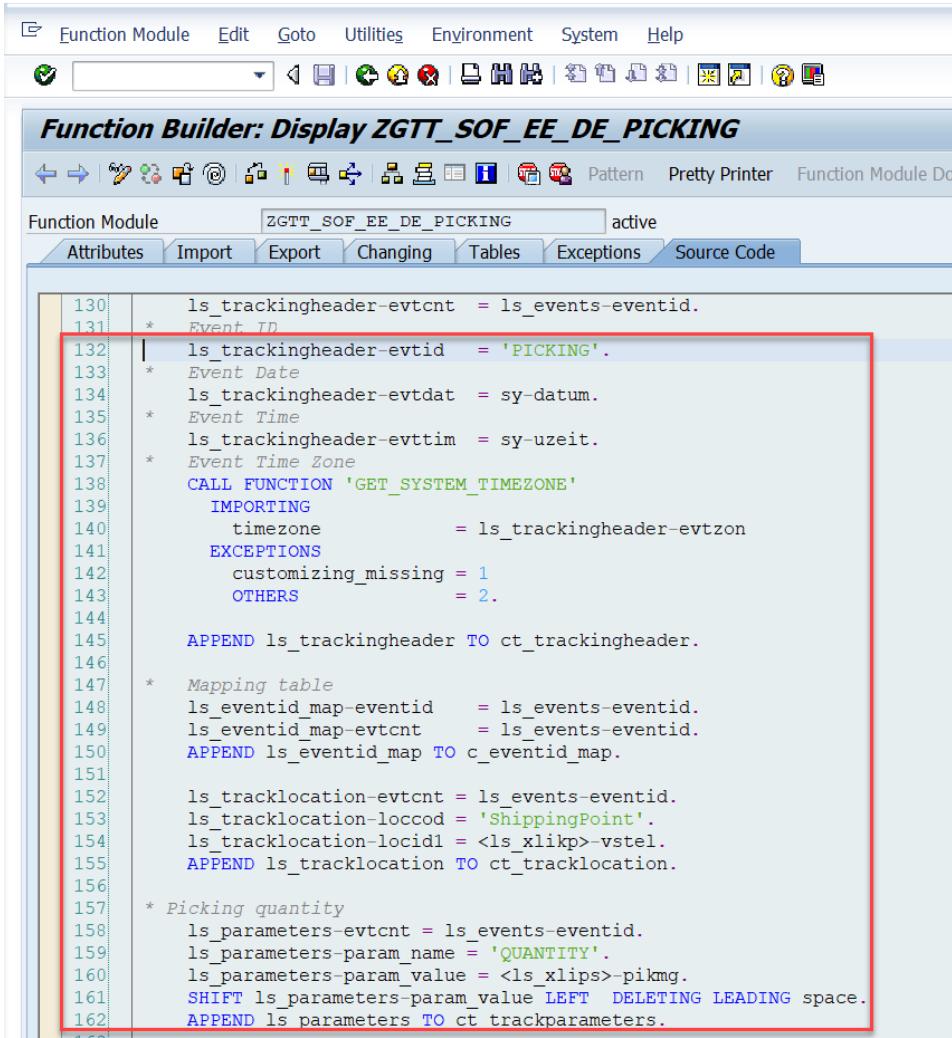
The screenshot shows the SAP Model Details interface for the 'Internal - Test' model. The 'sof' model is active. The 'IDOC Integration' tab is selected. The 'Tracked Process' dropdown is set to 'DeliveryItem'. The 'Integration Switch' is turned 'ON'. The 'Tracked Process Mapping' section shows 'ERP Object Type: Others' mapped to 'Application Object Type: ZGTT_DE_INT_ITEM'. The 'Tracked Process / Events (4)' table lists four events: DeliveryItemEvent (IDOC E1EHPAO), Picking (IDOC E1EVMPAR, Event Code PICKING), Packing (IDOC E1EVMPAR, Event Code PACKING), and DeliveryItemPOD (IDOC E1EVMPAR, Event Code DLV POD). The 'User Model Fields' table maps the 'quantity' field from the 'quantity' segment (E1EVMPAR) to the IDOC field 'QUANTITY'.

