



SAP Logistics Business Network, Global Track and Trace Option **Track Shipments - 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>

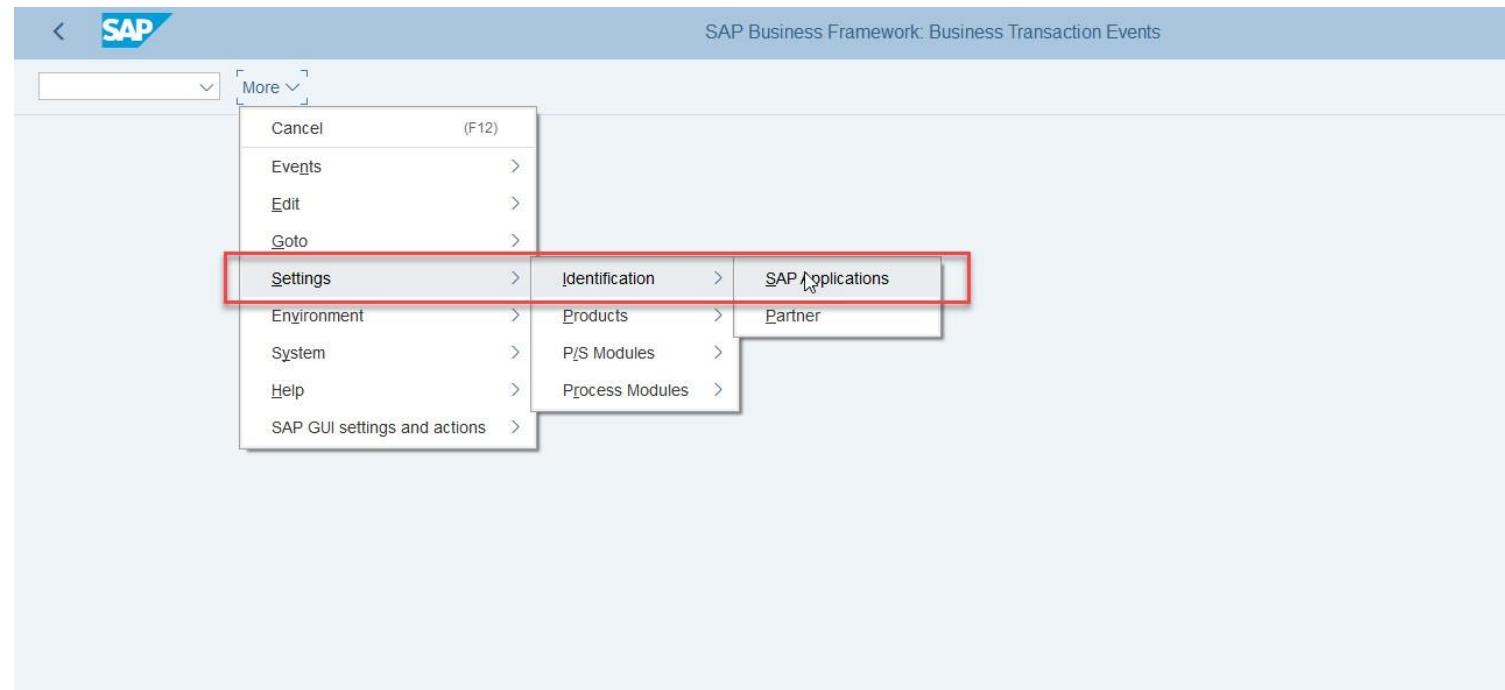
SAPNotes								
11 SAP Note(s) found								
SAP Component	Number	Versi...	Score	Title	Changed On	Status	Responsible	Category
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process		Program error
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer		Advance development
SCM-EM-AS	2834393	1	1	Solving ATC Issues	27.09.2019	Released for Customer		Program error
SCM-EM-AS	2819787	1	1	TM-EM integration - analyzing errors	25.07.2019	In Process		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		Program error
SCM-EM-AS	2609449	4	1	Delete orphaned entries in table /SAPTRX/AOTREF (2)	11.07.2019	Pilot Release		Workaround of missing
SCM-EM-AS	2502086	2	1	Aligning the BAPI processing mode with the communication mode	11.07.2017	Pilot Release		Special development
SCM-EM-AS	2339984	2	1	Orphaned EM inbound queues in application systems	18.04.2019	Released for Customer		Consulting
SCM-EM-AS	2159436	1	1	Runtme-Error "ABAP Programming" when trying to save delivery. System QSC-800	22.04.2015	In Process		Program error
SCM-EM-AS	1507998	4	1	Expert Consulting in the area of SAP Event Management	09.05.2011	Released for Customer		Consulting
IS-R-PUR-PCC	896191	3	1	FAQ: EM seasonal procurement (Consulting, Tips, Customizing)	13.07.2006	Released for Customer		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**

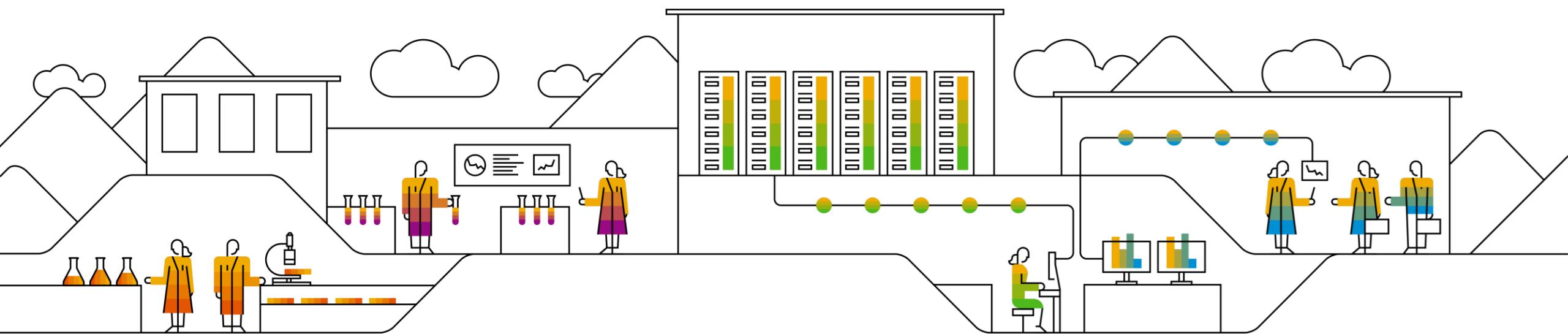
The screenshot shows a SAP Fiori application titled "Change View 'BTE Application Indicator': Overview". The interface includes a header with a SAP logo, navigation buttons (New Entries, Copy As..., Delete, Undo Change, Select All, Select Block, Deselect All, More), and action buttons (Display, Exit). The main area is a table with two columns: "Appl." and "Text". The "Appl." column lists various application codes, and the "Text" column provides a description. The row for "PI-EM" is highlighted with a red border, and its checkbox under "Text" is checked. Other rows 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. At the bottom of the table, there are buttons for "Position..." and "Entry 133 of 174". The footer contains "Save" and "Cancel" buttons.

Appl.	A	Text
PI-EM	<input checked="" type="checkbox"/>	SAP Event Manager Integration
PM	<input checked="" type="checkbox"/>	Instandhaltung
PM-BW	<input checked="" type="checkbox"/>	Instandhaltung-BW
PM-EQM	<input checked="" type="checkbox"/>	Instandhaltung, Equipment
PM-PAM	<input checked="" type="checkbox"/>	Instandhalt. Pool Assess 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

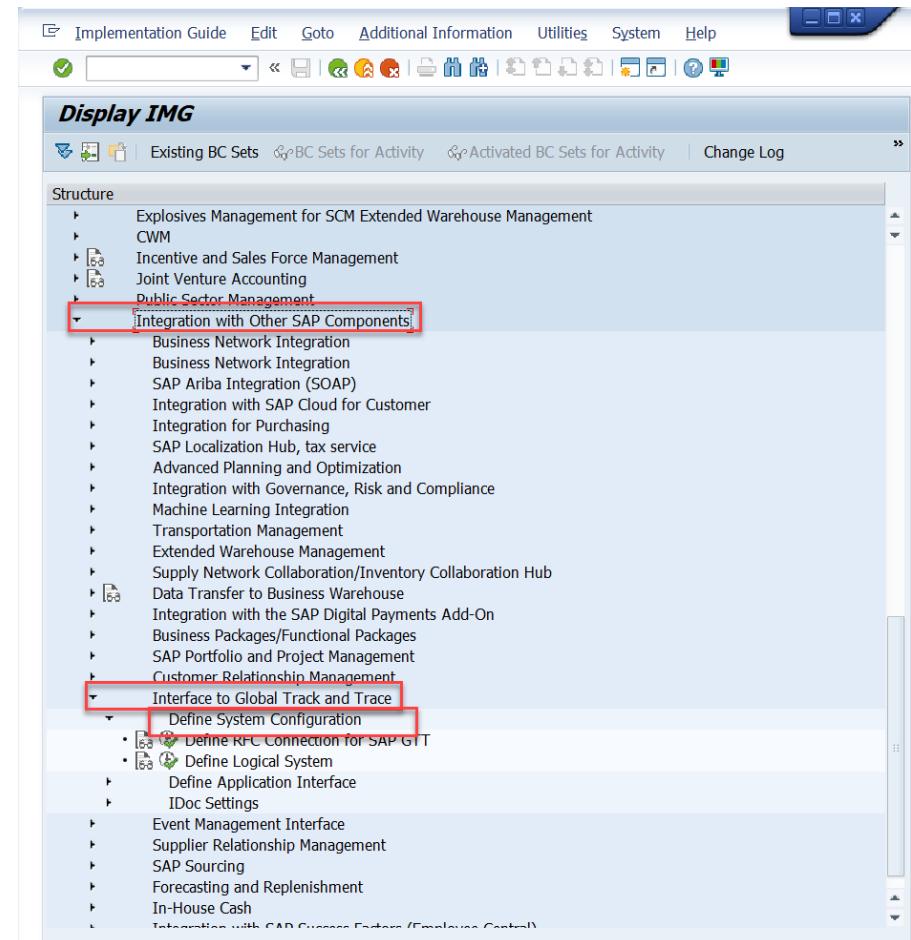
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

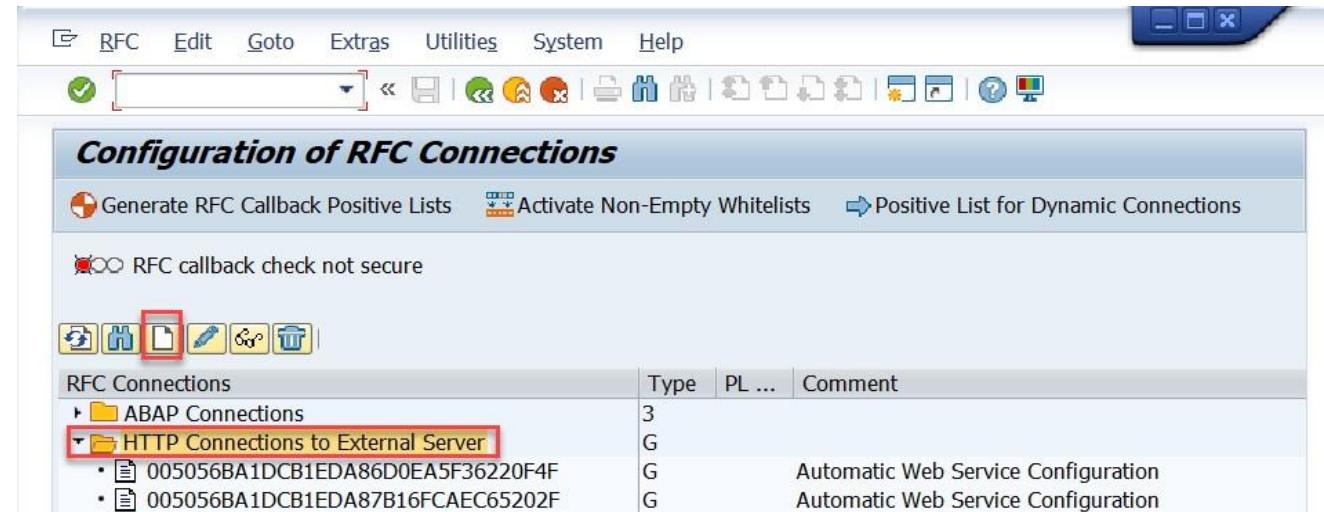
Define RFC Connection for SAP GTT



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 below:

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

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

Port: `443`

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

For the event:

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

For the tracked Process:

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

RFC Destination ZGTT_SST_FO_EVENT_ACC

Connection Test

RFC Destination	ZGTT_SST_FO_EVENT_ACC
Connection Type	G HTTP Connection to External Server
Description	
Description 1	RFC for Events of SST Sample Application to Acceptance
Description 2	
Description 3	

Administration Technical Settings Logon & Security Special Options

Target System Settings

Host	[REDACTED]	Port	443
Path Prefix	/api/idoc/em/v1/Event		

HTTP Proxy Options

Global Configuration	
Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

RFC Destination	RFC Destination Description	Host	Path Prefix	Port
ZGTT_SST_FO_EVENT_ACC	RFC for Event of Tracking Shipments Sample Application	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/Event	443
ZGTT_SST_FO_TP_ACC	RFC for Tracked Process of Tracking Shipments Sample Application	xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/TrackedProcess	443

STEP 1: Define RFC Connection for 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.

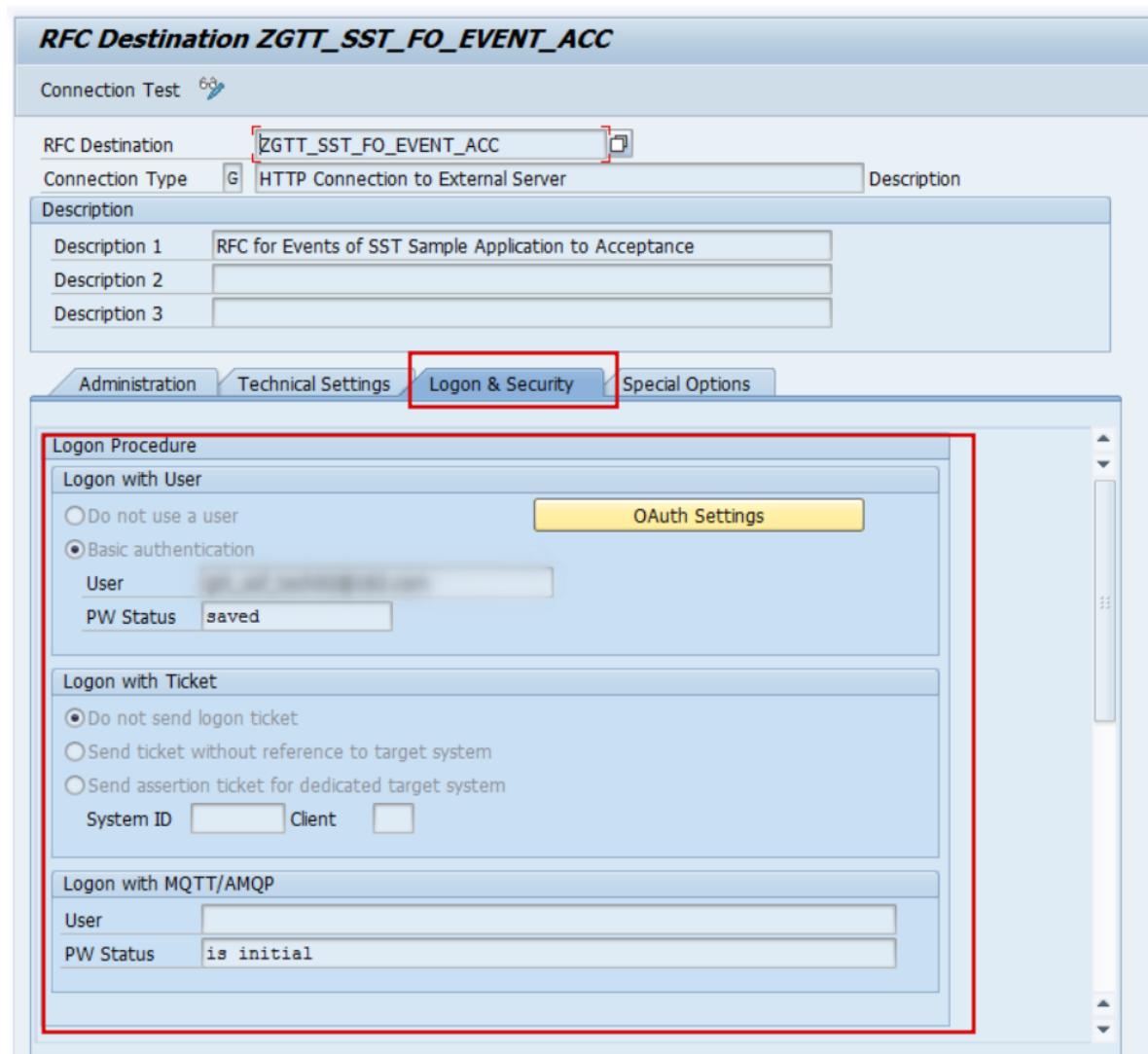
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

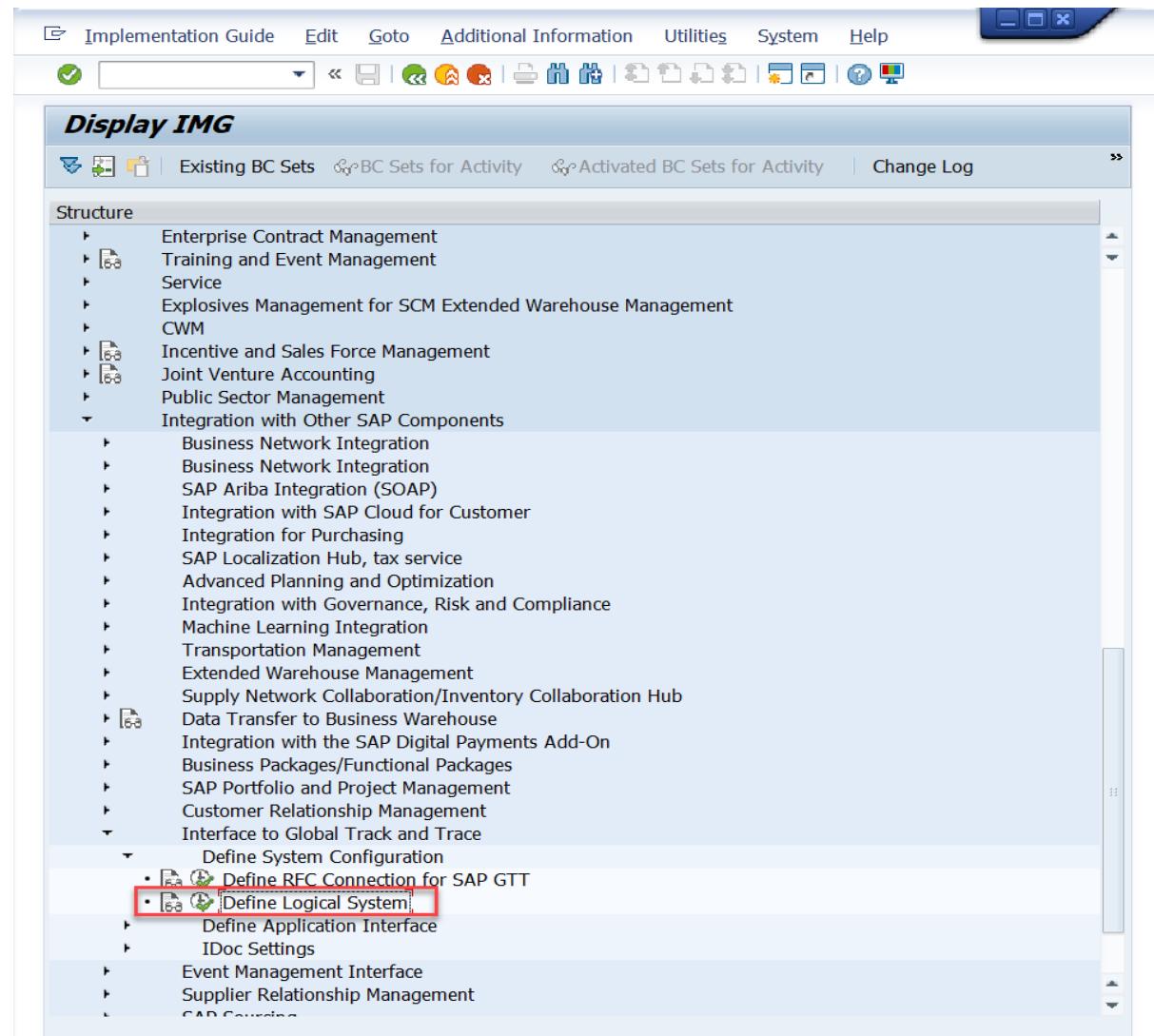
- one for event and
- the other for tracked process.



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

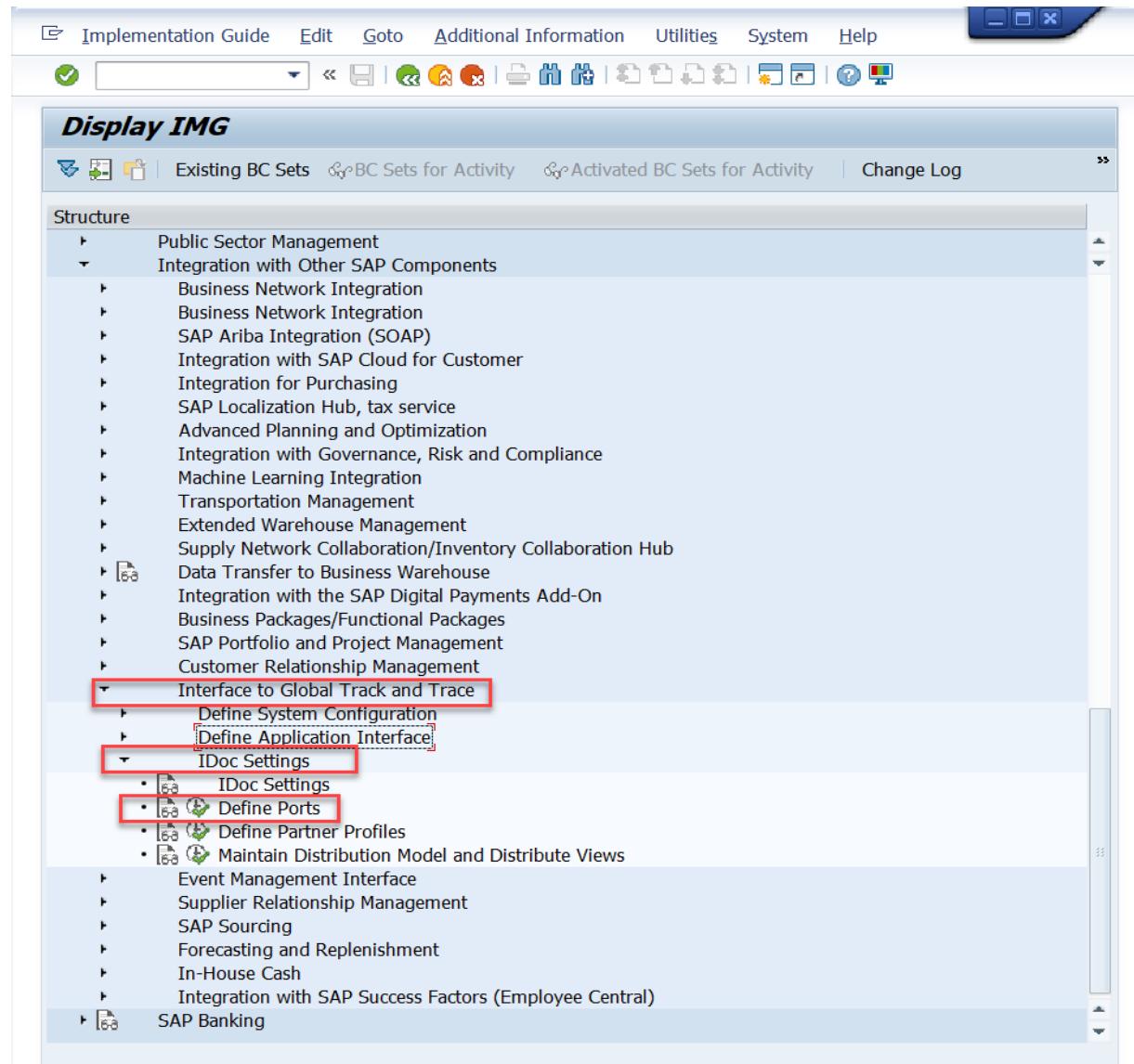
2-4: Save the configuration

Logical Systems	
Log.System	Name
ZGTTSSSTAC	Logical System For GTT SST - Acceptance

STEP 3: Define Ports

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

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



STEP 3: Define Ports

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

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

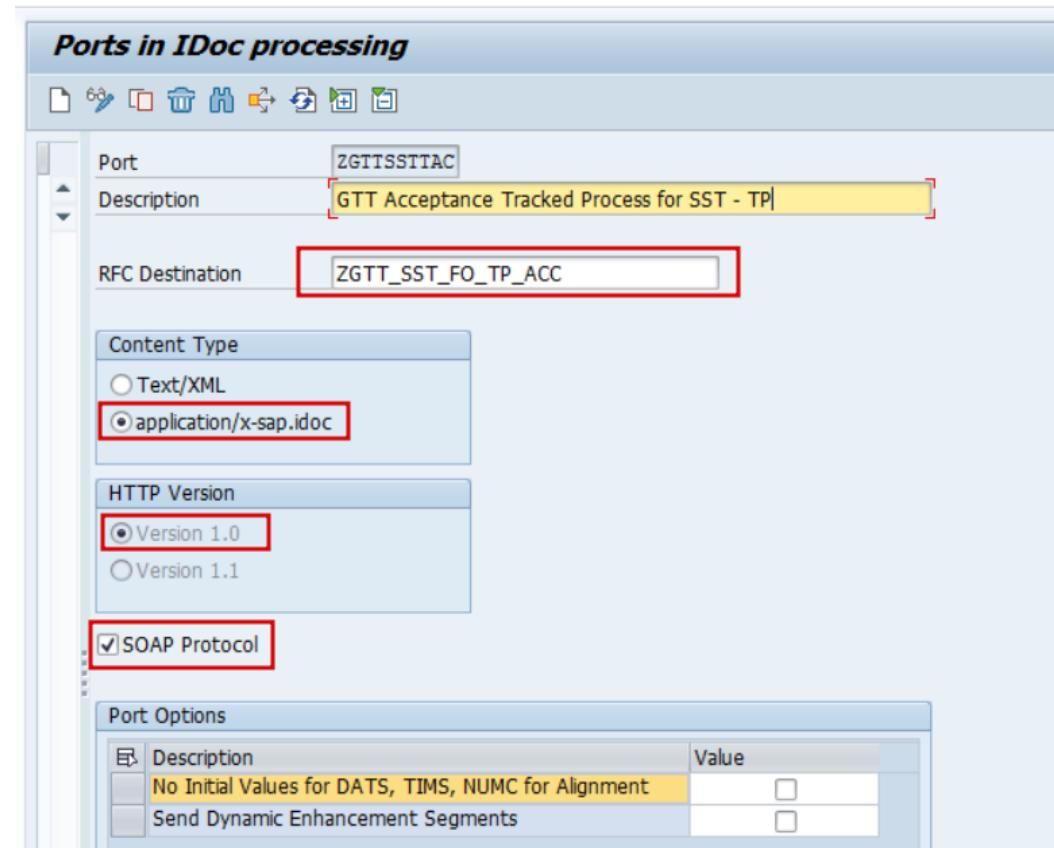
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol 8

3-8: Save the configuration

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

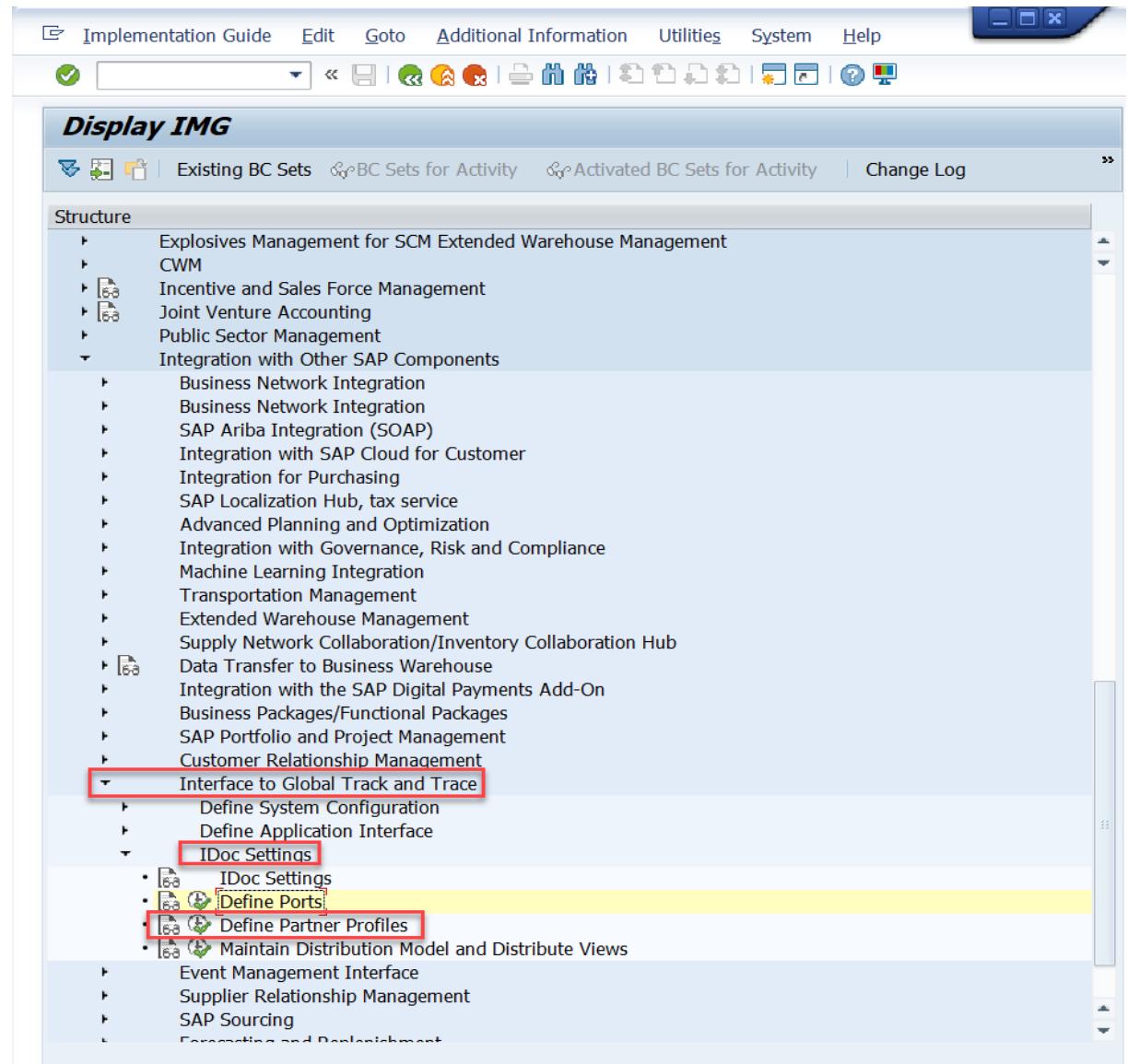


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSSSTEAC	GTT Acceptance Event for SST	ZGTT_SST_FO_EVENT_ACC	application/x-sap.idoc	Version 1.0	Checked
ZGTTSSTTAC	GTT Acceptance Tracked Process for SST	ZGTT_SST_FO_TP_ACC	application/x-sap.idoc	Version 1.0	Checked

STEP 4: Define Partner Profiles

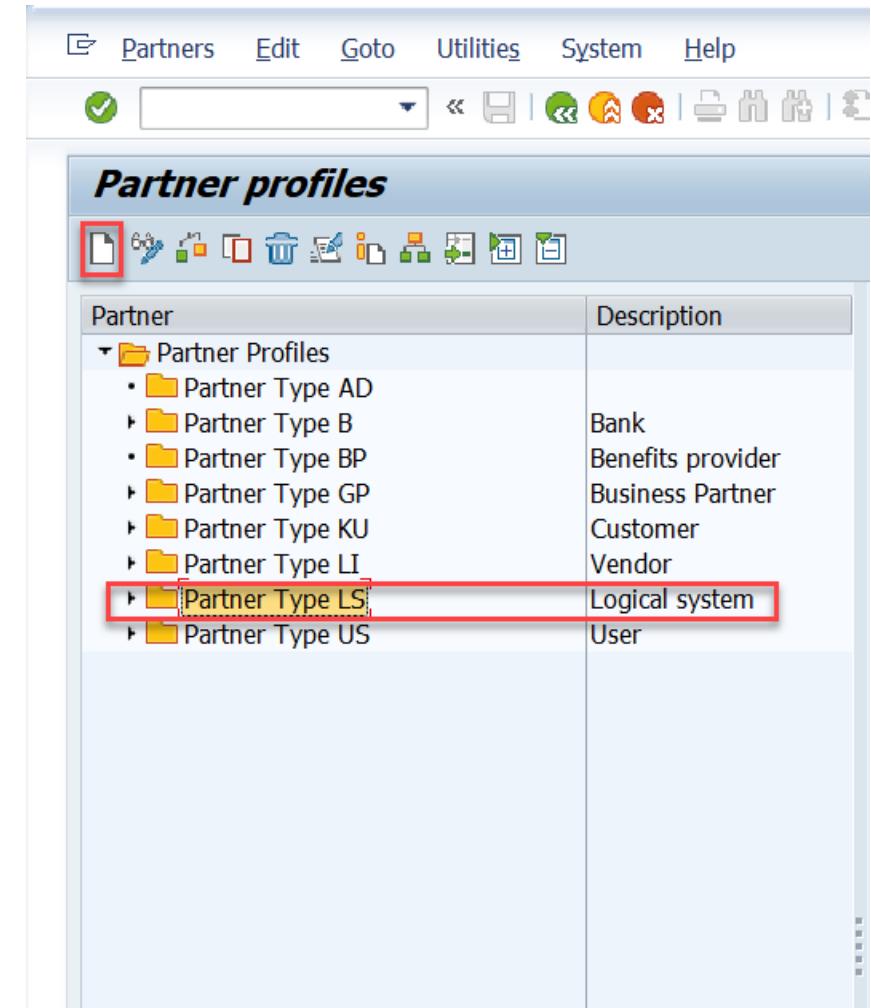
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -
> **Interface to Global Track and Trace** -
> **IDoc Settings**

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



STEP 4: Define Partner Profiles

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



STEP 4: Define Partner Profiles

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

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

The screenshot shows the SAP Partner profiles interface. The top section displays the partner profile details:

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system

The 'Processor' field is highlighted with a red box. Below this, the 'Ty.' field is set to 'US' and the 'Lang.' field is set to 'EN English'. The 'Processor' field contains a placeholder value.

The interface includes tabs for 'Post Processing: Valid Processors', 'Classification', and 'Telephony'. The 'Outbound' tab is selected, showing a table with two rows:

Partner Role	Message Type	Message Va...	Function	Test	Receiver P...	I...	Pa...	Basic Type
	AOPOST			<input type="checkbox"/>	ZGTTSSTTAC	0	EHPOST01	
	EVMSTA			<input type="checkbox"/>	ZGTTSSSTEAC	0	EVMSTA02	

The 'Inbound' tab is also visible at the bottom.

STEP 4: Define Partner Profiles

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

The screenshot shows the SAP Partner profiles interface. At the top, there is a toolbar with various icons. Below the toolbar, the main area is titled "Partner profiles". A partner profile is displayed with the following details:

- Partner No.**: ZGTTSSTAC (highlighted with a red box)
- Type**: LS (Logical system)
- Description**: Logical System For GTT SST - Accept

Below this, there are tabs for "Post Processing: Valid Processors", "Classification", and "Telephony". Under "Post Processing: Valid Processors", the following fields are shown:

- Ty.**: US
- Processor**: [Redacted]
- Lang.**: EN
- Language**: English

The main content area is divided into two sections: "Outbound" and "Inbound". The "Outbound" section contains a table with the following data:

Partner Role	Message Type	Message Va...	Function	Test	Receiver P...	I...	P...	Basic Type
	AOPOST			<input type="checkbox"/>	ZGTTSSTTAC	0		EHPOST01
	EVMSTA			<input type="checkbox"/>	ZGTTSSSTEAC	0		EVMSTA02

At the bottom of the "Outbound" section, there are three buttons: a magnifying glass icon, a plus sign icon (highlighted with a red box), and a minus sign icon.

The "Inbound" section below it is currently empty.

STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

For the tracked Process:

Message Type: AOPOST

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

4-9: Save the configuration

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

Partner profiles: Outbound parameters

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system
Partner Role		
Message Type	EVMSTA	
Message Code		
Message Function		
<input type="checkbox"/> Test		

Outbound Options Message Control Post Processing: Valid Processors Tele...

Receiver Port	ZGTTSSSTEAC	GTT Acceptance Tracked Proc...
Pack. Size		
<input type="checkbox"/> Queue Processing		
Output Mode		
<input checked="" type="radio"/> Pass IDoc Immediately		Output Mode 2
<input type="radio"/> Collect IDocs		

IDoc Type		
Basic Type	EVMSTA02	SCEM: Event Message Input
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		
Application Release		

STEP 4: Define Partner Profiles

4-10: Fill in the Message Type.

For the Tracked Process:

Message Type: AOPOST

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

4-12: Save the configuration

Partner No.	Type	Outbound	Message Type	Receiver Port	IDoc Type
ZGTTSSSTAC	LS	Yes	AOPOST	ZGTTSSSTTAC	EHPOST01
ZGTTSSSTAC	LS	Yes	EVMSTA	ZGTTSSSTEAC	EVMSTA02

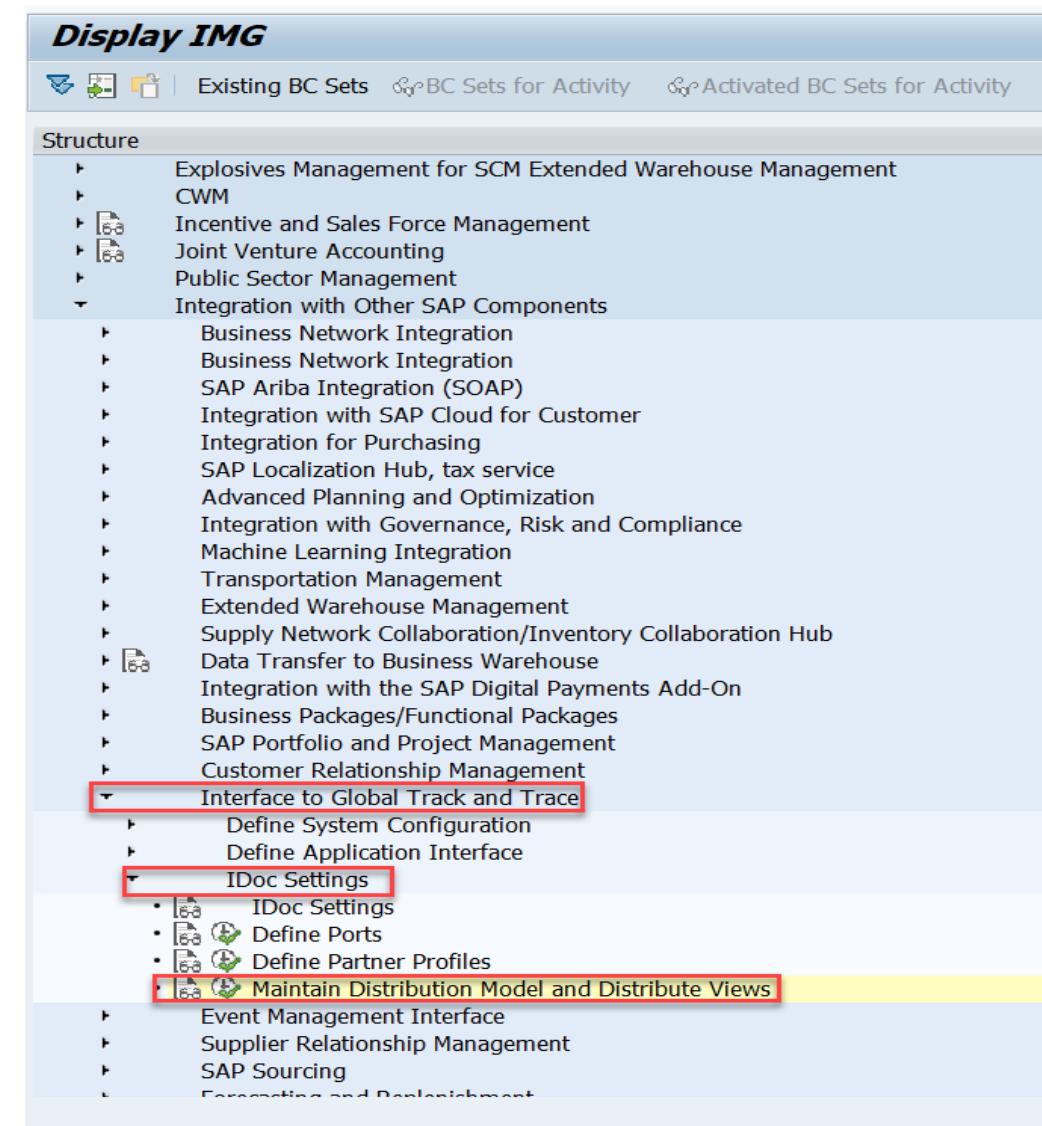
Partner profiles: Outbound parameters

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system
Partner Role		
Message Type	AOPOST	
Message Code		
Message Function		<input type="checkbox"/> Test
Outbound Options		
Receiver Port	ZGTTSSSTTAC	GTT Acceptance Tracked Proc...
Pack. Size		
<input type="checkbox"/> Queue Processing		
Output Mode		
<input checked="" type="radio"/> Pass IDoc Immediately		Output Mode 2
<input type="radio"/> Collect IDocs		
IDoc Type		
Basic Type	EHPOST01	SCEM: Event Handler Posting
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		Application Release

STEP 5: Maintain Distribution Model and Distribute Views

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

- 5-2: Choose activity **Maintain Distribution Model and Distribute Views**



STEP 5: Maintain Distribution Model and Distribute Views

5-3: Click **Edit**, then click **Create Model View** to create a new model view

5-4: Fill in the Short Text and Technical Name of the model view

5-5: Select the new model view and click **Add Message Type** to create a new message

5-6: Fill in the logical systems of Sender and Receiver, and the message type to continue.

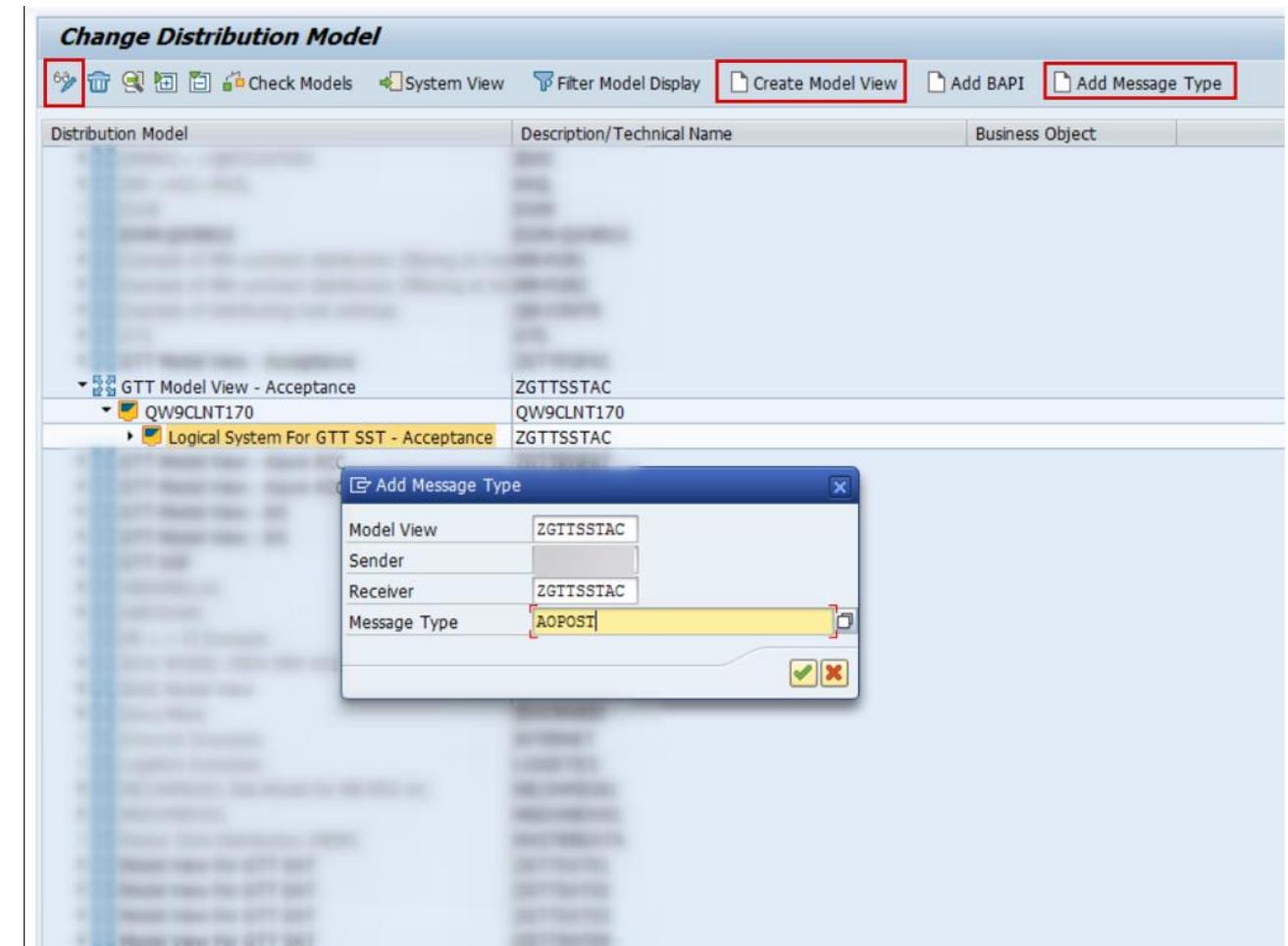
For the event:

Message Type: EVMSTA

For the tracked Process:

Message Type: AOPOST

5-7: Save the configuration



B) Configuration and Implementation

- Basic

B2. Extractor Configuration



STEP 6: Define CI Tenant for GTT

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

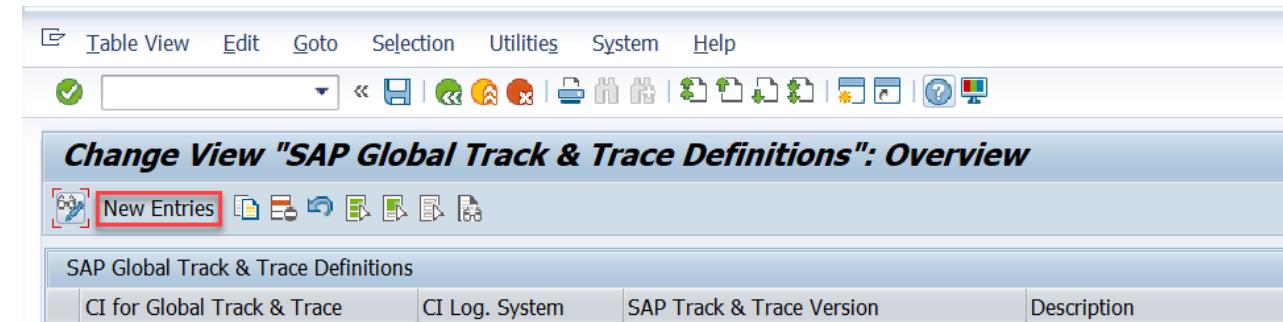
- 6-2: Choose activity
Define CI Tenant for SAP GTT



STEP 6: Define CI Tenant for GTT

6-3: Click **New Entries** to create a new CI tenant for GTT

6-4: Fill in the information for the new CI tenant. The **CI Log. System** is the logical system you created in STEP 2.