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

Field	IDOC Segment	IDOC Field
quantity	E1EVMPAR	QUANTITY

10: Coding Tips in the Event Data Function Modules

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



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOF_EE_DE_PICKING". The function module "ZGTT_SOF_EE_DE_PICKING" is active. The code editor displays the following ABAP code:

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

11: Enhancement codes for cross-processes tracking

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

- When the shipment process is updated and transported to GTT, the preceding delivery and item process, and their planned events needs to be updated and transported to GTT.

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

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

SAP Business Add-In Builder: Display Implementation Z_GTT_SOF_LE_SHIPMNT

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

Properties Interface

Interface Name: IF_EX_BADI_LE_SHIPMENT

Name of Implementing Class: ZCL_IM_GTT_SOF_LE_SHIPMNT

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

Default Implementation Class: []

11: Enhancement codes for cross-processes tracking

The cross processes tracking scenarios cover below:

Shipment -> Delivery and Delivery Item:

1\ Tracking ID (Delta Transport)

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

2\ Shipment Composition (Full Transport)

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

3\ Planned Event in Delivery (Full Transport)

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

4\ Planned Event in Delivery Item (Full Transport)

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

12: Known Issues

1. Planned Event Extension not enabled

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

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

2. IDOC sequencing issue

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

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

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

13: Solution of IDOC Sequencing Issue

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

2. Create CI tenant.

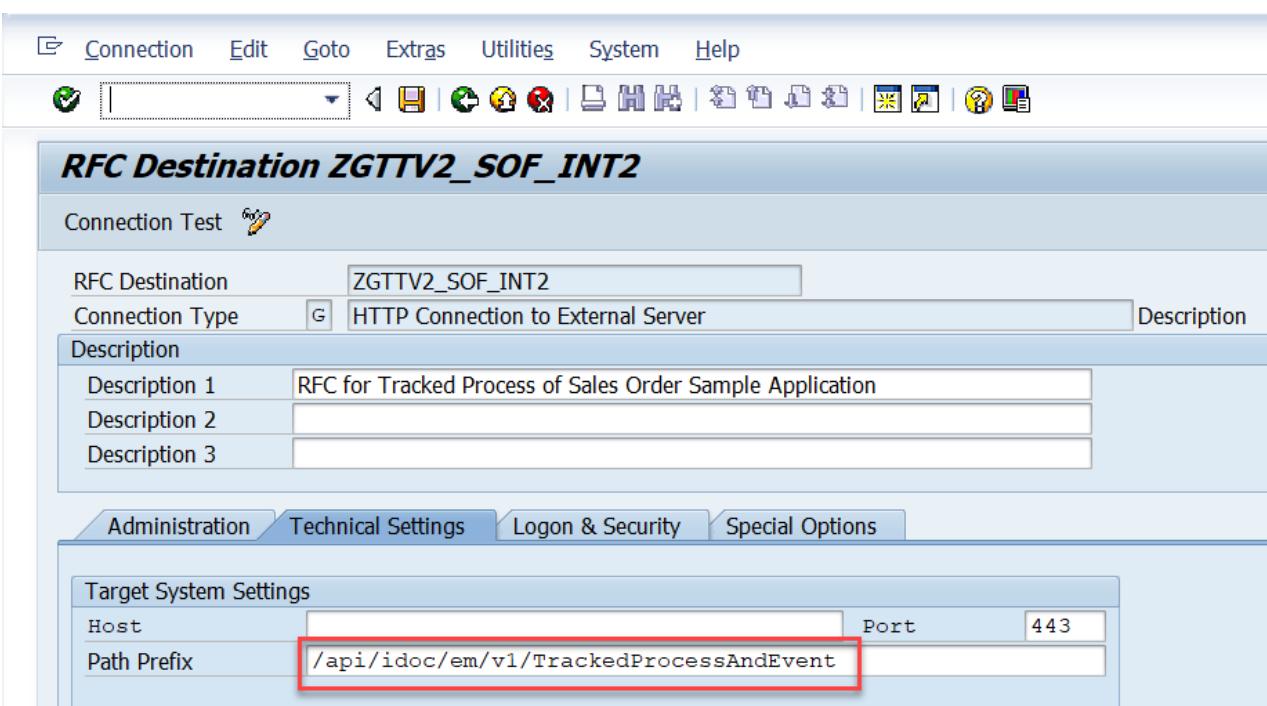
Select “**GTT2.0 Logistics Business Network - Track and Trace**” for SAP Track & Trace Version