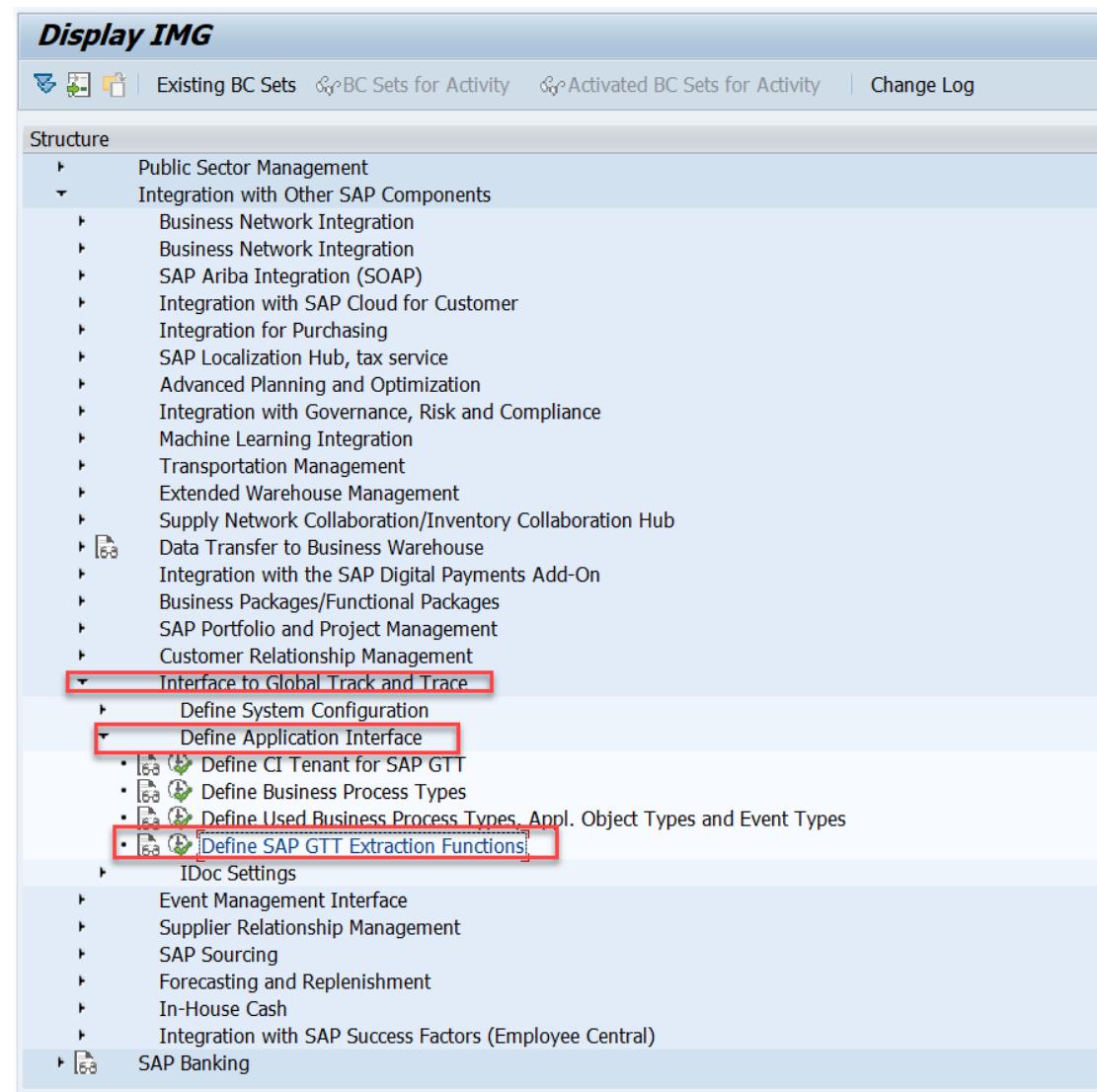
The screenshot shows the SAP Global Track & Trace Definitions table view. The table has columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. One row is present in the table, with the values: ZGTTSSSTAC, ZGTTSSSTAC, Global Track & Trace, and CI For GTT Freight Order Sample APP - Acceptance. The "New Entries" button from the previous screenshot is also visible at the top left of this table view.

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

STEP 7: Define GTT Extraction Functions

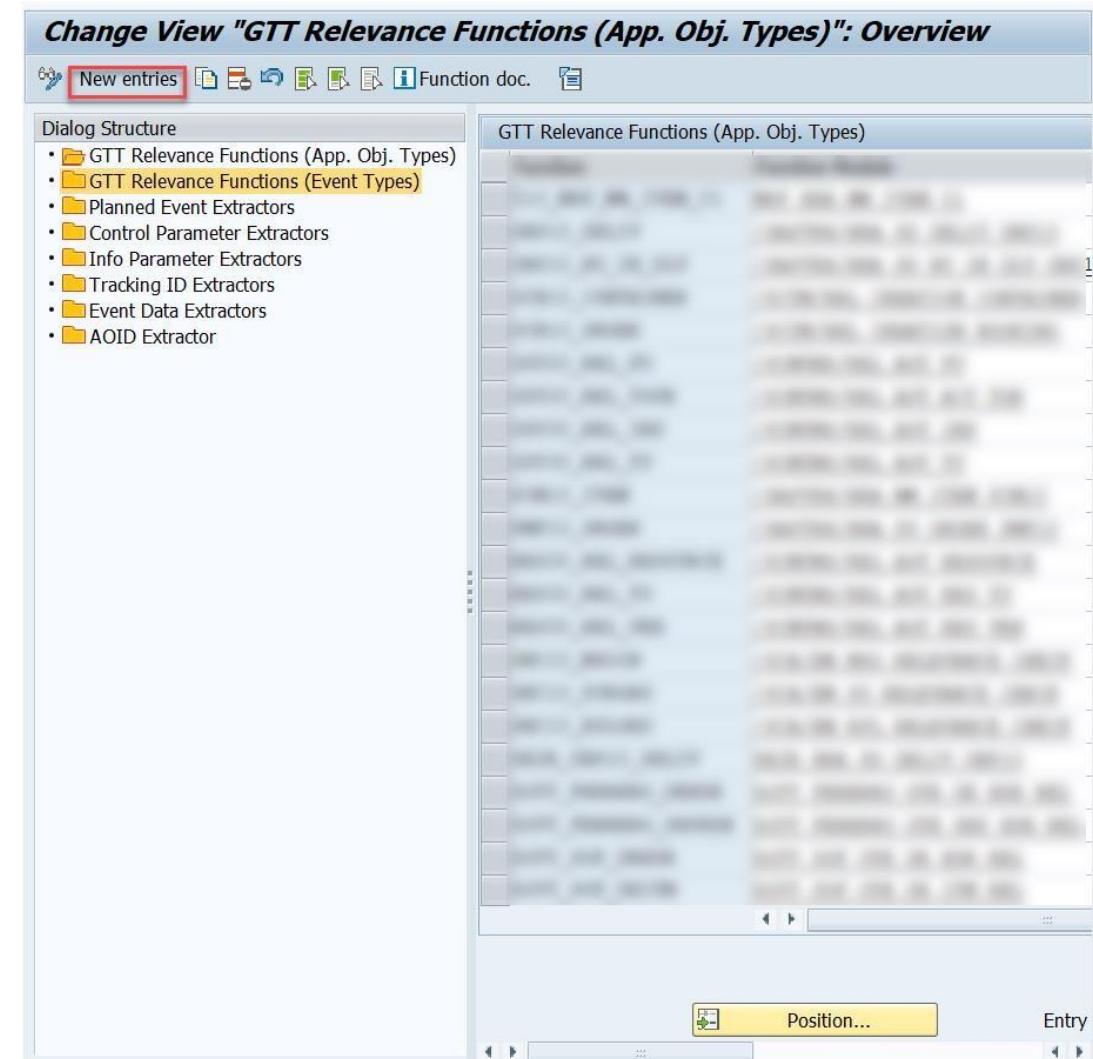
- 7-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace -**
-> **Define Application Interface**

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



STEP 7: Define GTT Extraction Functions

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



STEP 7: Define GTT Extraction Functions

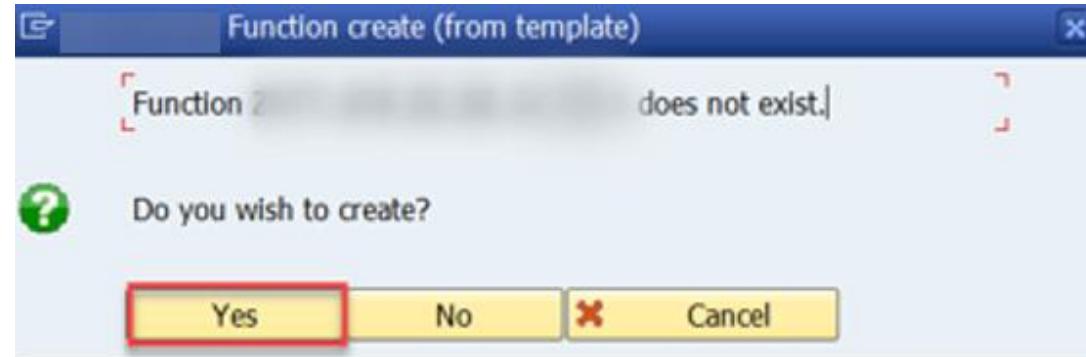
7-4: Input the **Function name** and **Function Module** for the newly created extraction function

7-5: Click **Save**

Change View "GTT Relevance Functions (App. Obj. Types)": Overview		
New entries		
Dialog Structure		
• GTT Relevance Function • GTT Relevance Function • Planned Event Extractor • Control Parameter Extractor • Info Parameter Extractor • Tracking ID Extractors • Event Data Extractors • AOID Extractor		
Function	Function Module	Description
ZSST_GTT_FO_HDR	ZSST_GTT_OTE_FO_HDR_REL	Appl. Object Type Relevance for Freight Order Header

STEP 7: Define GTT Extraction Functions

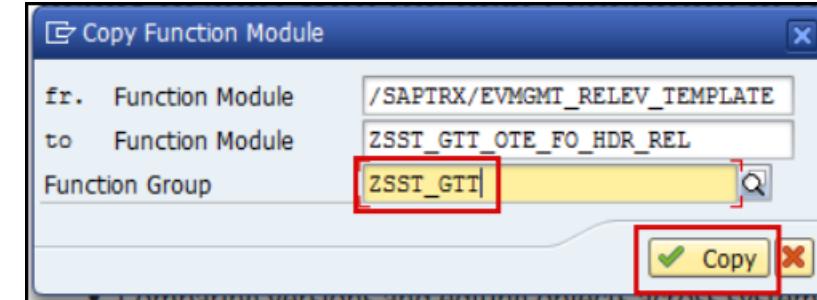
7-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



STEP 7: Define GTT Extraction Functions

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

7-8: Click **Copy**



STEP 7: Define GTT Extraction Functions

7-9: Use T-Code SE80 to check the function module you just created

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

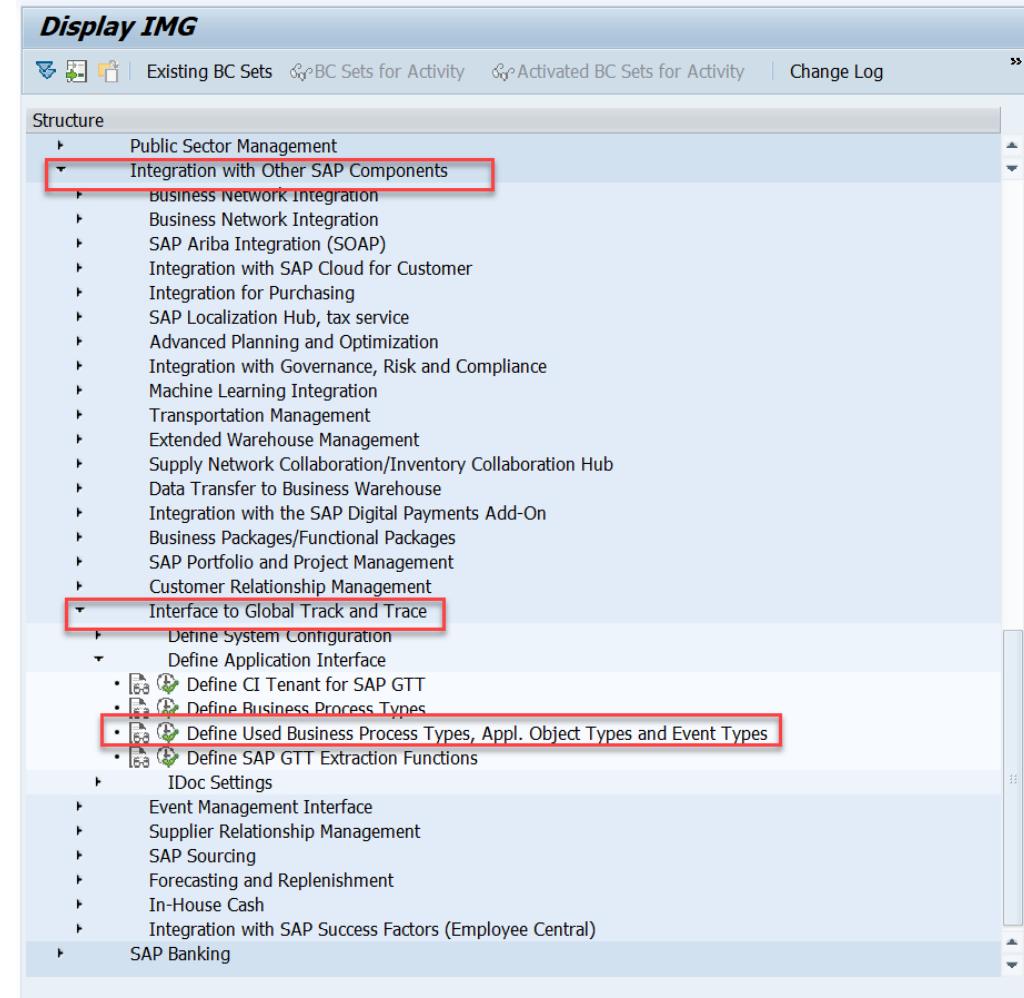
The screenshot shows the SAP Function Builder interface with the following details:

- Repository Browser:** The "Function Group" dropdown is set to "ZSST_GTT".
- Function Module:** The selected function module is "ZSST_GTT_OTE_FO_HDR_REL".
- Source Code:** The code editor displays the ABAP source code for the function module. The code defines a function with various imports, exports, tables, and exceptions. It also includes local data declarations and a TRY...EXCEPT block.
- Code Lines:** The code is numbered from 1 to 29.
- Scope:** The scope is indicated as "\FUNCTION ZSST_GTT_OTE_FO_HDR_REL".
- ABAP:** The language is identified as ABAP.
- Line Number:** The current line is 9.
- Column Number:** The current column is 10.

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

- 8-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace** -
-> **Define Application Interface**

- 8-2: Choose activity **Define Used Business Process Types, Appl. Object Types and Event Types**



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

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

In the following:

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

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

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

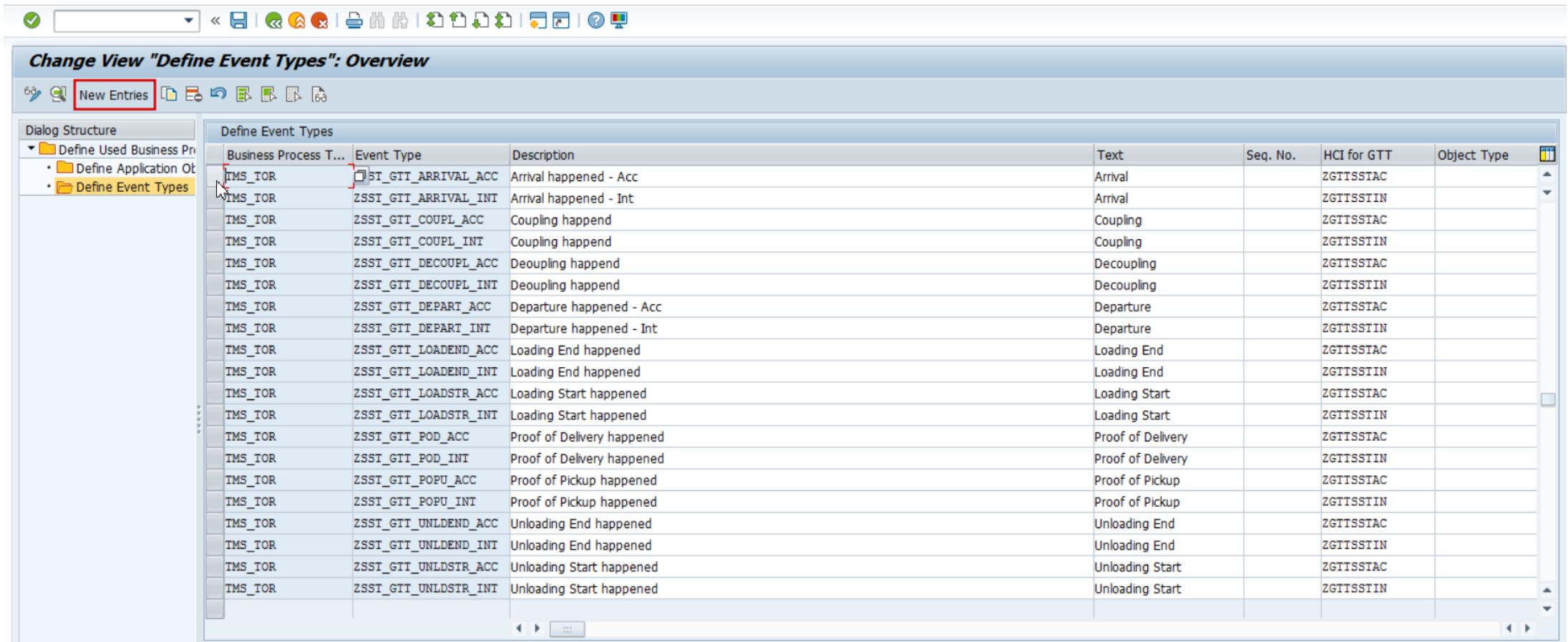
8-3: Choose the business process type from the **Define Used Business Process Types** on the right side

8-4: Double click **Define Event Types**

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

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

8-5: Click **New Entries** to create a new event type



The screenshot shows the SAP interface for defining event types. The title bar reads "Change View 'Define Event Types': Overview". The left sidebar, titled "Dialog Structure", shows a tree view with "Define Business Pro..." expanded, and "Define Event Types" selected. A red box highlights the "New Entries" button in the toolbar at the top. The main area is a grid titled "Define Event Types" with columns: Business Process T..., Event Type, Description, Text, Seq. No., HCI for GTT, and Object Type. The grid lists various event types such as ST_GTT_ARRIVAL_ACC, ZSST_GTT_ARRIVAL_INT, and ZSST_GTT_COUPL_ACC, each with a corresponding description and other metadata.

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

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

8-6: Fill in the **Event Type** and **Text** fields

8-7: Fill in the information required in

the **General Data** tab.

HCI for GTT is the CI Tenant you created in STEP 6.

Event Function is the extractor function you created in STEP 7.

8-8: Check **GTT Relevant**

Bus. Proc. Type	TMS_TOR
Event Type	ZSST_GTT_ARRIVAL_ACC
Text	Arrival

General Data Control Tables Global Track & Trace Relevance

Sequencing / Destination

Seq. No.

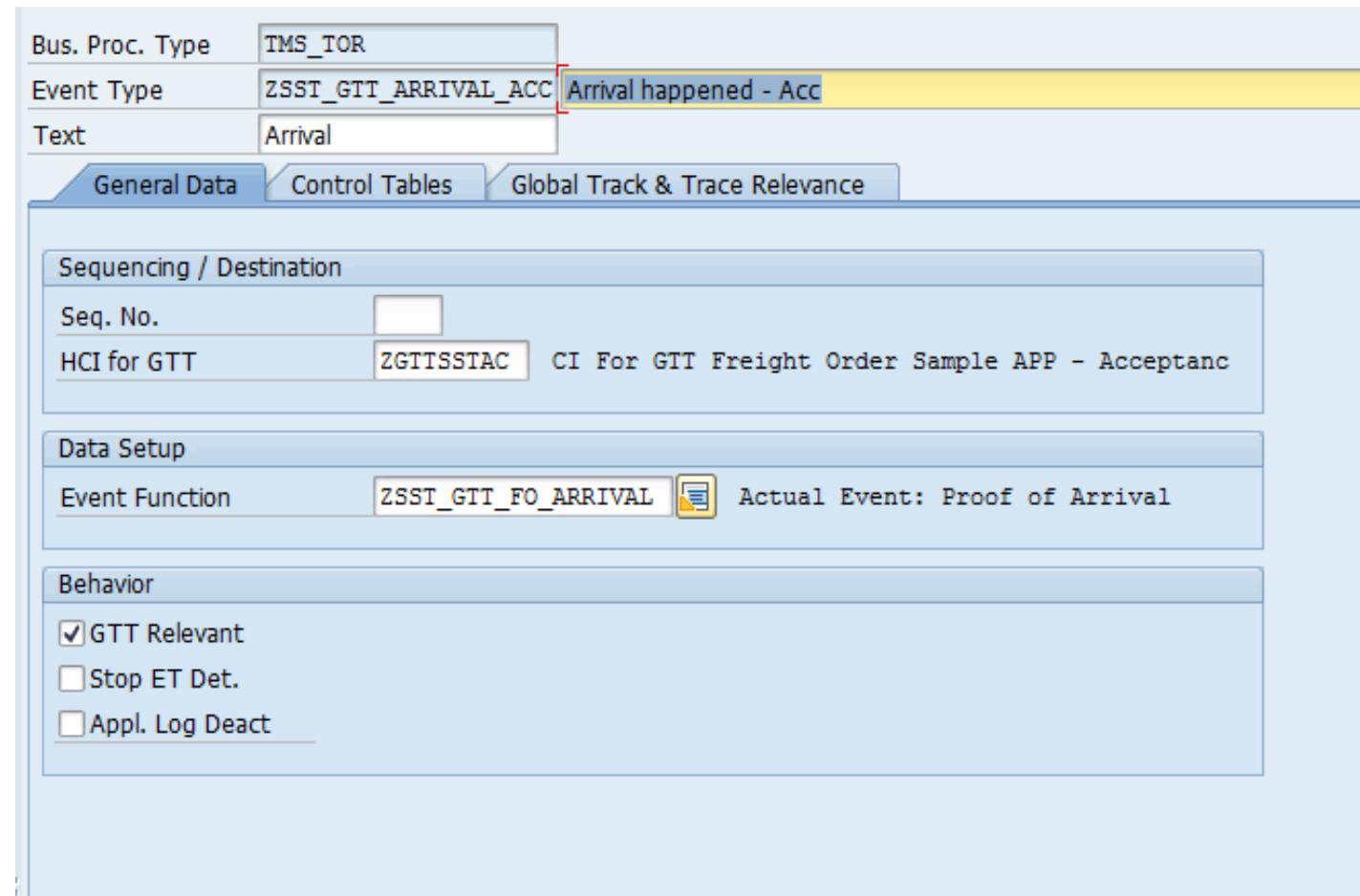
HCI for GTT ZGTTSSSTAC CI For GTT Freight Order Sample APP - Acceptanc

Data Setup

Event Function ZSST_GTT_FO_ARRIVAL Actual Event: Proof of Arrival

Behavior

GTT Relevant
 Stop ET Det.
 Appl. Log Deact

A screenshot of the SAP Fiori interface for defining business process types. The top section shows basic configuration: Bus. Proc. Type (TMS_TOR), Event Type (ZSST_GTT_ARRIVAL_ACC) with a tooltip 'Arrival happened - Acc', and Text (Arrival). Below this is a tab navigation bar with 'General Data' selected, followed by 'Control Tables' and 'Global Track & Trace Relevance'. The main area contains three sections: 'Sequencing / Destination' (Seq. No. empty, HCI for GTT set to ZGTTSSSTAC, CI For GTT Freight Order Sample APP - Acceptanc), 'Data Setup' (Event Function set to ZSST_GTT_FO_ARRIVAL, Actual Event: Proof of Arrival), and 'Behavior' (checkboxes for GTT Relevant (checked), Stop ET Det., and Appl. Log Deact.).

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

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

Caution:

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

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

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

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

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

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

Click **Save**.

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

General Data Control Tables Global Track & Trace Relevance

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

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

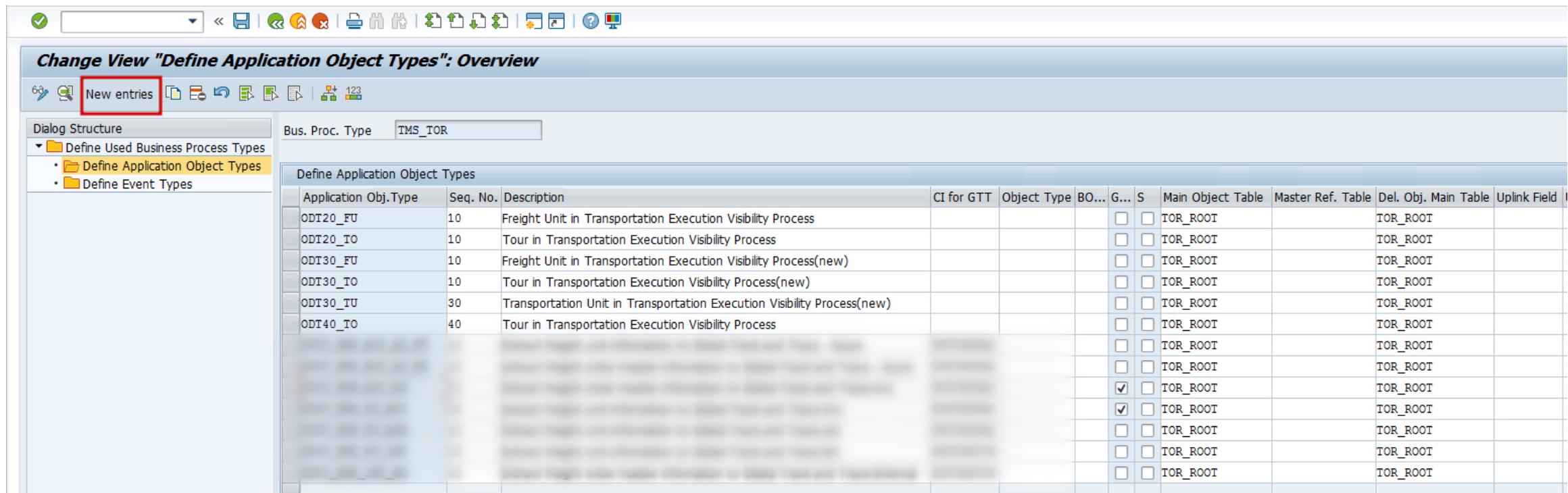
8-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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

8-13: Click **New Entries** to create a new AOT



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

8-14: Fill in the **Appl. Obj. Type** and **Text** fields

8-15: Fill in the information required in the **General Data** tab.

HCI for GTT is the CI Tenant you created in STEP 6.

Event Function is the extractor function you created in STEP 7.

8-16: Check **GTT Relevant**

The screenshot shows the SAP Fiori interface for defining business process types. At the top, there are three input fields: 'Bus. Proc. Type' (TMS_TOR), 'Appl. Obj. Type' (ZGTT_SHP_ACC_HD), and 'Text'. A tooltip for 'Appl. Obj. Type' says 'Extract freight order header information to Global Track and Trace-Acc'. Below these are five tabs: General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The 'Object Identification' tab is selected. Under 'Object Identification', there are three sections: Sequencing / Destination, Business Object Reference, and Behavior. In the 'Sequencing / Destination' section, 'Seq. No.' is 10 and 'CI for GTT' is ZGTTSSSTAC. In the 'Business Object Reference' section, 'Object Type' is empty and 'BO Setup Fnct.' has a value. In the 'Behavior' section, the checkbox 'GTT Relevant' is checked, while 'Stop AO Determ.' and 'Appl. Log Deact' are unchecked. An 'Alt. BusProcType' field is also present.

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

8-17: Fill in the **Main Object Table** and **Master Table**

Caution:

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

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

The screenshot shows a configuration screen for a business process type. At the top, there are three input fields: 'Bus. Proc. Type' (TMS_TOR), 'Appl. Obj. Type' (ZGTT_SHP_ACC_HD), and 'Text'. A tooltip for the application object type says 'Extract freight order header information to Global Track and Trace-Acc'. Below these are five tabs: General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. Under 'Object Identification', there are two sections: 'Data Source for Created and Updated Objects' (Main Obj. Table: TOR_ROOT, Master Table: empty) and 'Data Source for Deleted Objects' (Del.Obj. Table: TOR_ROOT). There is also a section for 'Reference Between Main and Master Table' which includes fields for 'First Field Reference from Main to Master Table' and 'Second Field Reference from Main to Master Table'.

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

8-18: In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need.

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

Click **Save**.

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_ACC_HD
Text	Extract freight order header information to Global Track and Trace-Acc

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

GTT Rel. Method Check Function (Function Module) ▾

GTT Rel. Function ZSST_GTT_FO_HDR  Appl. Object Type Relevance for Freight Order Header

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

8-19: In the **Object Identification** tab, choose the **AOID Method** and **Cntrl Tab Type**

Click **Save**.

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

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

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

Method for determination of AOID

AOID Method Determine from Field

Application Object ID Source

First Field to Build Appl. Obj. ID	Cntrl Tab. Type	Main Object Table
	AO ID Field	TOR_ID
Second Field to Build Appl. Obj. ID	Cntrl Tab. Type	
	AO ID Field	

Determine AOID By Function

AOID Function

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

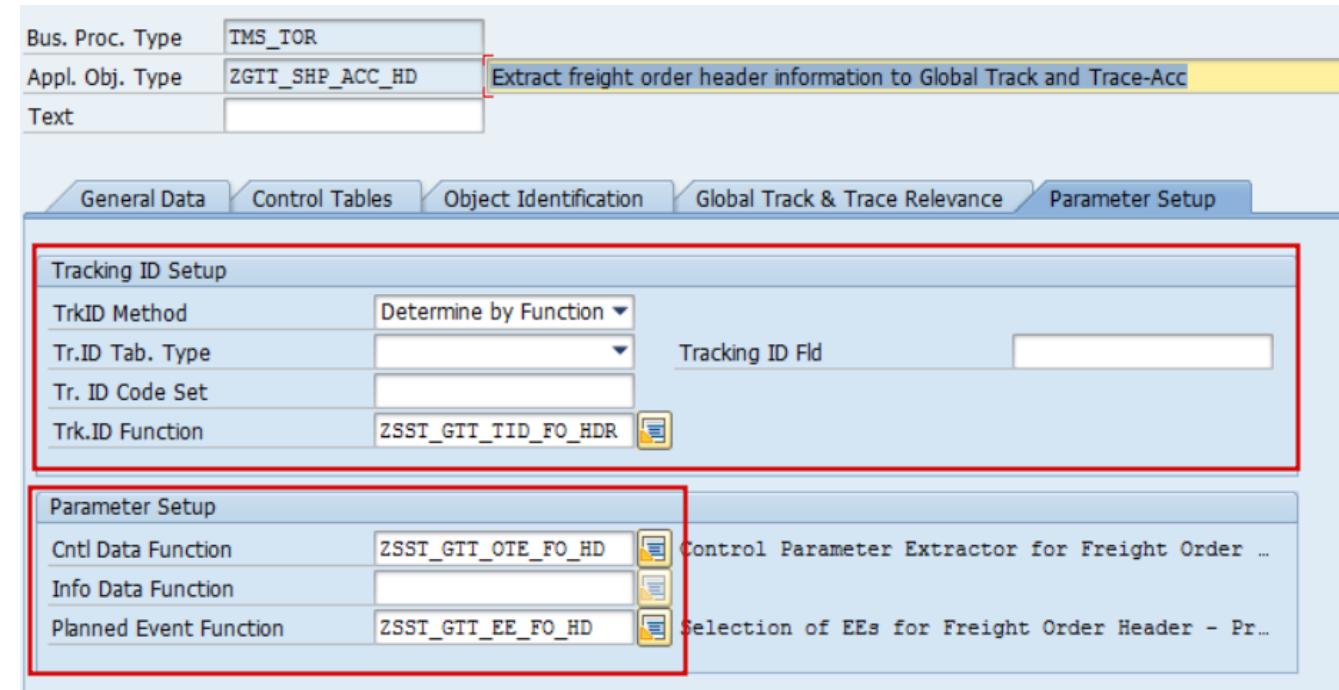
8-20: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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

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

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

Click **Save**.



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

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

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameters	
Sequencing / Destination	
Seq. No.	10
CI for GTT	ZGTTSSSTAC CI For GTT Freight Order Sample APP - Acceptance
Business Object Reference	
Object Type	
BO Setup Fnct.	
Behavior	
<input checked="" type="checkbox"/> GTT Relevant	
<input type="checkbox"/> Stop AO Determ.	
<input type="checkbox"/> Appl. Log Deact	
Alt. BusProcType	

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameters	
Data Source for Created and Updated Objects	
Main Obj. Table	TOR_ROOT
Master Table	
Data Source for Deleted Objects	
Del.Obj. Table	TOR_ROOT
Reference Between Main and Master Table	
First Field Reference from Main to Master Table	
Second Field Reference from Main to Master Table	

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

The image displays two side-by-side screenshots of the SAP Fiori interface, likely from the SAP S/4HANA or SAP ERP system, illustrating the configuration of Business Process Types (BPT) and Application Object Types (AOT).

Screenshot 1 (Left): General Data Configuration

- General Data:**
 - Bus. Proc. Type: TMS_TOR
 - Appl. Obj. Type: ZGTT_SHP_FU_ACC
 - Text: Extract freight unit information to Global Track and Trace-Acc
- Control Tables:**
 - Method for determination of AOID
 - AOID Method: Determine from Field
- Object Identification:**
 - First Field to Build Appl. Obj. ID: Cntrl Tab. Type: Main Object Table; AO ID Field: TOR_ID
 - Second Field to Build Appl. Obj. ID: Cntrl Tab. Type: (empty); AO ID Field: (empty)
- Global Track & Trace Relevance:**
 - Determine AOID By Function: AOID Function: (empty)

Screenshot 2 (Right): Parameter Setup Configuration

- General Data:**
 - Bus. Proc. Type: TMS_TOR
 - Appl. Obj. Type: ZGTT_SHP_FU_ACC
 - Text: Extract freight unit information to Global Track and Trace-Acc
- Control Tables:**
 - Tracking ID Setup:
 - TrkID Method: Determine by Function
 - Tr.ID Tab. Type: (empty)
 - Tr. ID Code Set: (empty)
 - Trk.ID Function: ZSST_GTT_TID_FO_HDR (with a help icon)
 - Parameter Setup:
 - Cntl Data Function: ZSST_GTT_OTE_FO_HD (with a help icon)
 - Info Data Function: (empty)
 - Planned Event Function: ZSST_GTT_EE_FO_HD (with a help icon)
- Object Identification:**
 - Bus. Proc. Type: TMS_TOR
 - Appl. Obj. Type: ZGTT_SHP_FU_ACC
 - Text: Extract freight unit information to Global Track and Trace-Acc
- Global Track & Trace Relevance:**
 - GTT Rel. Method: Check Function (Function Module)
 - GTT Rel. Function: ZSST_GTT_FO_HDR (with a help icon)
- Parameter Setup:**
 - (This tab is visible at the bottom of the right-hand screenshot)

C) Download ABAP Code from GitHub

C1. Initial Download ABAP code from GitHub(Only for TSO)

C2. Update ABAP code from GitHub(Only for TSO)

C3. Download Another ABAP code from GitHub(TPO)

C4. Initial Download ABAP code from GitHub(Include TSO/TPO/TS)



C) Download ABAP Code from GitHub

C1. Initial Download ABAP code from GitHub(Only for TSO)



STEP 1: Install ABAPGit

You need to install ABAPGit before downloading codes from GitHub.

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

[Improve this page](#)

Summary #

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

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

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

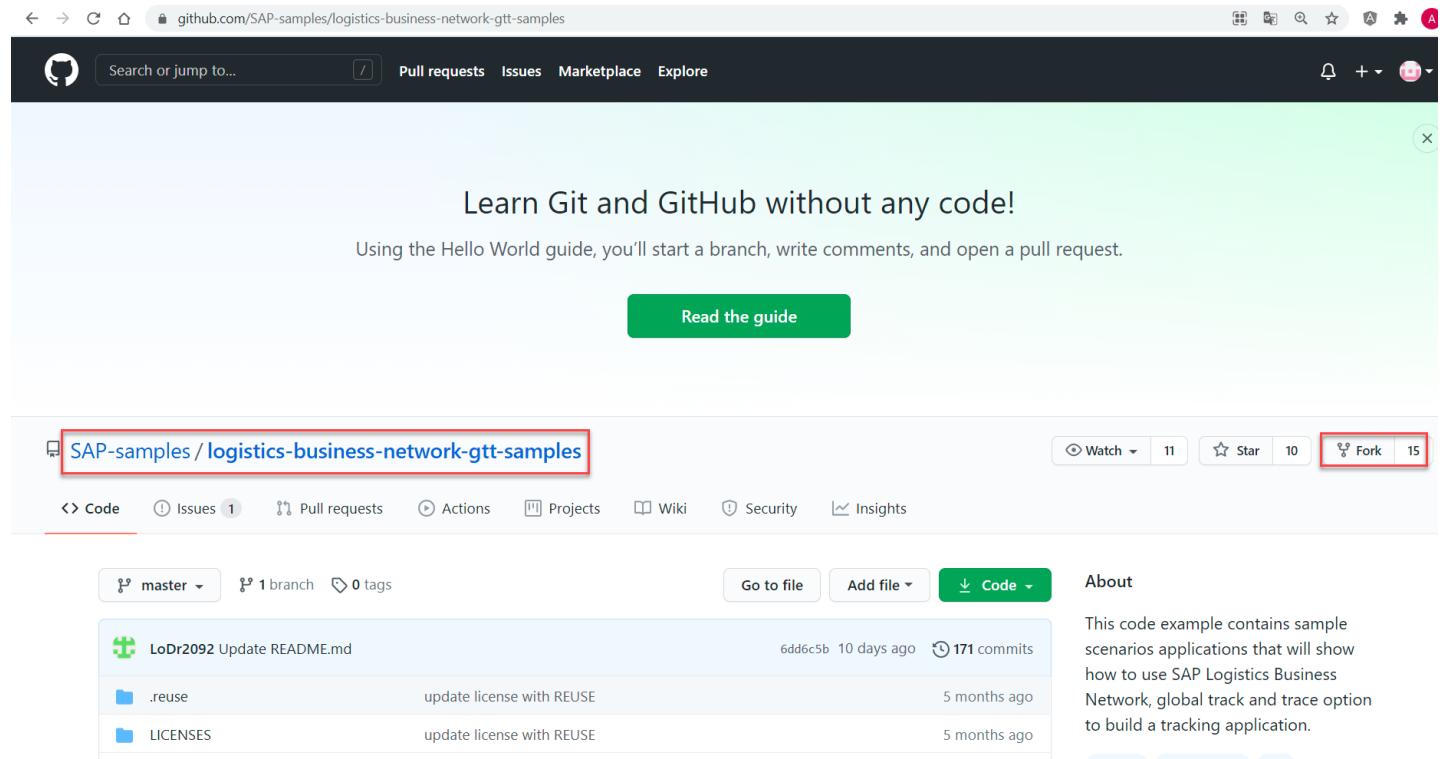
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `.$ZABAPGIT`).

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

STEP 2: Fork Sample code Repository

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

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



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

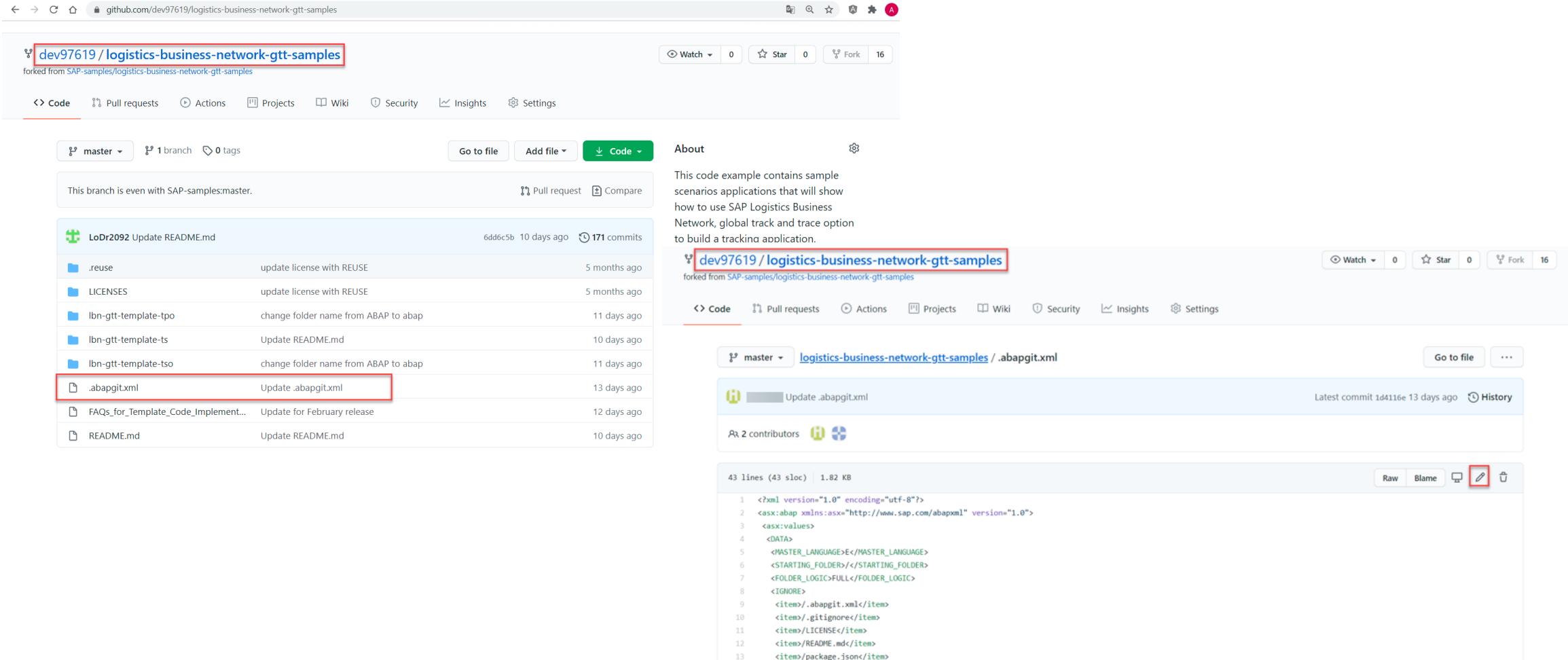
3-1: In the user’s account repository, click the file ‘.abapgit.xml’

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one for '.abapgit.xml' which is highlighted with a red box. The commit message is 'Update .abapgitxml'. Other commits include updates to README.md, LICENSES, and folder names. The repository has 0 stars, 0 forks, and 16 issues. The 'About' section describes the code example as containing sample scenarios for SAP Logistics Business Network, global track and trace options. It includes links to 'Readme', 'Releases', and 'Packages'.