SAP Global Track & Trace Definitions				
CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description	
ZGTTSOFIN2	ZLSGTTINT	GTT2.0 Logistics Business N...	CI For GTT V2 Integration system Sales Order Sample APP	

3. Create RFC destination

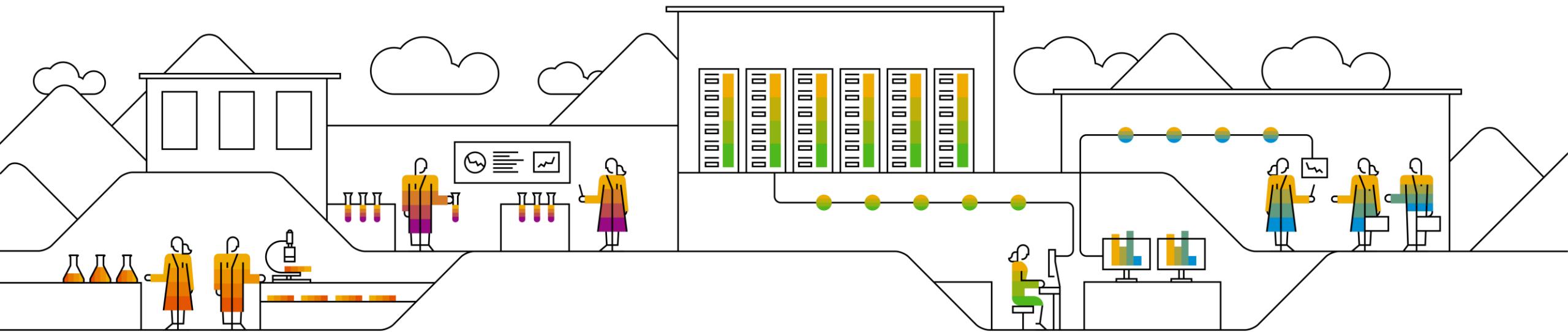
You need to configure only one RFC connection for both event and tracked process.

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



The screenshot shows two SAP GUI windows. The top window is titled "Change View 'SAP Global Track & Trace Definitions': Overview of Select". It displays a table with one row: "ZGTTSOFIN2" in the CI for Global Track & Trace column, "ZLSGTTINT" in the CI Log. System column, "GTT2.0 Logistics Business N..." in the SAP Track & Trace Version column, and "CI For GTT V2 Integration system Sales Order Sample APP" in the Description column. The bottom window is titled "RFC Destination ZGTTV2_SOFTWARE_INT2". It has tabs for Administration, Technical Settings, Logon & Security, and Special Options. The "Administration" tab is selected. Under "Target System Settings", there are fields for Host and Port (set to 443). The "Path Prefix" field contains the value "/api/idoc/em/v1/TrackedProcessAndEvent", which is highlighted with a red box.

Thanks



Follow us



www.sap.com/contactsap

© 2021 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. 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, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See www.sap.com/copyright for additional trademark information and notices.

THE BEST RUN  SAP[®]