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

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

3-2: Click  button to edit the file



The screenshot shows two GitHub repository pages. The top page is for the repository `dev97619 / logistics-business-network-gtt-samples`, which is a fork of `SAP-samples/logistics-business-network-gtt-samples`. It displays a list of commits, including one from `LoDr2092` that updated the `README.md` file. The bottom page is a detailed view of the `.abapgit.xml` file within the same repository. The file content is as follows:

```
43 lines (43 sloc) | 1.82 KB
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>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>’ like 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 specific line of code, '<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>', is highlighted with a red box. To the right, a 'Commit changes' dialog is displayed. The dialog contains a text input field with 'Update .abapgit.xml', a description input field with 'Add an optional extended description...', and two radio button options for committing. The first option, 'Commit directly to the master branch', is selected and has a red box around it. The second option, 'Create a new branch for this commit and start a pull request', is also present. At the bottom are 'Commit changes' and 'Cancel' buttons, with the 'Commit changes' button also having a red box.

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

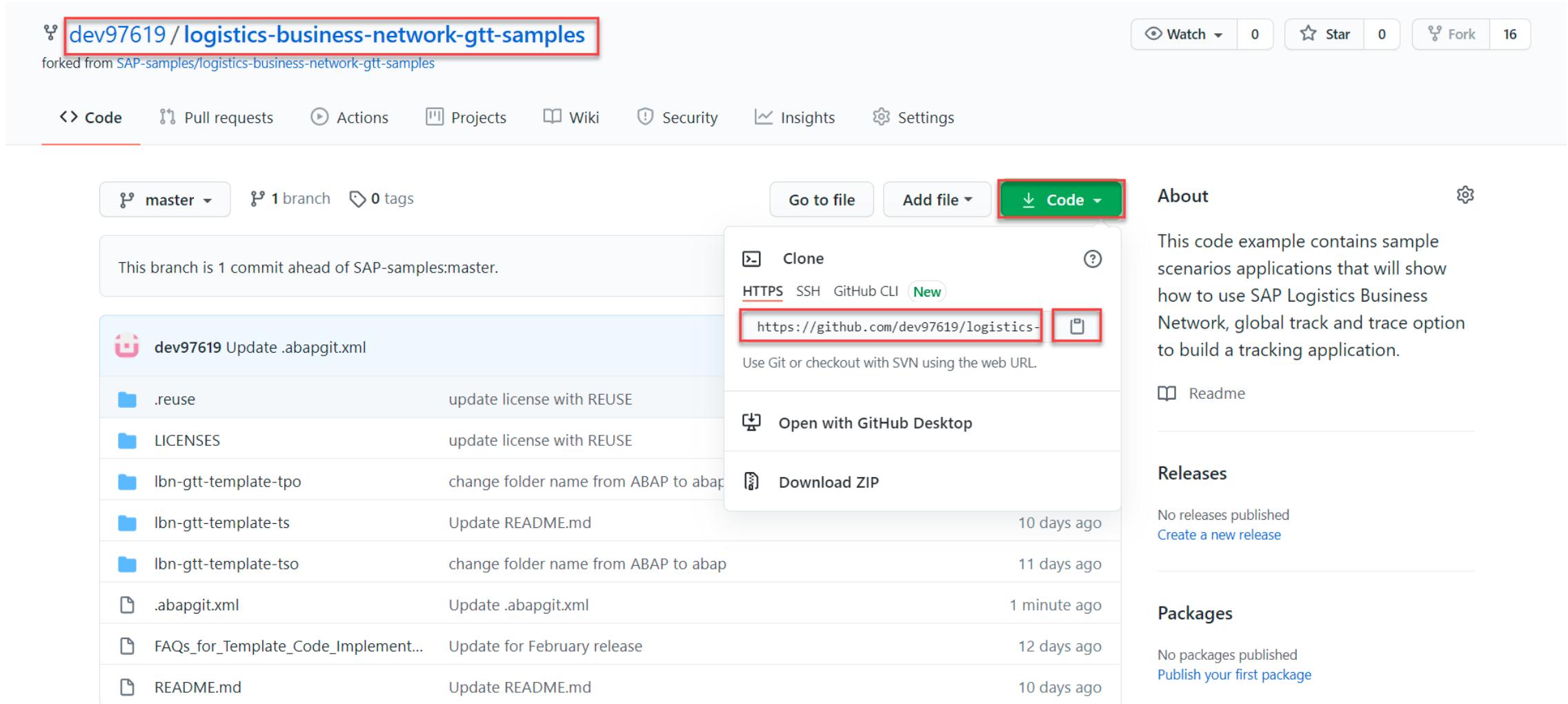
-o Commit directly to the master branch.

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

Commit changes Cancel

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

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



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository has 0 stars and 16 forks. The 'Code' tab is selected. A commit from 'dev97619' titled 'Update .abapgit.xml' is highlighted. The 'Code' dropdown menu is open, showing the 'Clone' section with the URL <https://github.com/dev97619/logistics-business-network-gtt-samples>. The URL is highlighted with a red box, and a copy icon is also highlighted with a red box.

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

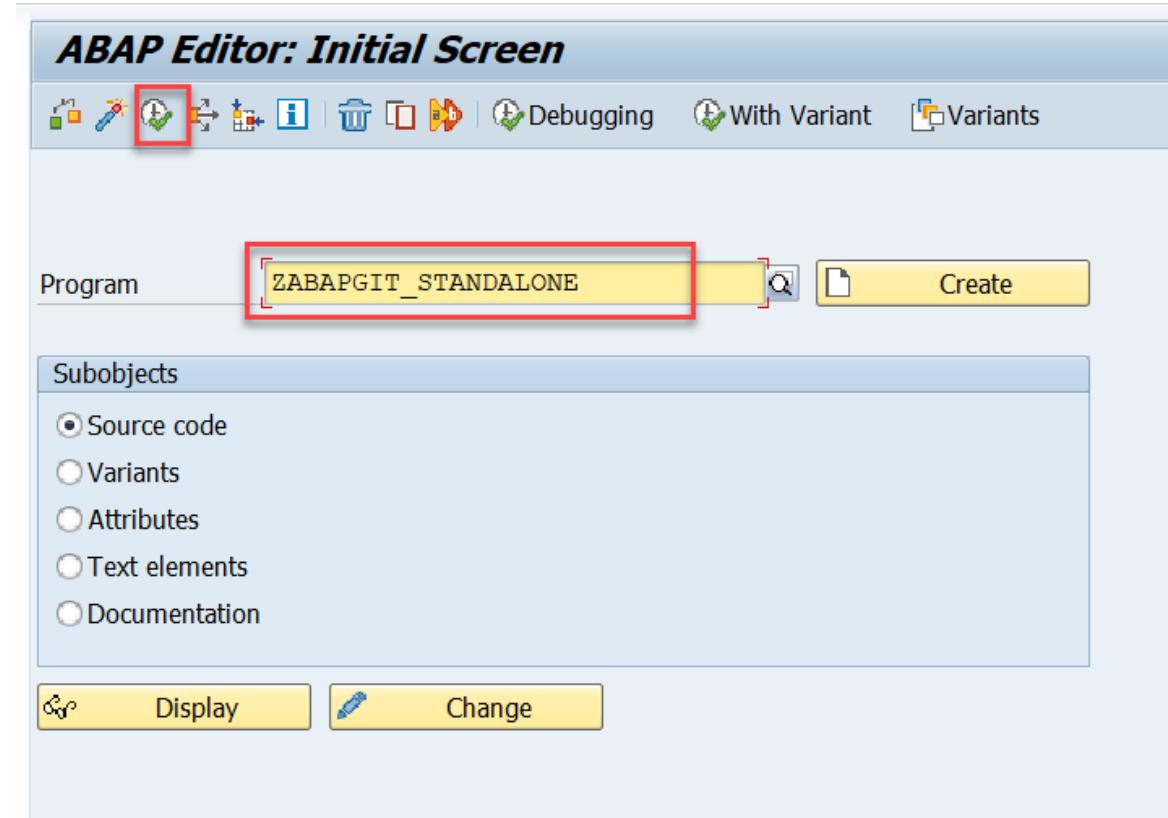
No releases published
Create a new release

No packages published
Publish your first package

STEP 4: Download ABAP code from GitHub

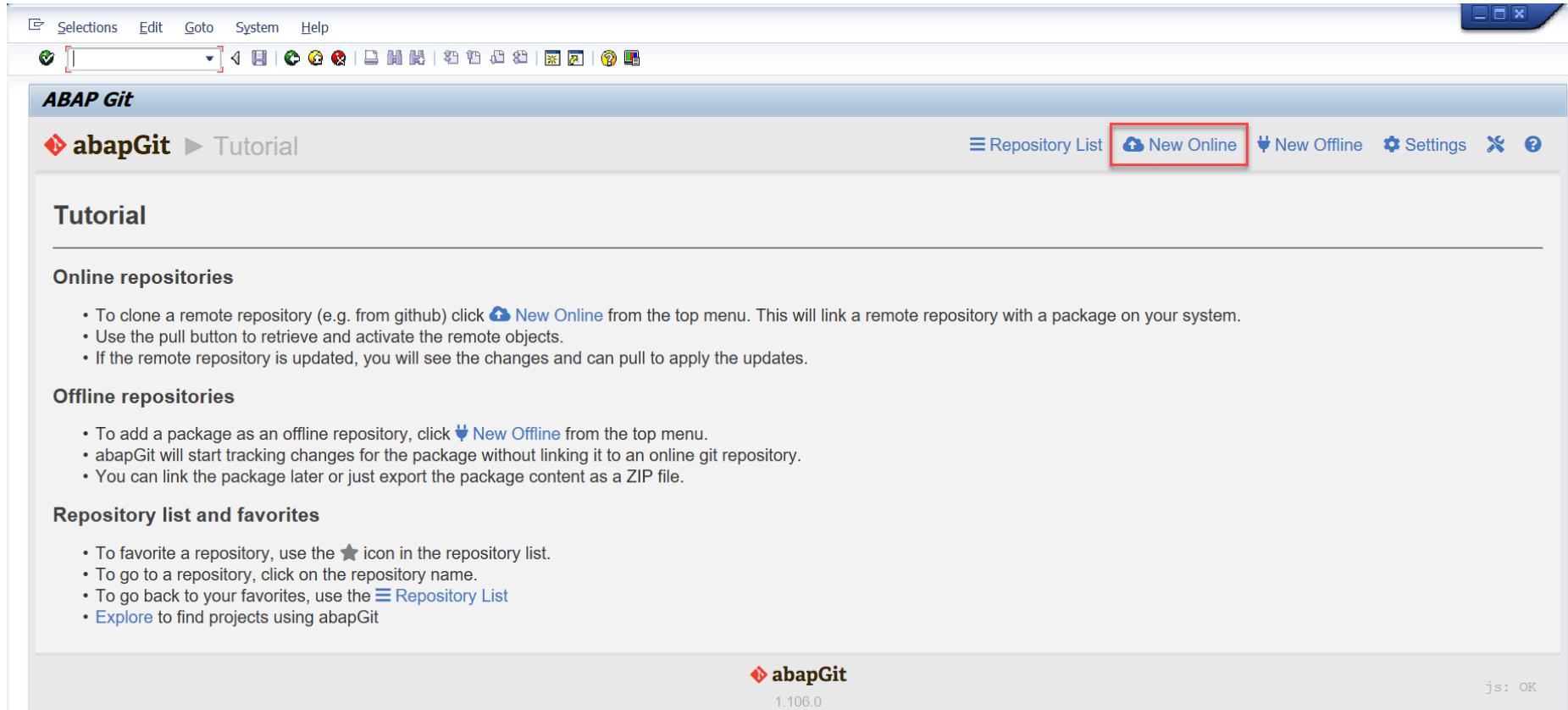
4-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE

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



STEP 4: Download ABAP code from GitHub

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



STEP 4: Download ABAP code from GitHub

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

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

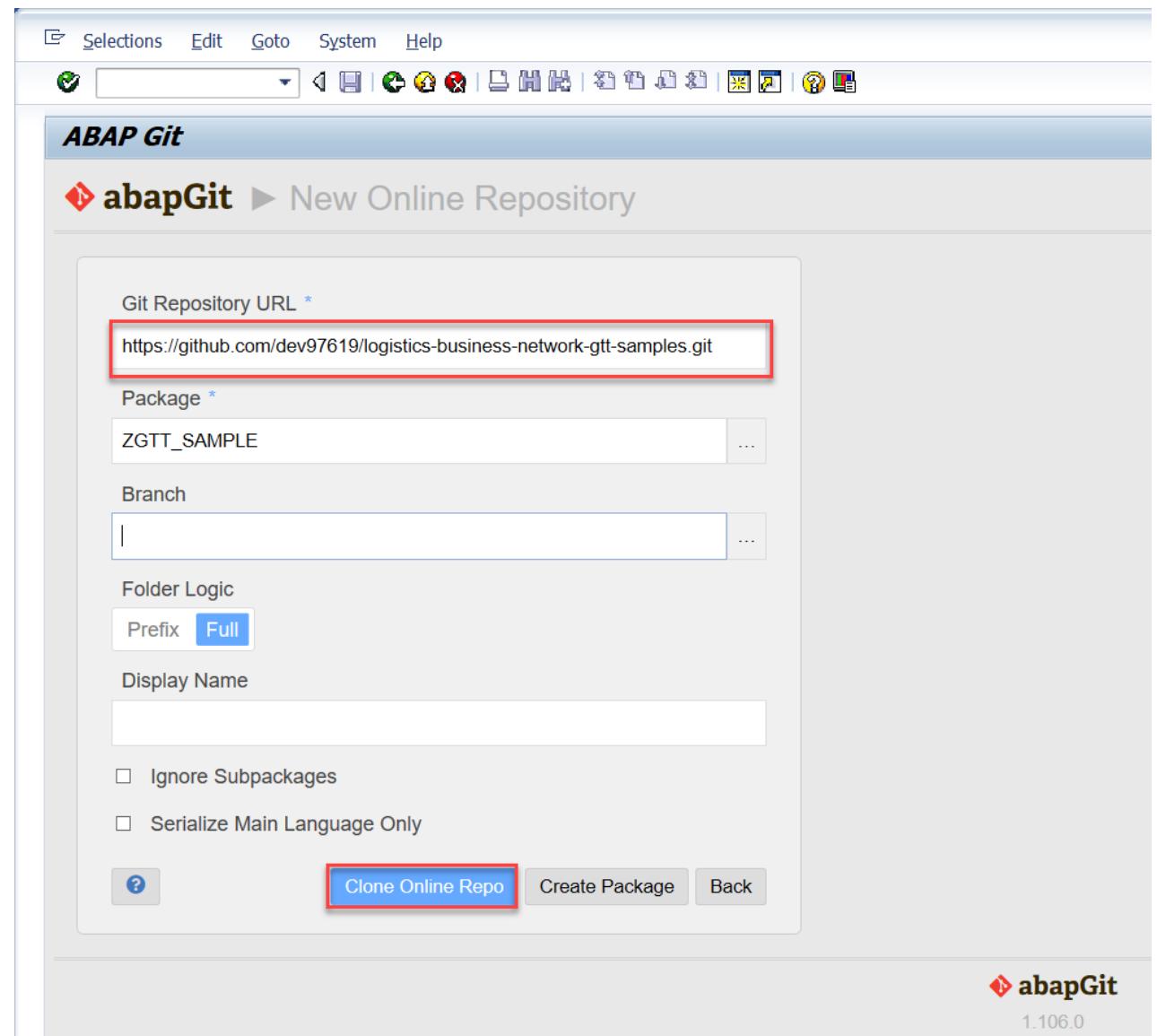
Caution:

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

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

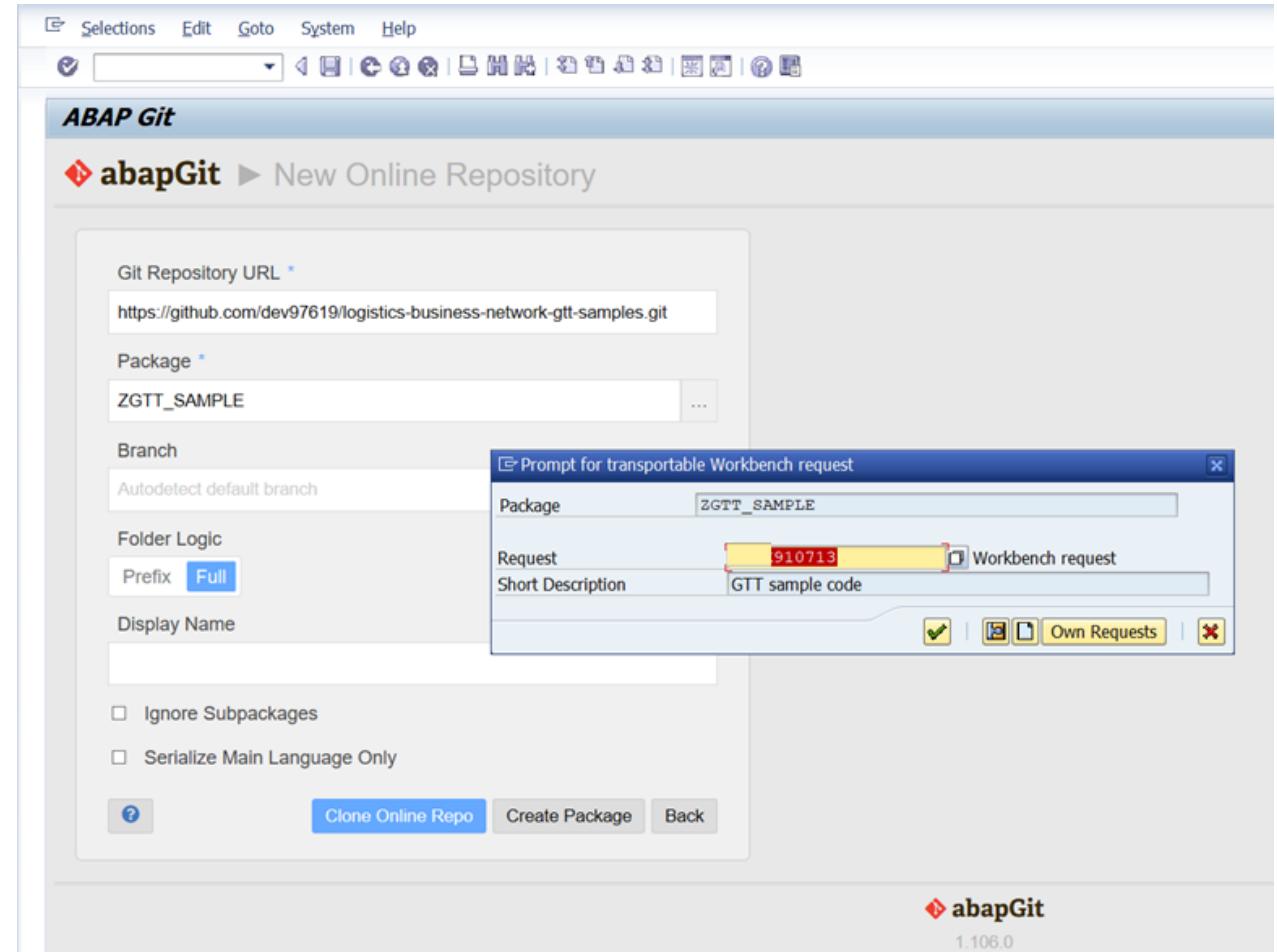
4-6: Set *Full* for **Folder Logic**

4-7: Click **Clone online repo** to download the code



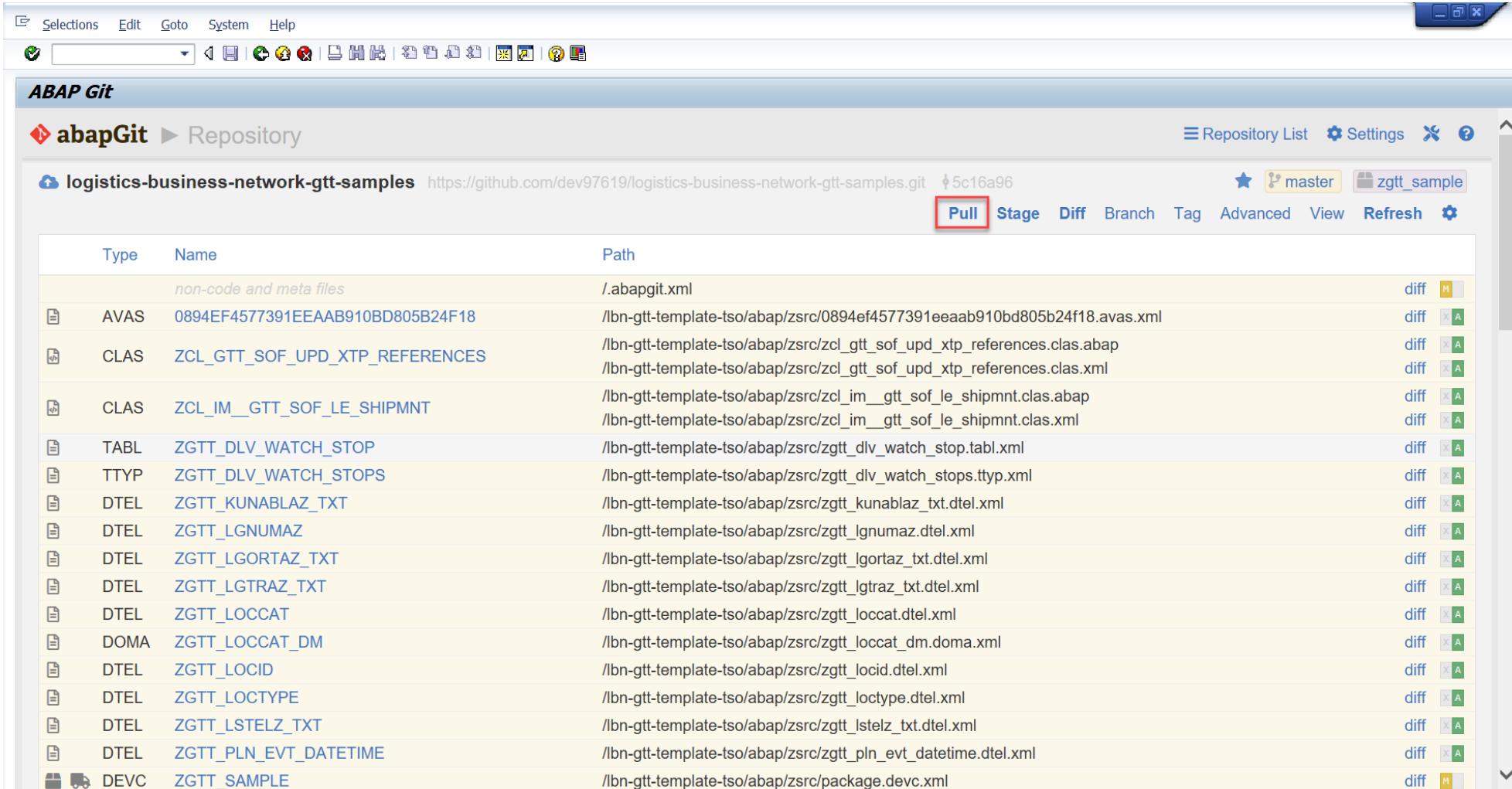
STEP 4: Download ABAP code from GitHub

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



STEP 4: Download ABAP code from GitHub

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

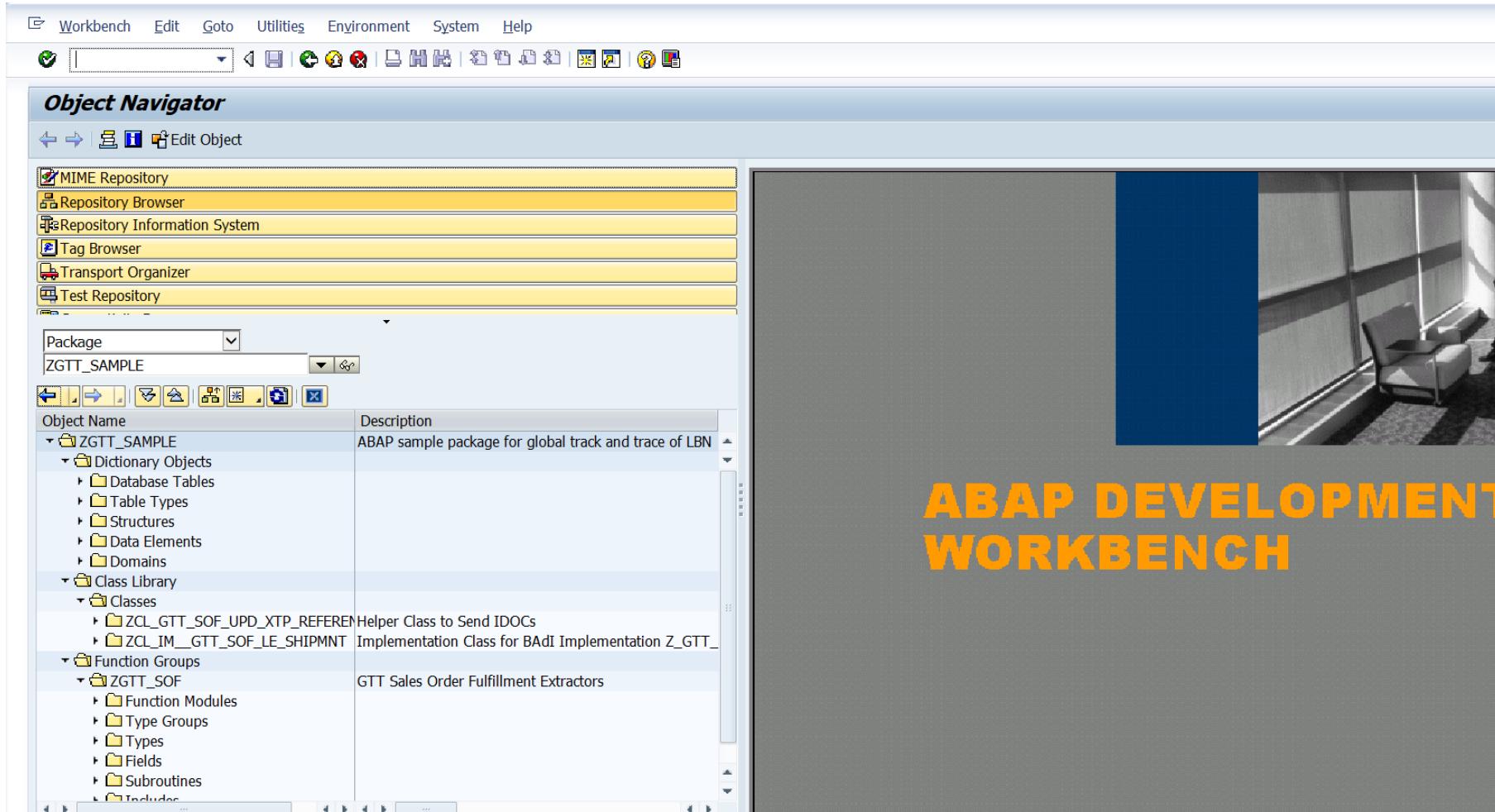


The screenshot shows the ABAP Git interface within an SAP application. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. The main area is titled 'ABAP Git' and shows a repository named 'abapGit' under 'Repository'. The URL is listed as 'logistics-business-network-gtt-samples' with the commit hash '5c16a96'. A toolbar at the top right includes icons for Repository List, Settings, Refresh, and Help. Below the toolbar, there are buttons for 'Pull' (highlighted with a red box), Stage, Diff, Branch, Tag, Advanced, View, Refresh, and Settings. The main content is a table with columns 'Type', 'Name', and 'Path'. The 'Type' column includes icons for AVAS, CLAS, TABL, TTYP, DTEL, DOMA, and DEV. The 'Name' column lists various file names like '0894EF4577391EEAAB910BD805B24F18', 'ZCL_GTT_SOUPD_XTP_REFERENCES', etc. The 'Path' column shows the full path for each file, such as '/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml'. To the right of the table, there are 'diff' buttons and small preview icons.

Type	Name	Path	diff
	non-code and meta files	/.abapgit.xml	[diff]
AVAS	0894EF4577391EEAAB910BD805B24F18	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	[diff]
CLAS	ZCL_GTT_SOUPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap	[diff]
		/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	[diff]
CLAS	ZCL_IM_GTT_SOUPD_XTP_REFERENCES	/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_LGNUMAZ	/lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml	[diff]
DTEL	ZGTT_LGORTAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml	[diff]
DTEL	ZGTT_LGTRAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	[diff]
DTEL	ZGTT_LOCCAT	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat.dtel.xml	[diff]
DOMA	ZGTT_LOCCAT_DM	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat_dm.doma.xml	[diff]
DTEL	ZGTT_LOCID	/lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	[diff]
DTEL	ZGTT_LOCTYPE	/lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	[diff]
DTEL	ZGTT_LSTELZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	[diff]
DTEL	ZGTT_PLN_EVT_DATETIME	/lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml	[diff]
DEV	ZGTT_SAMPLE	/lbn-gtt-template-tso/abap/zsrc/package.devc.xml	[diff]

STEP 4: Download ABAP code from GitHub

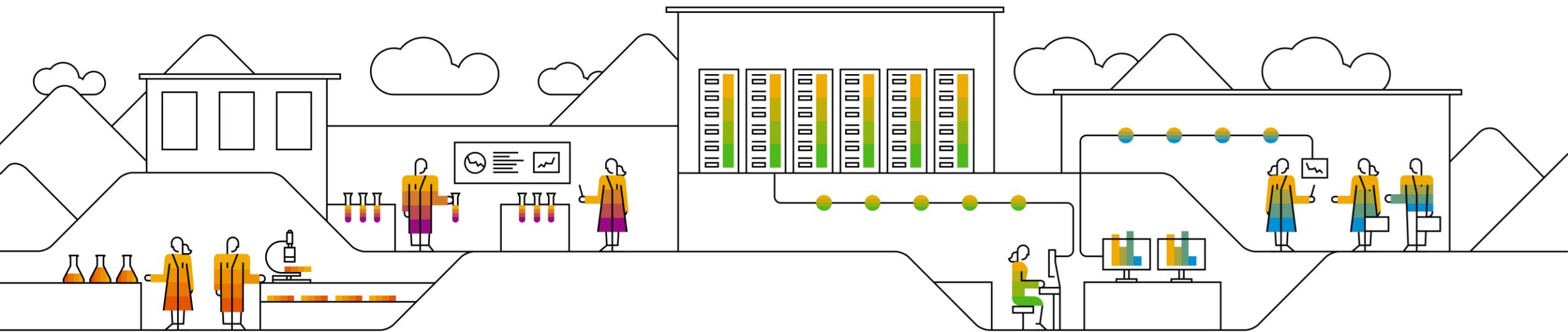
4-10: After you download the code, you can check them with T-code *SE80*.



**ABAP DEVELOPMENT
WORKBENCH**

C) Download ABAP Code from GitHub

C2. Update ABAP code from GitHub(Only for TSO)



STEP 1: Delete the user's Account Repository

1-1: Assume you've already installed the sample code of TSO to your local SAP system with the version of the previous release, in the latest release, there will be some code changes in public sample code, you need to update the local code according to the latest public sample code.

1-2: Navigate to the user's account's repository, click "settings"

The screenshot shows a GitHub repository page for the repository `dev97619 / logistics-business-network-gtt-samples`. The 'Settings' tab is highlighted with a red box. The page displays the following information:

- Code**: master branch, 2 branches, 0 tags.
- Branch Status**: This branch is 2 commits ahead of SAP-samples:master.
- Commits** (Listed from newest to oldest):
 - dev97619 Update .abapgit.xml (8b46800, 29 minutes ago, 173 commits)
 - .reuse (update license with REUSE, 5 months ago)
 - LICENSES (update license with REUSE, 5 months ago)
 - 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, 29 minutes ago)
 - FAQs_for_Template_Code_Implement... (Update for February release, 12 days ago)
 - README.md (Update README.md, 10 days ago)
- About**: This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.
- Readme**: Readme link.
- Releases**: No releases published. Create a new release link.
- Packages**: No packages published. Publish your first package link.

STEP 1: Delete the user's Account Repository

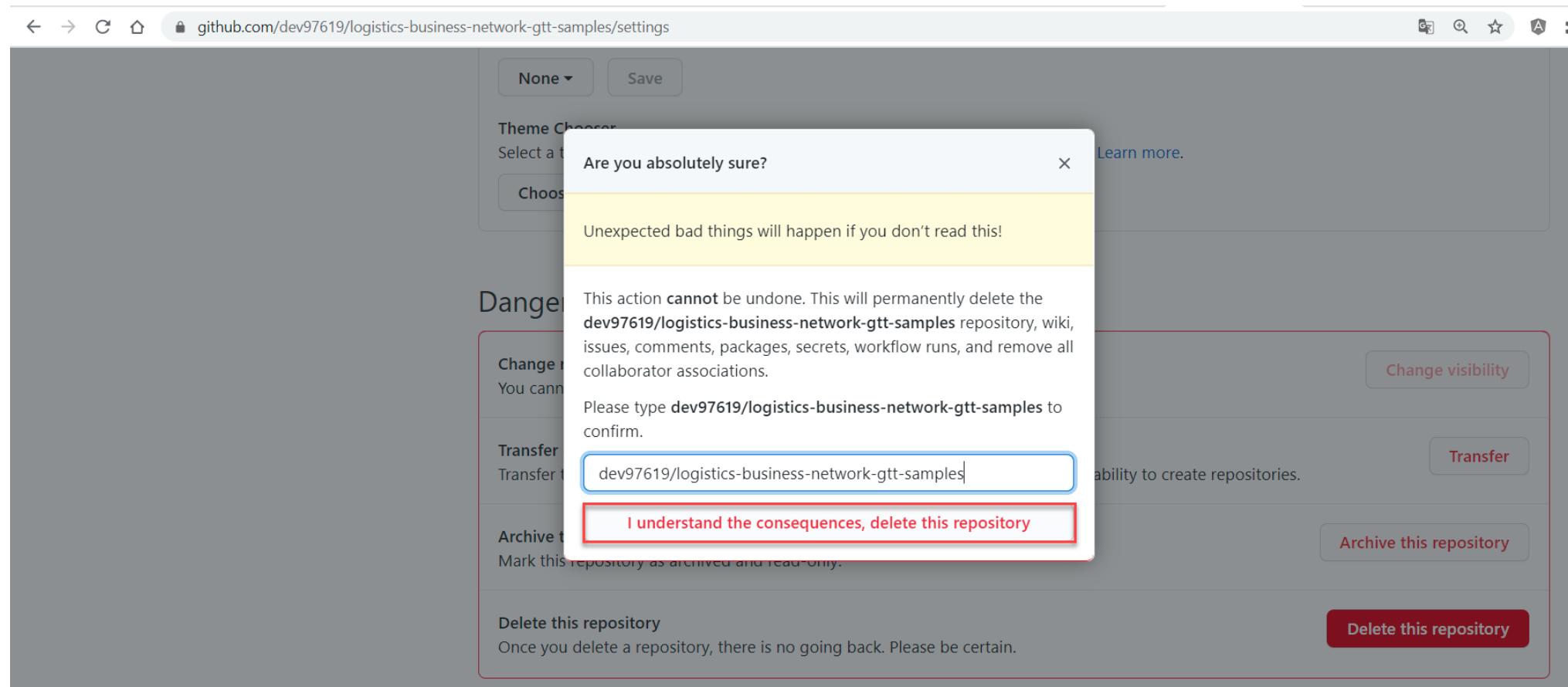
1-3: Scrolling down and find the button “Delete this repository” and click it.

The screenshot shows a GitHub repository settings page for 'github.com/dev97619/logistics-business-network-gtt-samples'. At the top, there is a 'Theme Chooser' section with a dropdown set to 'None' 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](#). A red-bordered 'Change visibility' button is present.
- Transfer ownership**: Transfer this repository to another user or to an organization where you have the ability to create repositories. A red-bordered 'Transfer' button is present.
- Archive this repository**: Mark this repository as archived and read-only. A red-bordered 'Archive this repository' button is present.
- Delete this repository**: Once you delete a repository, there is no going back. Please be certain. This button is also red-bordered and highlighted with a red box.

STEP 1: Delete the user's Account Repository

1-4: The popup shows some warning messages and follow the step it mentioned, click the button “I understand the consequences, delete this repository”



STEP 1: Delete the user's Account Repository

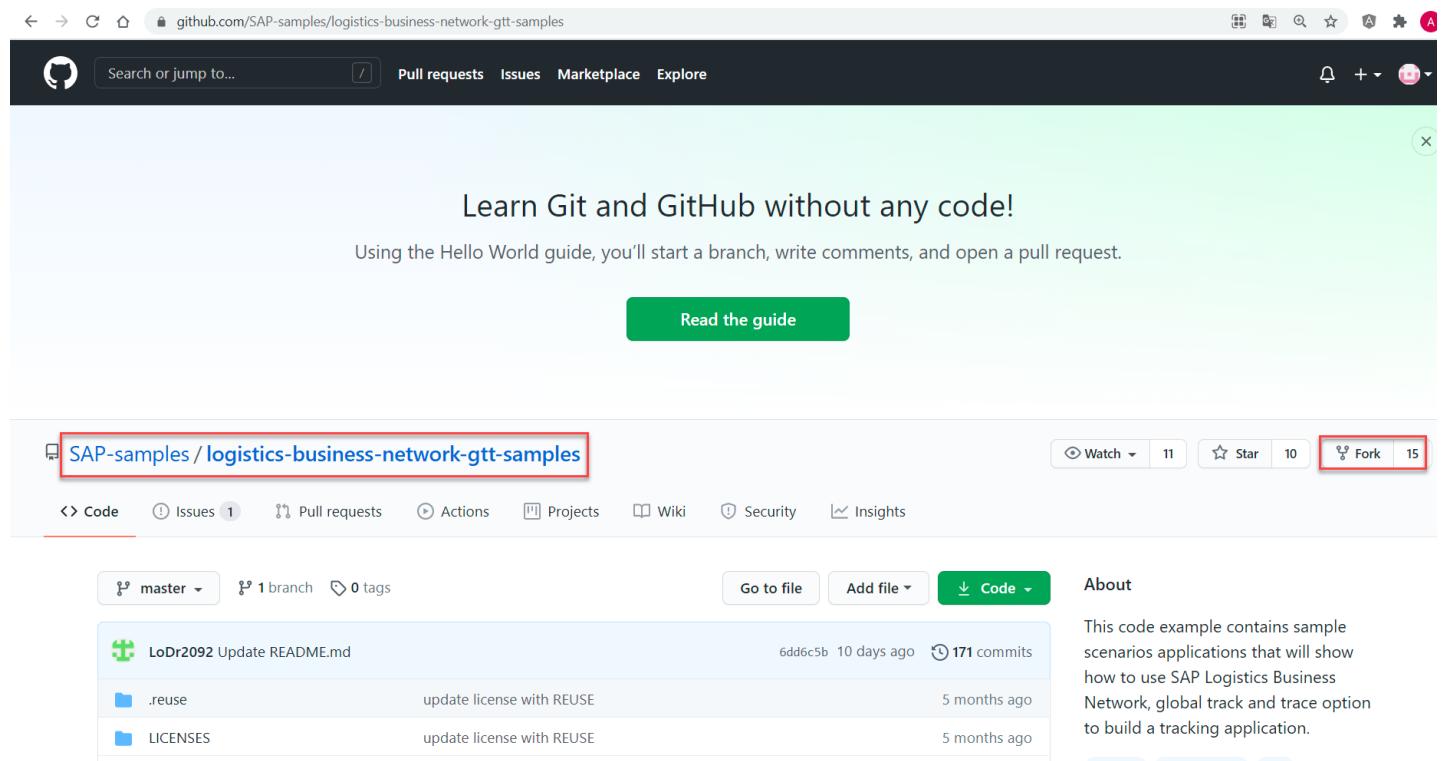
1-5: The user account's repository is deleted.

The screenshot shows a dark-themed GitHub interface. At the top, there is a navigation bar with a search bar, a pull requests tab, issues, marketplace, and explore links. On the right side of the header are notifications, a plus sign for creating new items, and a user profile icon. Below the header, a message box contains the text "Your repository \"dev97619/logistics-business-network-gtt-samples\" was successfully deleted." A red rectangular box highlights this message. The main content area features a "Create your first project" section with a "Create repository" button (which is green) and an "Import repository" link. To the right, a large modal window titled "Learn Git and GitHub without any code!" is displayed. It contains the text "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below this text are two buttons: a green "Read the guide" button and a white "Start a project" button. The entire interface has a clean, modern design with a light blue background for the main content area.

STEP 2: Fork Sample code Repository

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

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



STEP 2: Fork Sample code Repository

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

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with SAP-samples:master.

Go to file Add file Code

Pull request Compare

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

About

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

Readme

Releases

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

Packages

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

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

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

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. A red box highlights the repository name 'dev97619 / logistics-business-network-gtt-samples' in the header and the '.abapgit.xml' file in the commit list.

This branch is even with SAP-samples:master.

File	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 .abapgitxml	13 days ago
FAQs_for_Template_Code_Implement...	Update for February release	12 days ago
README.md	Update README.md	10 days ago

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

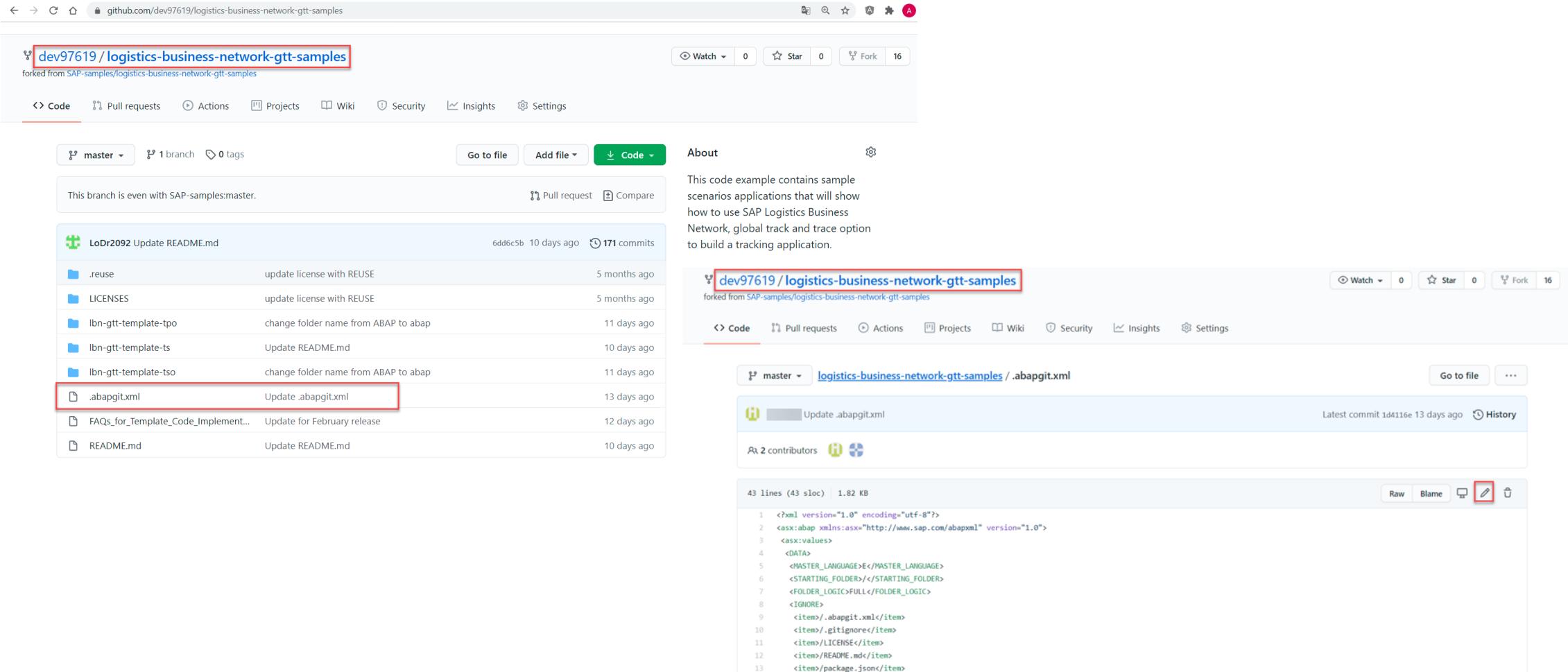
Readme

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

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

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

3-2: Click  button to edit the file



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

Top Repository Page:

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

Bottom File Page:

- File: .abapgit.xml
- Commit: Update .abapgit.xml (latest commit 1d4116e 13 days ago)
- Contributors: 2 contributors
- Code View:

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

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

3-3: Add the sentence of ‘<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>’ like 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 right, a 'Commit changes' dialog is open over a code editor. The code editor shows the file '.abapgit.xml' with the following content:

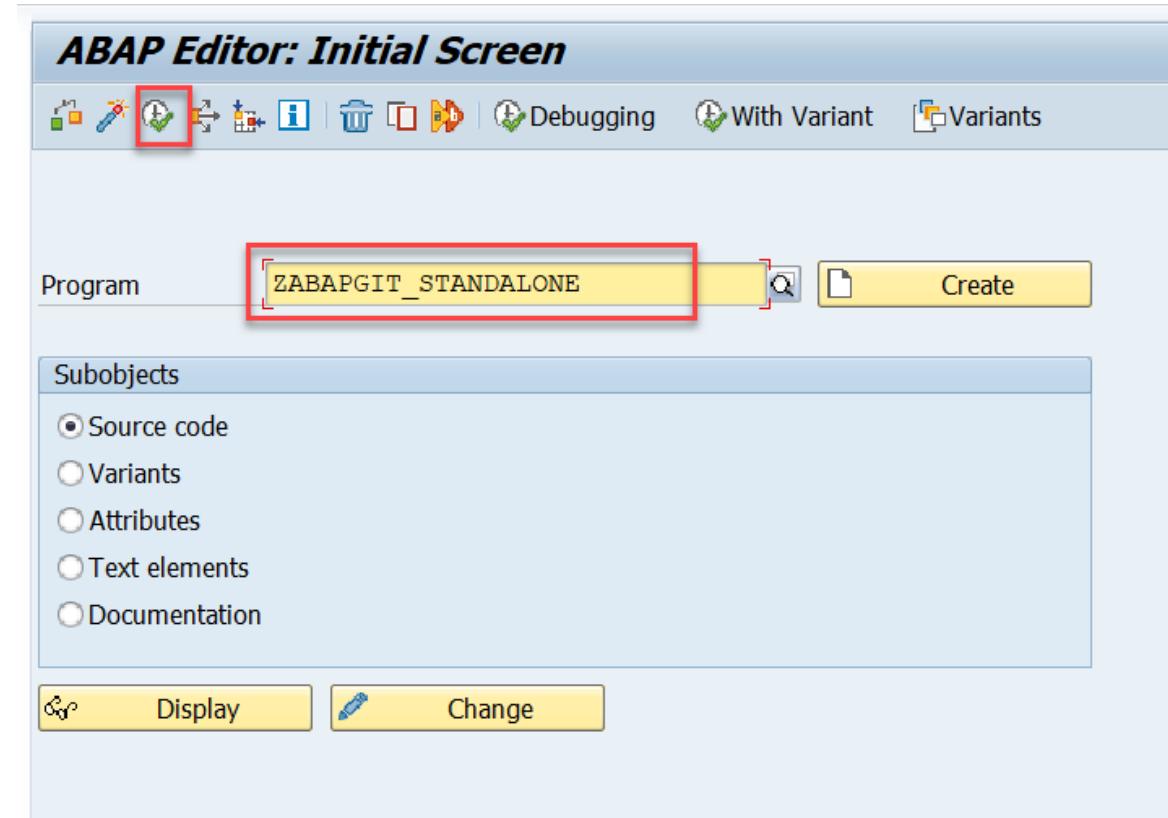
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>
7       <FOLDER_LOGIC>FULL</FOLDER_LOGIC>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

The line '6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>' is highlighted with a red box. In the commit dialog, the message 'Update .abapgit.xml' is entered. The 'Commit directly to the master branch' radio button is selected. The 'Commit changes' button is highlighted with a red box.

STEP 4: Update ABAP code from GitHub

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

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



STEP 4: Update ABAP code from GitHub

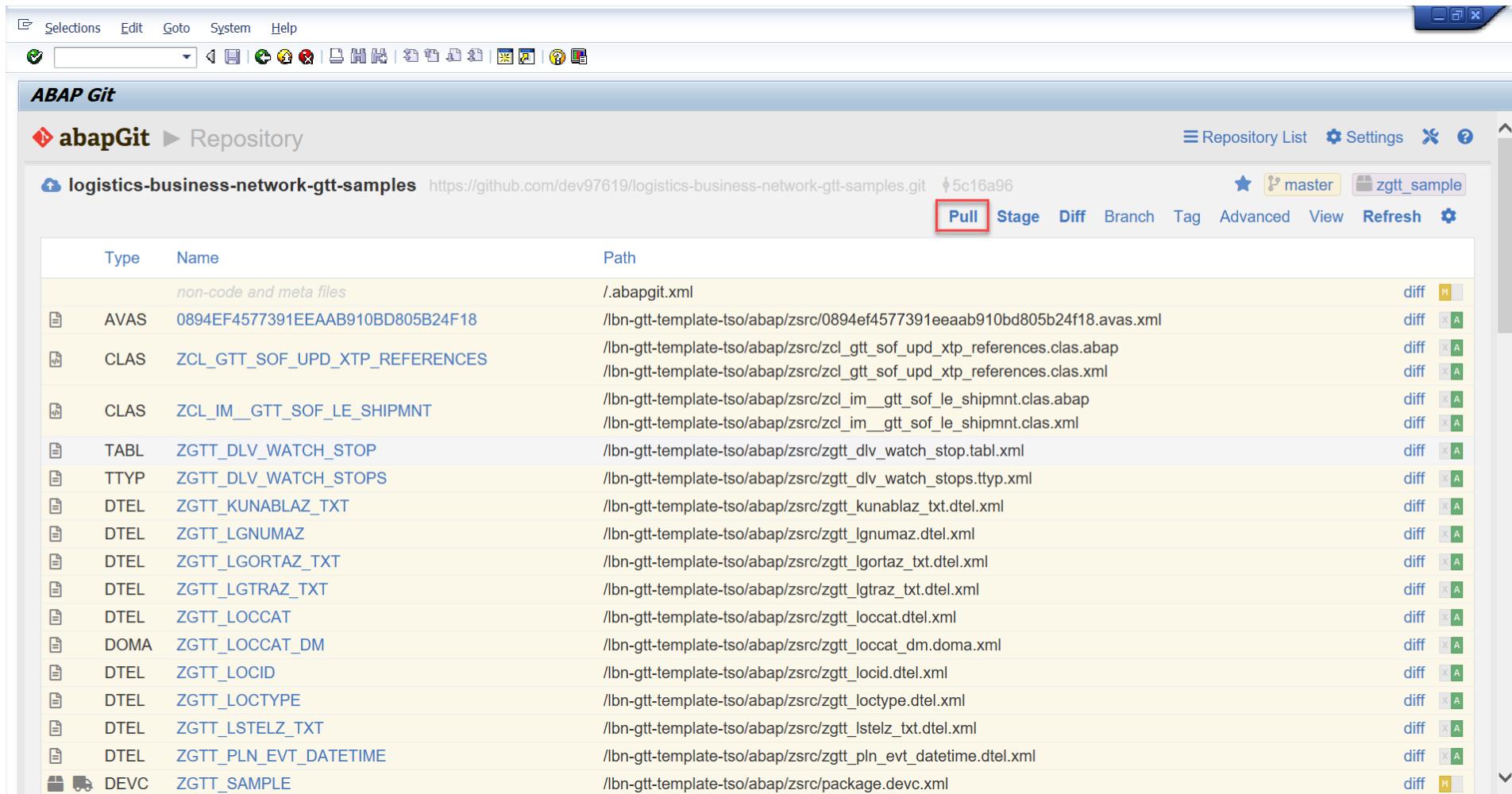
4-3: Check the URL is not changed after your recreation of repository copy. Access the TSO Repository by clicking button.

The screenshot shows the ABAP Git interface within a SAP application window. The title bar includes standard SAP menu items: Selections, Edit, Goto, System, Help, along with various toolbar icons. The main area is titled "ABAP Git" and displays a "Repository List". A header bar contains the "abapGit" logo, a "Repository List" link, and several action buttons: "New Online", "New Offline", "Settings", and a question mark icon. Below this is a filter bar with a "Filter:" input field, a "Only Favorites" checkbox, and a "Detail" checkbox. The main table lists repositories with columns: Name, Url, Package, Branch, and Action. One repository is listed: "logistics-business-network-gtt-samples" with Url "github.com/dev97619/logistics-business-network-gtt-samples.git", Package "zgtt_sample", Branch "master", and Action buttons for "Check", "Stage", "Patch", and "Settings". The "Settings" button has a red box drawn around it. At the bottom of the interface is a footer with the "abapGit" logo and version "1.106.0", and the text "js: OK".

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

STEP 4: Update ABAP code from GitHub

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



The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it, the title bar says "ABAP Git". Underneath, it shows the repository "abapGit > Repository" and the specific repository "logistics-business-network-gtt-samples" with the URL "https://github.com/dev97619/logistics-business-network-gtt-samples.git". The commit hash "5c16a96" is also displayed. A "master" branch is selected. The main area is a table with columns "Type", "Name", and "Path". The "Pull" button in the header is highlighted with a red box. The table lists several files and their paths, such as "AVAS", "ZCL_GTT_SOUPD_XTP_REFERENCES", "ZCL_IM_GTT_SOUPD_LE_SHIPMNT", etc.

Type	Name	Path	diff	M
	non-code and meta files	/abapgit.xml	diff	M
AVAS	0894EF4577391EEAAB910BD805B24F18	//lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	diff	X A
CLAS	ZCL_GTT_SOUPD_XTP_REFERENCES	//lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	diff	X A
CLAS	ZCL_IM_GTT_SOUPD_LE_SHIPMNT	//lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	diff	X A
TABL	ZGTT_DLV_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stop.tabl.xml	diff	X A
TTYP	ZGTT_DLV_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zggt_dlv_watch_stops.ttyp.xml	diff	X A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_kunablaTxt.dtel.xml	diff	X A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml	diff	X A
DTEL	ZGTT_LGORAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgorazTxt.dtel.xml	diff	X A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lgtrazTxt.dtel.xml	diff	X A
DTEL	ZGTT_LOCCAT	//lbn-gtt-template-tso/abap/zsrc/zggt_locat.dtel.xml	diff	X A
DOMA	ZGTT_LOCCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zggt_locat_dm.doma.xml	diff	X A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	diff	X A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	diff	X A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zggt_lstelzTxt.dtel.xml	diff	X A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml	diff	X A
DEV	ZGTT_SAMPLE	//lbn-gtt-template-tso/abap/zsrc/package.devcl.xml	diff	M

C) Download ABAP Code from GitHub

C3. Download Another ABAP code from GitHub(TPO)

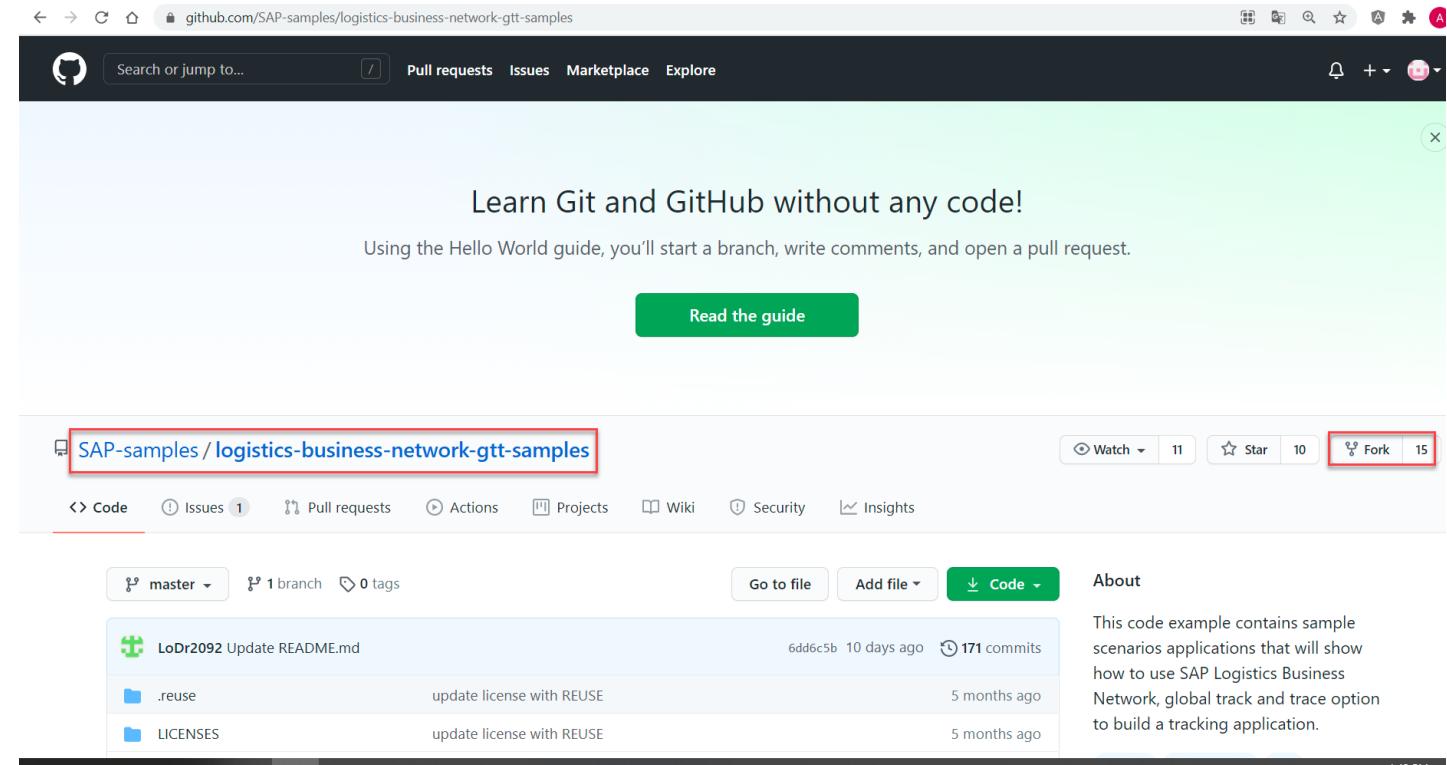


STEP 1: Fork Sample code Repository

1-1. Assume that using ABAPGit, you have already installed the sample code of TSO to your local SAP system and meanwhile you also want to install sample code of TPO to your local SAP system.

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

1-3. Click the “Fork” button, it will copy the newest version of sample codes into the user’s account and meanwhile it will navigate to user’s own repository



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

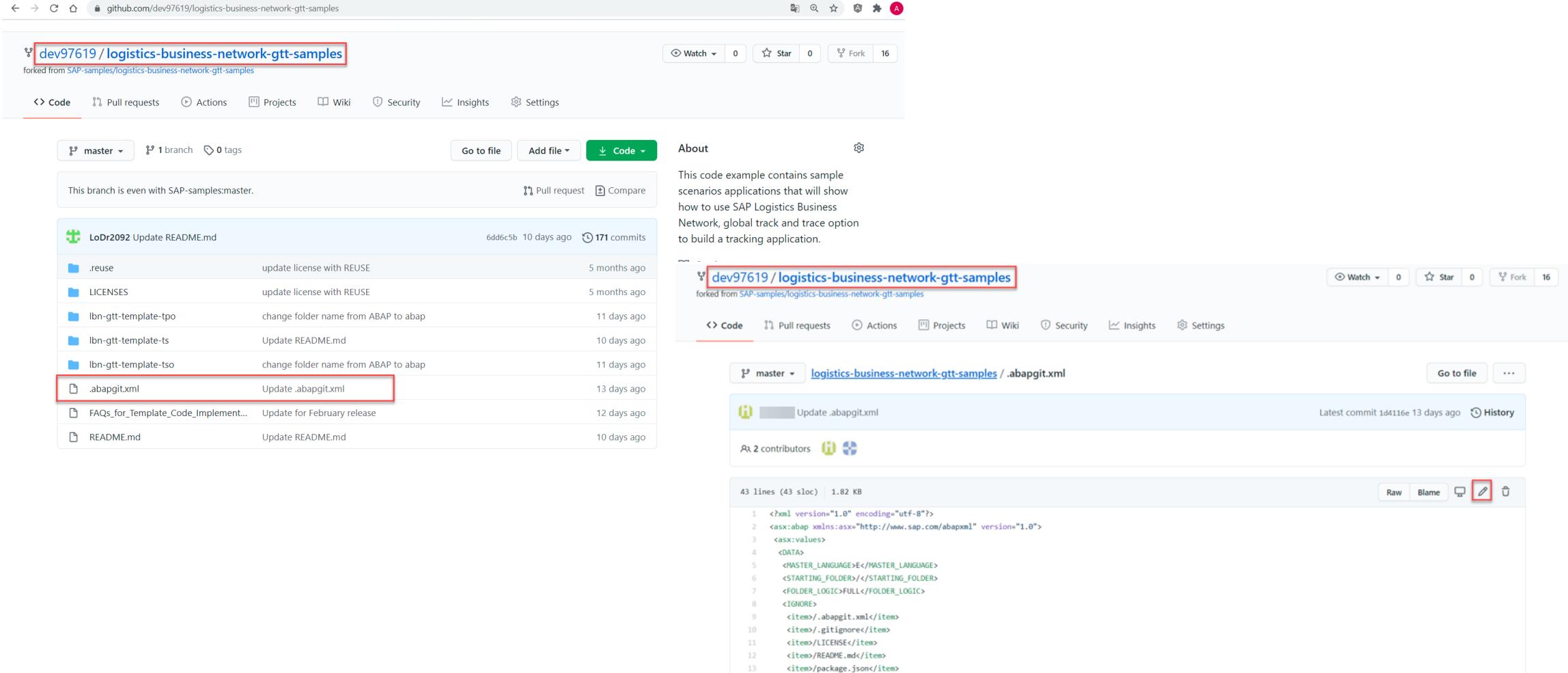
2-1: In the user’s account repository, click the file ‘.abapgit.xml’

The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The 'master' branch is active, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including one for '.abapgit.xml' which is highlighted with a red box. The commit message is 'Update .abapgitxml'. Other commits include updates to README.md, LICENSES, and folder names. The repository has 0 stars, 0 forks, and 16 issues. The 'About' section describes the code example as containing sample scenarios for SAP Logistics Business Network, global track and trace options. It includes links to 'Readme', 'Releases', and 'Packages'.

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

STEP 2: Change Configuration file '.abapgit.xml'

2-2: Click  button to edit the file



The screenshot shows two GitHub repository pages. The top page is for the forked repository `dev97619/logistics-business-network-gtt-samples`. The bottom page is for the original repository `SAP-samples/logistics-business-network-gtt-samples`. Both pages have their respective names highlighted with red boxes.

Top Repository (Forked):

- Code Tab:** Shows 1 branch and 0 tags. A commit by LoDr2092 titled "Update README.md" is listed, along with several commits related to license updates and folder renames.
- Bottom Repository (Original):** Shows 171 commits. A specific commit titled ".abapgit.xml" is highlighted with a red box.

Bottom Repository (Original) - .abapgit.xml File View:

- File Details:** The file has 43 lines (43 sloc) and is 1.82 KB.
- Actions:** Buttons for "Raw", "Blame", "Edit" (with a red box), and "Delete".
- Content:** The XML code defines a configuration for ABAP Git. It includes sections for master language, starting folder, folder logic, and ignore patterns.

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

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

2-3: Add the sentence of ‘<STARTING_FOLDER> /lbn-gtt-template-tpo/abap/zsrc/ </STARTING_FOLDER>’ like 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 code editor on the left displays the XML configuration, with the line containing the new folder path highlighted and framed in red. The commit message field contains 'Update .abapgit.xml'. The commit options section includes a radio button for committing directly to the 'master' branch. The 'Commit changes' button at the bottom 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-tpo/abap/zsrc/</STARTING_FOLDER>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
    <IGNORE>
      <item>/.abapgit.xml</item>
      <item>/.gitignore</item>
    </IGNORE>
  </asx:values>
</asx:abap>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

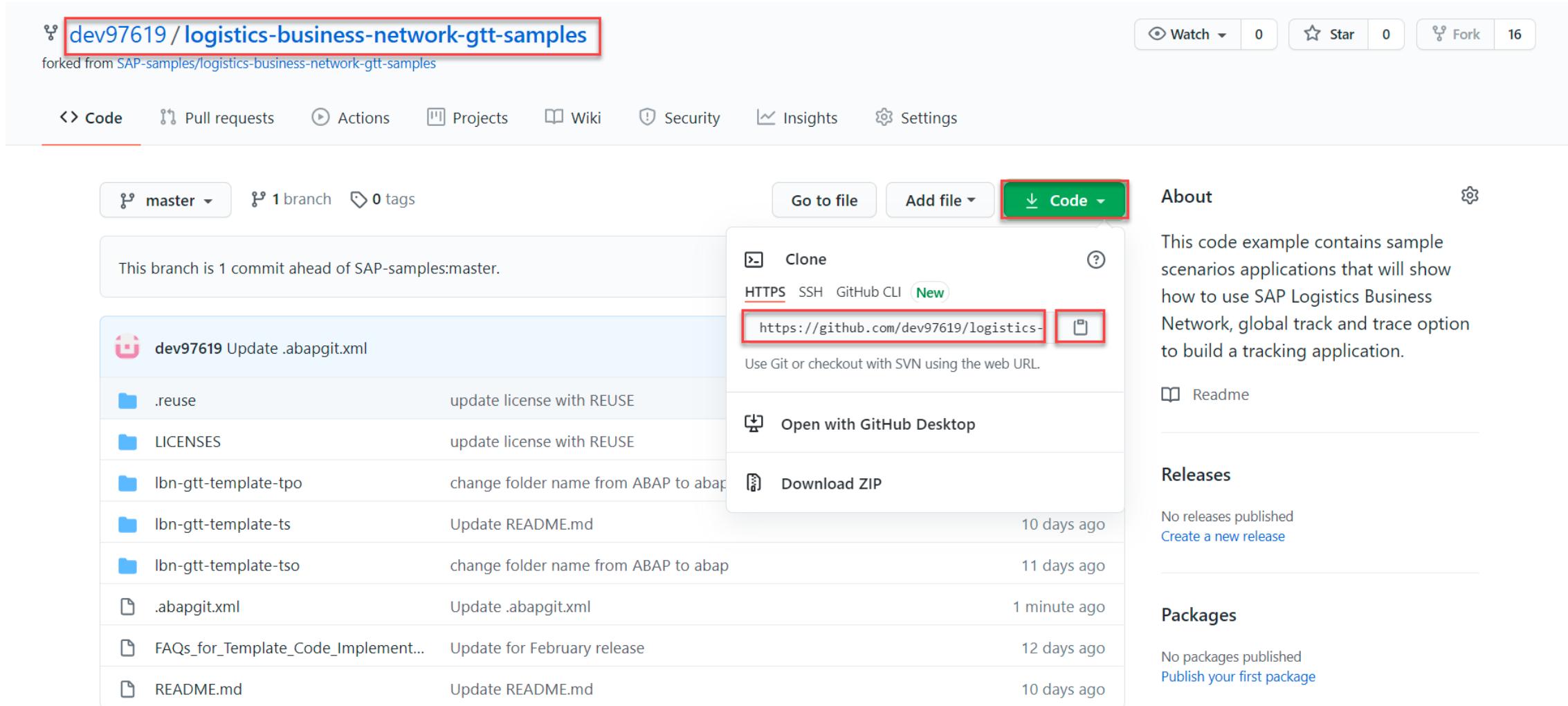
-o- Commit directly to the master branch.

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

Commit changes Cancel

STEP 2: Change Configuration file '.abapgit.xml'

2-5: Go to the root and copy the repository URL by clicking  button



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository has 0 stars and 16 forks. The 'Code' tab is selected. The repository has 1 branch and 0 tags. A message indicates it is 1 commit ahead of SAP-samples:master. A list of commits shows updates to '.abapgit.xml' and other files. On the right, there's a 'Code' dropdown menu with options like 'Clone' (highlighted with a red box) and 'Download ZIP'. The 'Clone' option shows the URL <https://github.com/dev97619/logistics-business-network-gtt-samples>, which is also highlighted with a red box. The 'About' section describes the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options. Other sections include 'Readme', 'Releases' (no releases), and 'Packages' (no packages).

dev97619 / logistics-business-network-gtt-samples

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

Ibn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

FAQs_for_Template_Code_Implement... Update for February release

README.md Update README.md

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

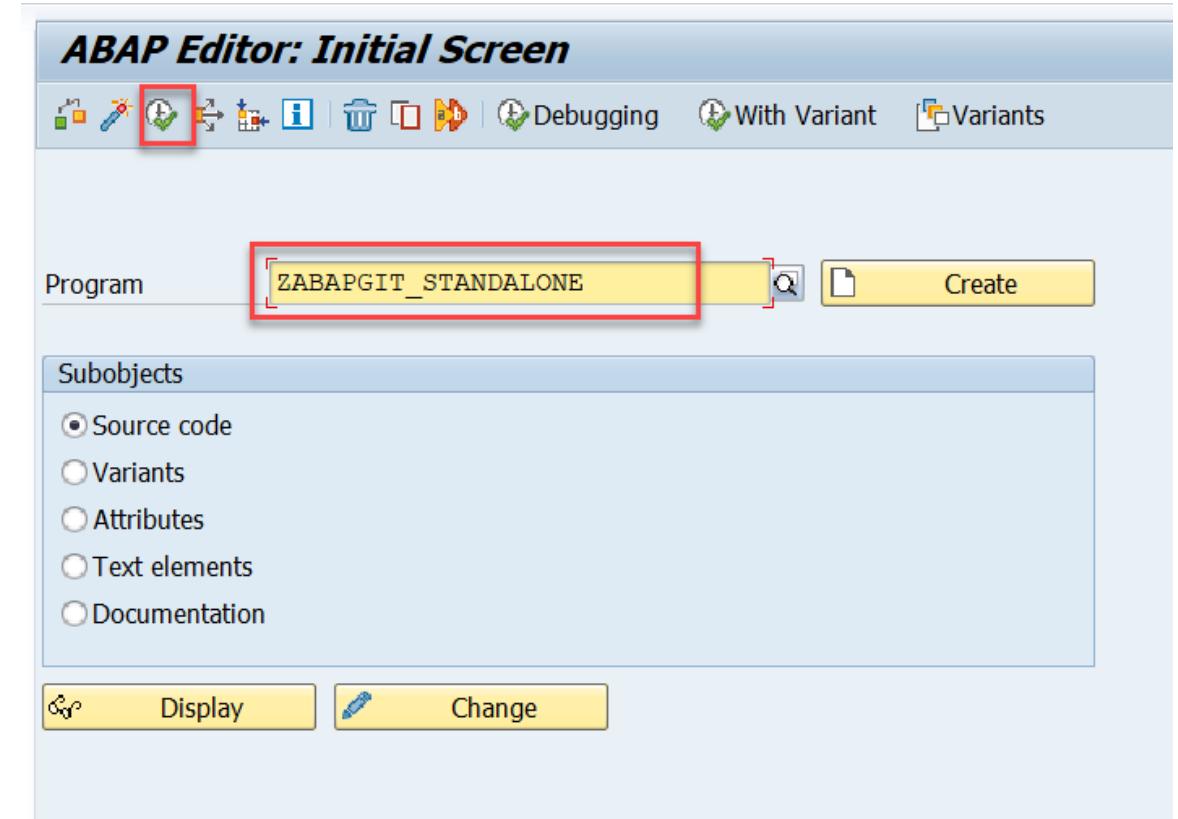
No packages published

Publish your first package

STEP 3: Remove TSO Repository in ABAPGit

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

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



STEP 3: Remove TSO Repository in ABAPGit

3-3: Access the TSO Repository by clicking  button



The screenshot shows the ABAPGit interface with the following details:

- Toolbar:** Selections, Edit, Goto, System, Help.
- Repository List:** The title bar says "ABAP Git" and "abapGit ► Repository List".
- Filter:** Filter field, Only Favorites, Detail.
- Table Headers:** Name, Url, Package, Branch, Action.
- Table Data:** One row for "logistics-business-network-gtt-samples".

Name	Url	Package	Branch	Action
logistics-business-network-gtt-samples	github.com/dev97619/logistics-business-network-gtt-samples.git	zgtt_sample	master	Check Stage Patch Settings 
- Footer:** abapGit 1.106.0, js: OK.

STEP 3: Remove TSO Repository in ABAPGit

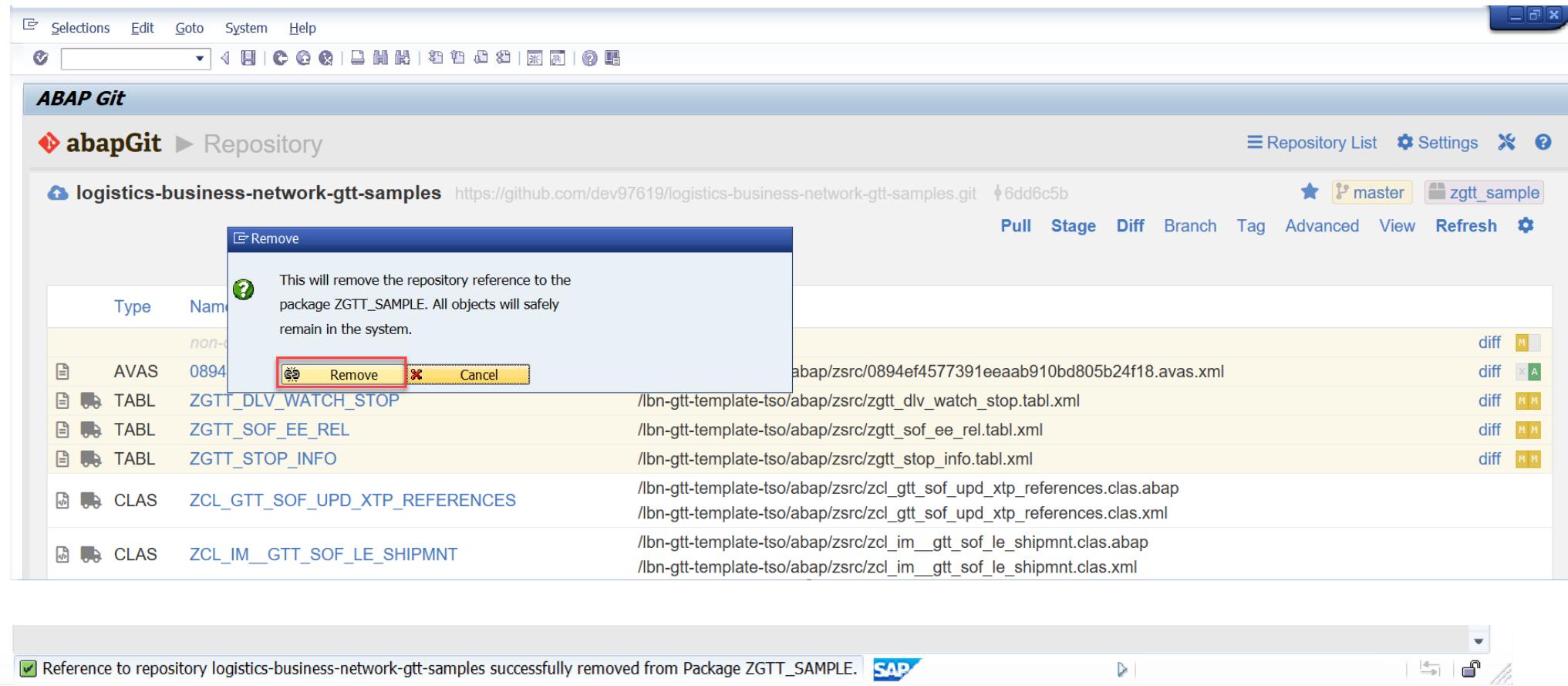
3-4: Choose sub menu “Remove” under the “Advanced” menu, click it.

The screenshot shows the ABAPGit interface within SAP. The top navigation bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the toolbar, the title bar says 'ABAP Git' and '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', and 'Path'. The 'Advanced' menu is open, and its sub-menu 'Remove' is highlighted with a red box. Other options in the Advanced menu include 'Reset Local (Force Pull)', 'Checkout commit', 'Background Mode', 'Change Remote', 'Make Off-line', 'Force Stage', 'Transport to Branch', 'Add all objects to transport request', 'Syntax Check', 'Run Code Inspector', 'Update Local Checksums', 'Beta - Data', and 'Uninstall'.

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

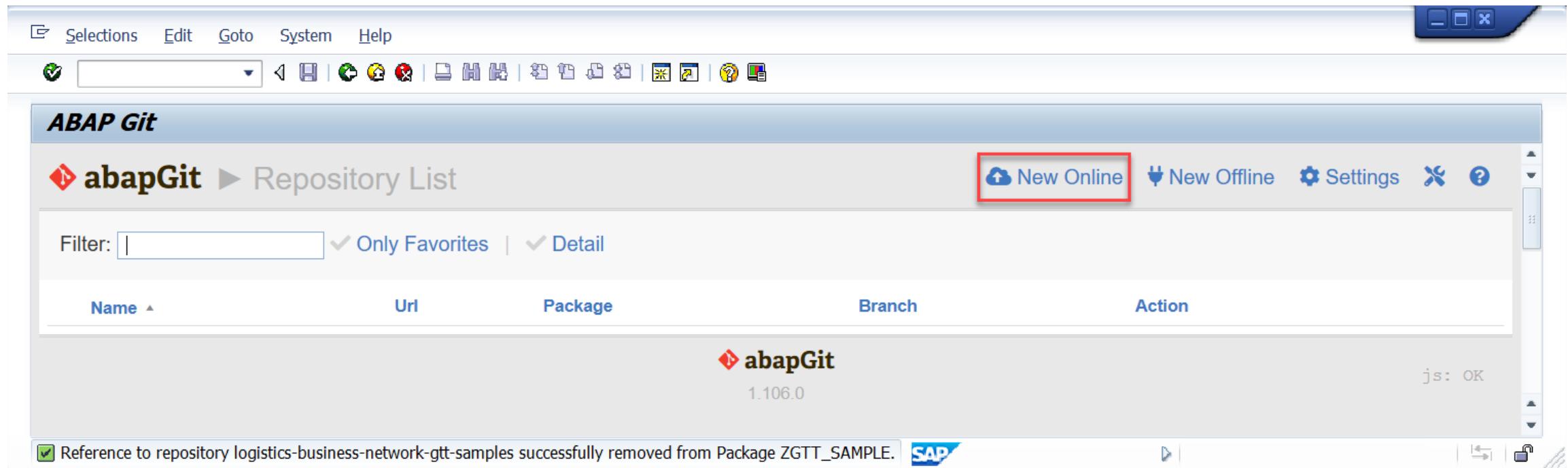
STEP 3: Remove TSO Repository in ABAPGit

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



STEP 4: Download TPO code from GitHub

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



STEP 4: Download TPO code from GitHub

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

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

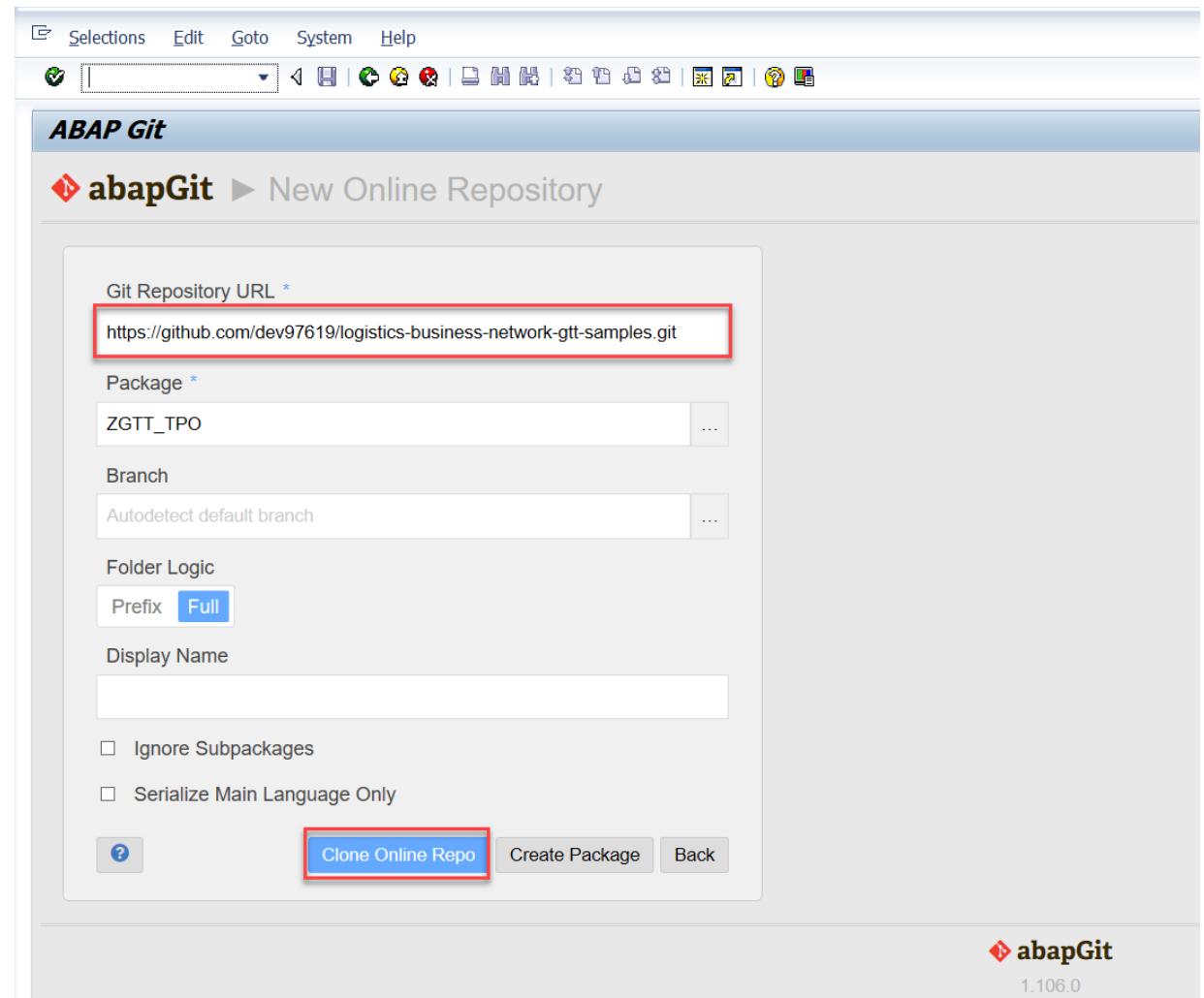
Caution:

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

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

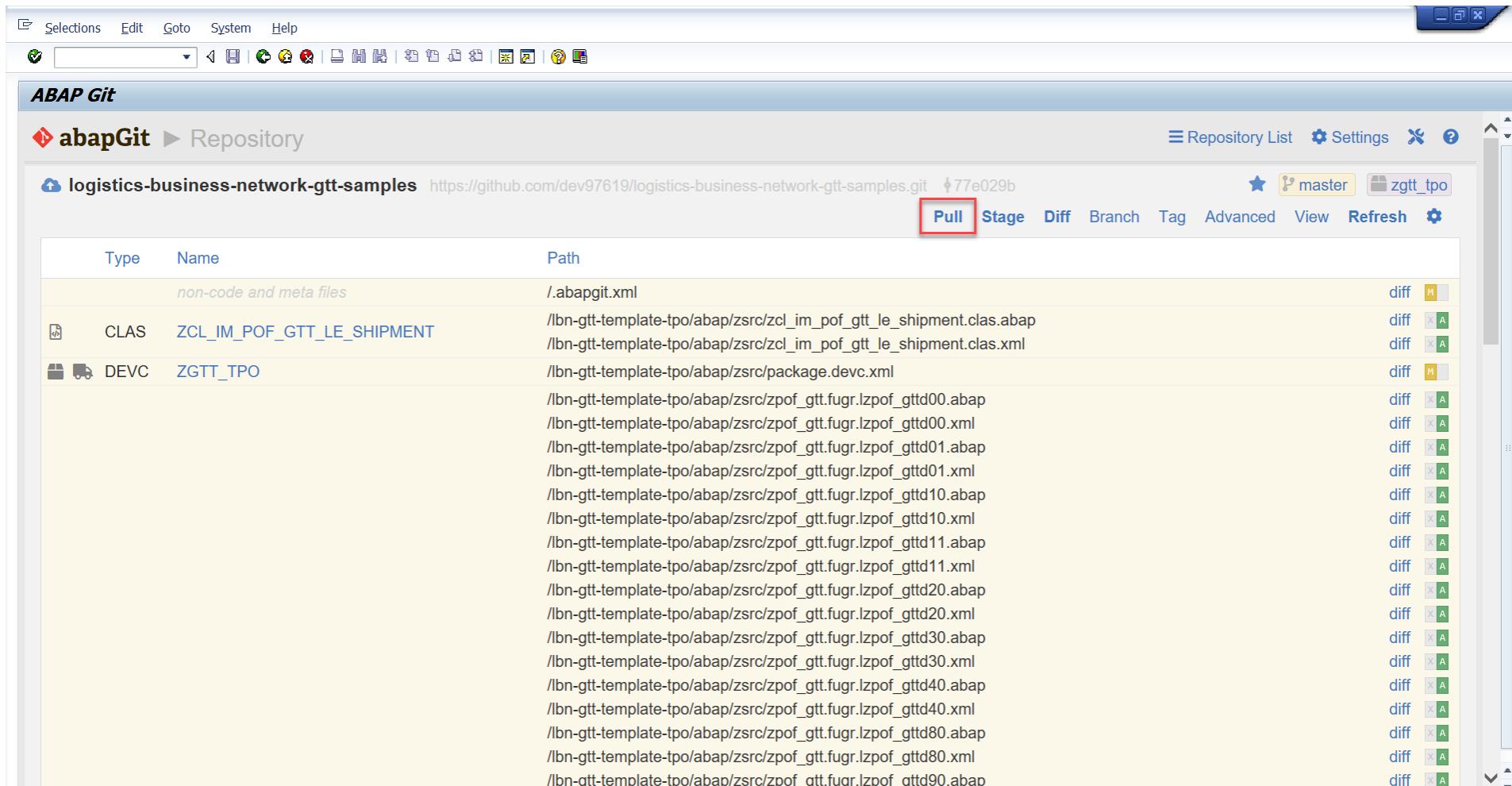
4-4: Set **Full** for **Folder Logic**

4-5: Click **Clone online repo** to download the code



STEP 4: Download ABAP code from GitHub

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



The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation "abapGit > Repository". The main area displays a list of files under the repository "logistics-business-network-gtt-samples" with the URL "https://github.com/dev97619/logistics-business-network-gtt-samples.git". The commit ID "77e029b" is shown. On the right side of the header, there are buttons for "Repository List", "Settings", and "Refresh". Below these buttons, the "Pull" button is highlighted with a red box. The file list table has columns for "Type", "Name", and "Path". The "Path" column contains numerous ABAP and XML files related to GTT templates. To the right of each file entry, there are "diff" buttons and small colored status indicators (yellow, green, or grey).

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

C) Download ABAP Code from GitHub

C4. Initial Download ABAP code from GitHub(include TSO/TPO/TS)



STEP 1: Install ABAPGit

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

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

[Improve this page](#)

Summary #

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

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

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

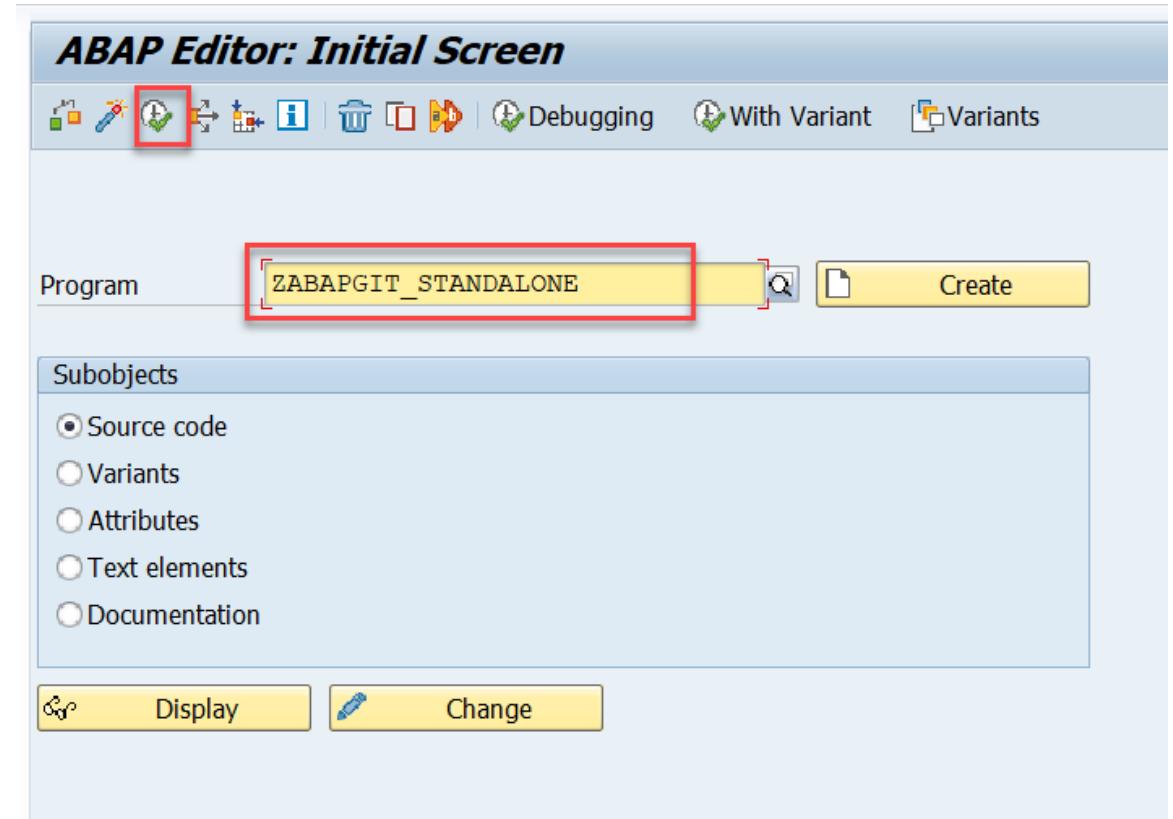
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `.$ZABAPGIT`).

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

STEP 2: Download ABAP Code

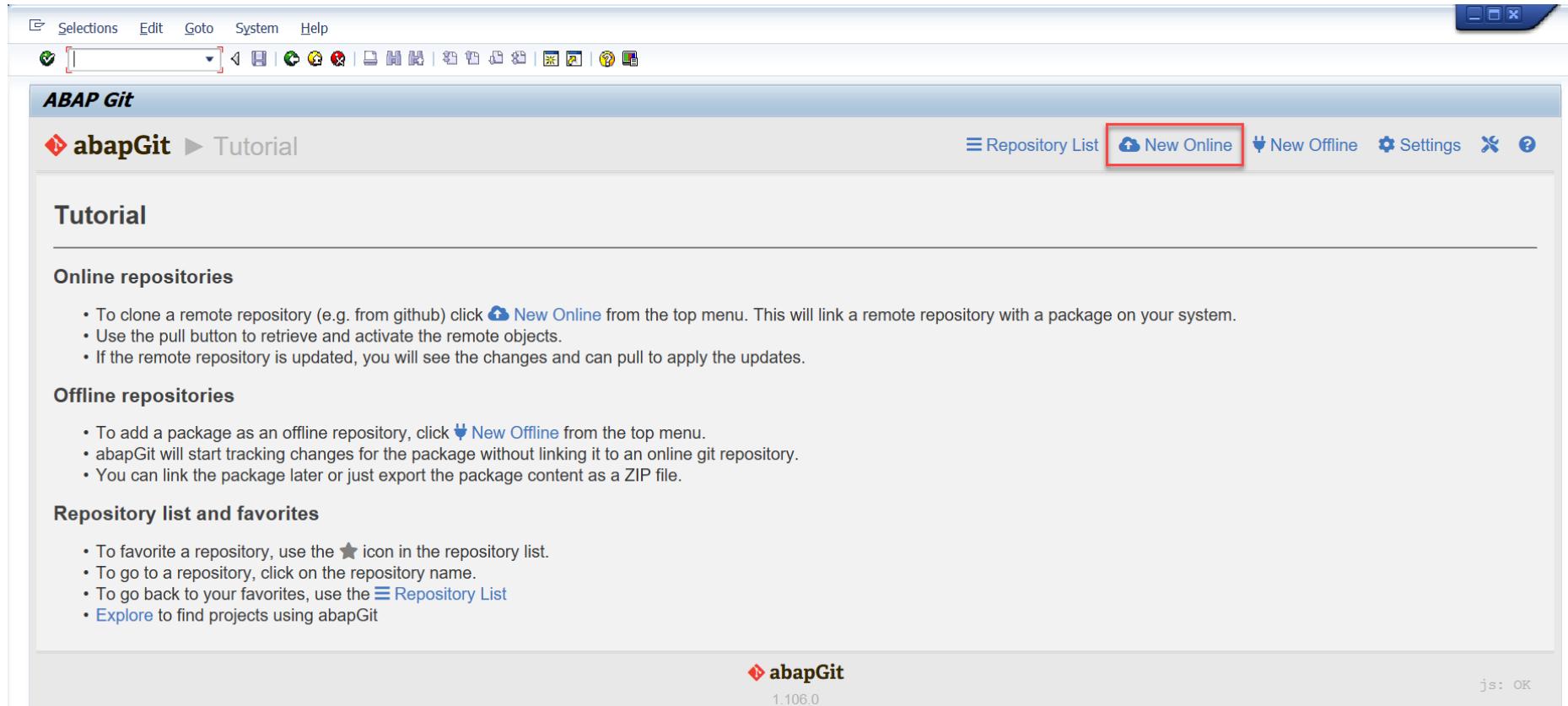
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE

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



STEP 2: Download ABAP Code

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



STEP 2: Download ABAP Code

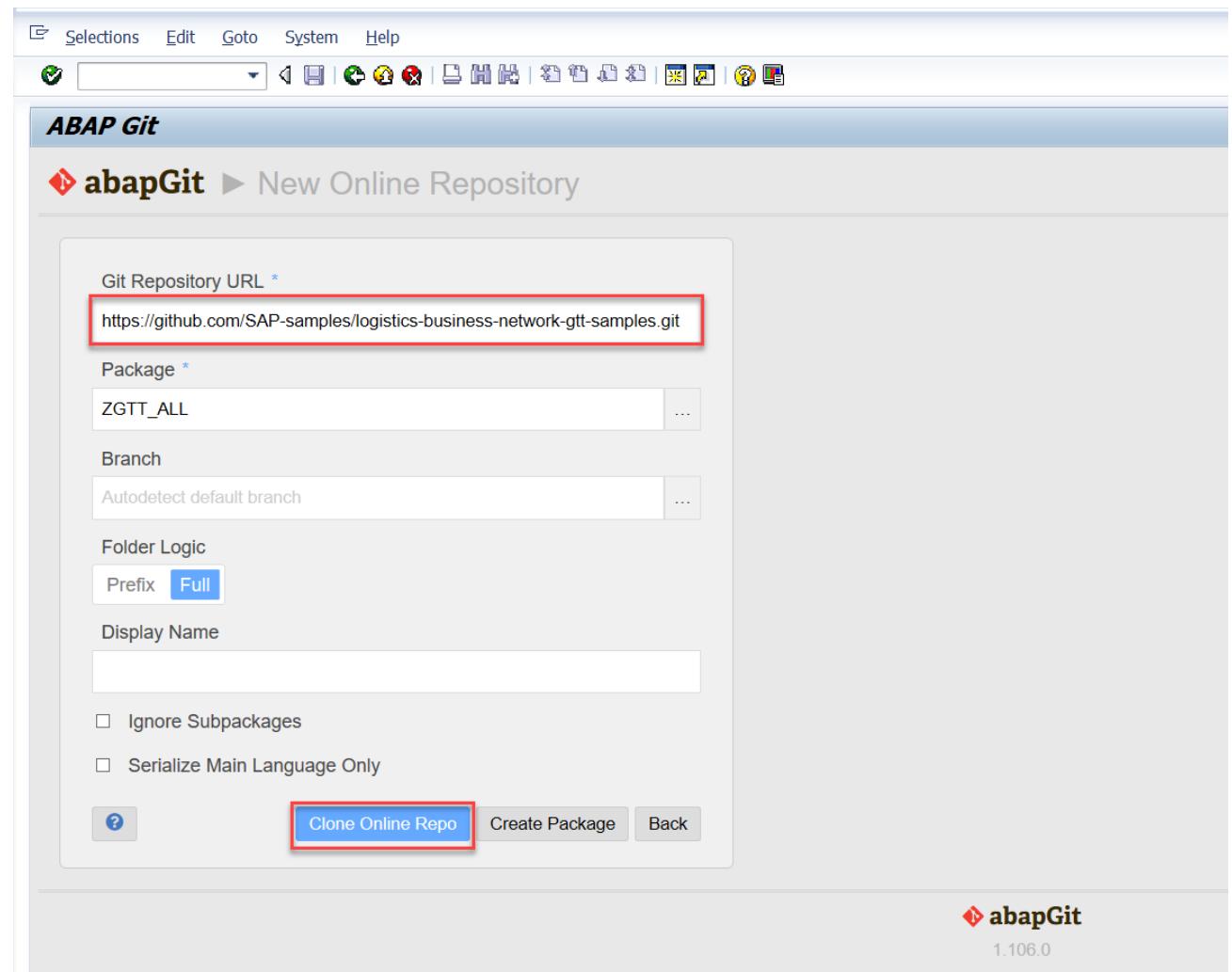
2-4: Fill in the **Git repository URL**:

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

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

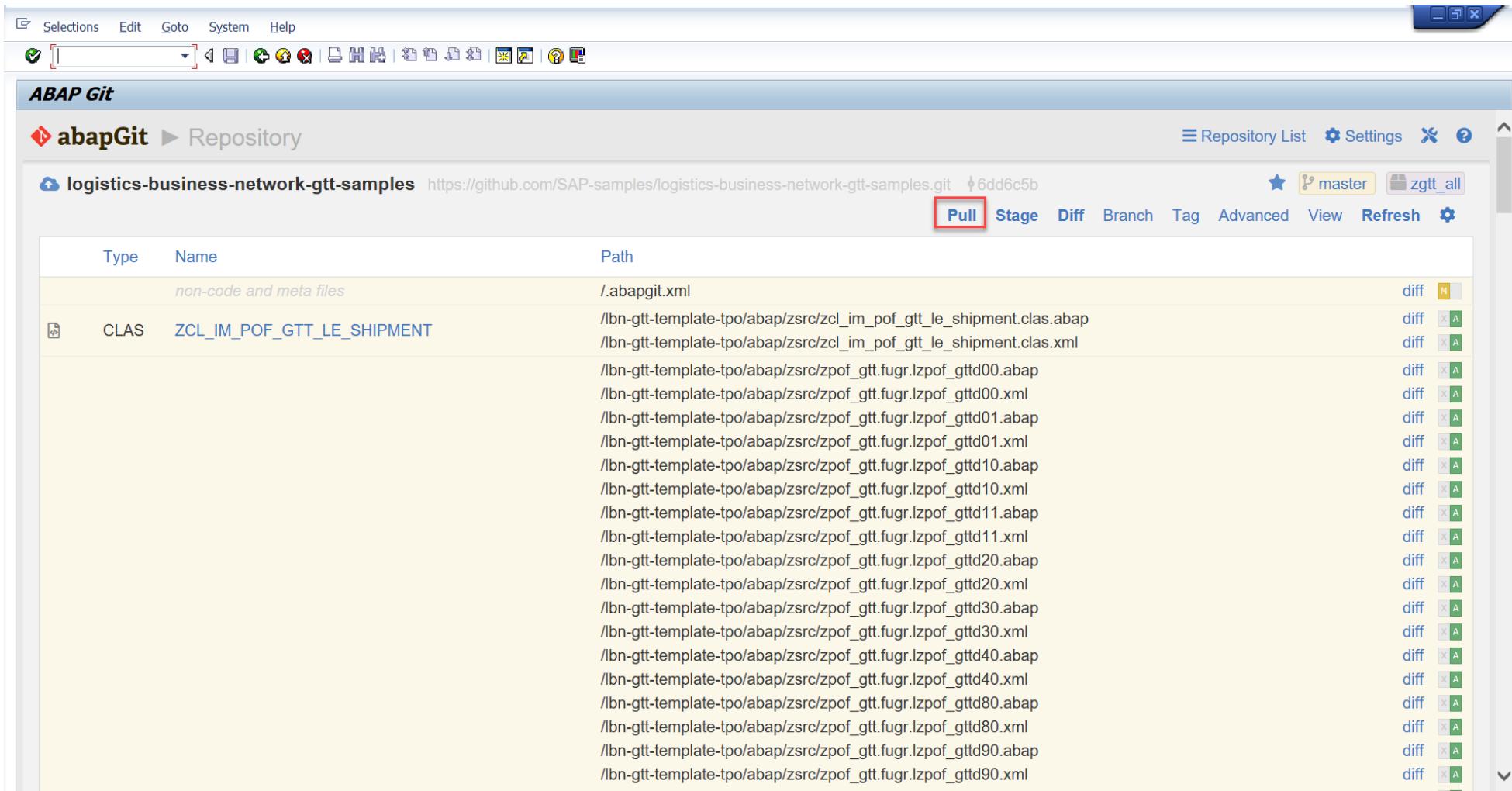
2-6: Set *Full* for **Folder Logic**

2-7: Click **Clone online repo** to download the code



STEP 2: Download ABAP Code

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

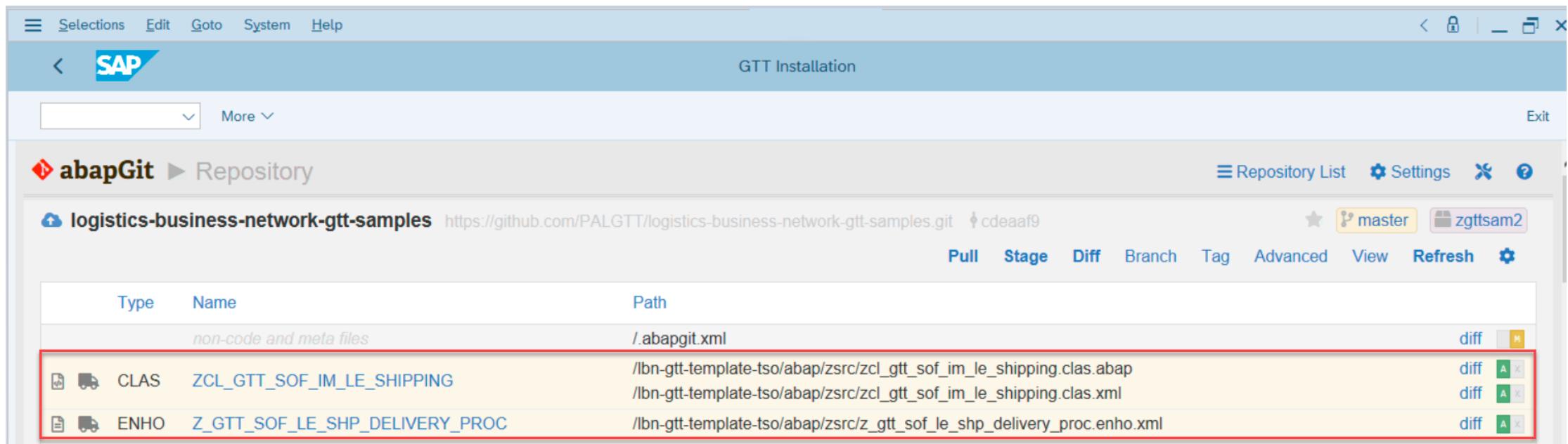


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the path 'abapGit > Repository'. A repository card for 'logistics-business-network-gtt-samples' is displayed, including its URL (<https://github.com/SAP-samples/logistics-business-network-gtt-samples.git>) and commit hash ('6dd6c5b'). The repository name is 'logistics-business-network-gtt-samples'. The 'master' branch is selected, indicated by a yellow star icon. A red box highlights the 'Pull' button in the top navigation bar. The main content area is a table with columns 'Type', 'Name', and 'Path'. The table lists several files and directories under the 'ZCL_IM_POF_GTT_LE_SHIPMENT' class. The 'Path' column shows paths like '/.abapgit.xml', '/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap', and '/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml'. The 'Type' column shows 'non-code and meta files' for the first item and 'CLAS' for the class definition. The 'Name' column lists the file names. To the right of each row are 'diff' buttons and small colored status indicators (yellow, green, blue, grey).

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

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

Symptom: If the user update the ABAP code by report ZABAPGIT_STANDALONE, there will be a code difference as below: because the enhancement implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC is already obsolete and removed from the GitHub, the report ZABAPGIT_STANDALONE cannot remove the object which was already deleted in GitHub.



The screenshot shows the SAP GTT Installation interface. The title bar says "GTT Installation". The main area is titled "abapGit" and "Repository". It displays a list of files from the repository "logistics-business-network-gtt-samples" at commit "cdeaa9". The files listed are:

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] [diff 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]

The last two rows, which correspond to the deleted enhancement implementation, are highlighted with a red border and have a yellow "diff" icon next to them.

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

Solution:

Option 1)

1-2) Deactivate the BADI implementation

Option 2)

2-1) Deactivate the enhancement implementation Z_GTT_SOF_LE_SHIP_DELIVERY_PROC

2-2) Delete the BADI implementation class ZCL_GTT_SOF_IM_LE_SHIPPING

Notes:

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

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

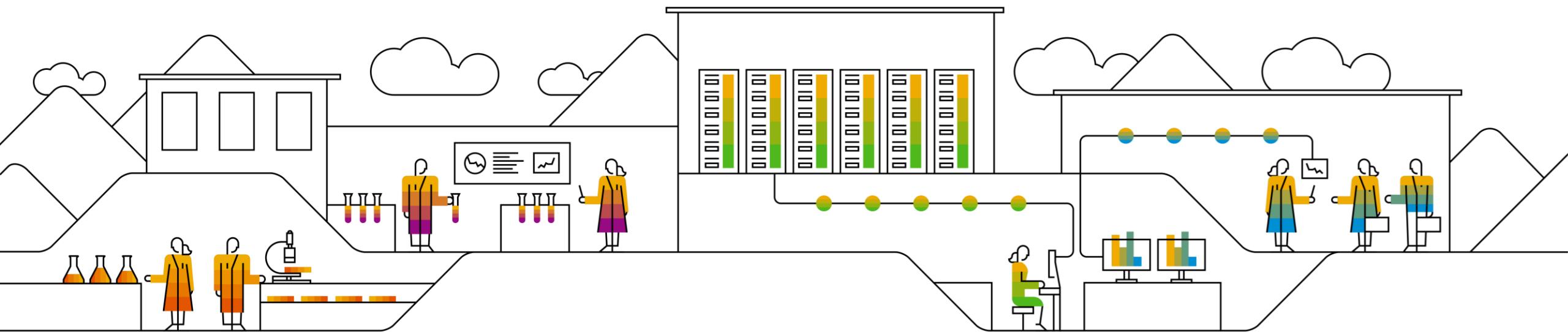
Known Issue: Remote Deleted Object Cannot be Synchronized to the Local Object

For option 1: Go to Transaction code SE19 and deactivate the BADI implementation.

The screenshot shows two SAP application windows. The top window is titled "BAdI Builder: Initial Screen for Implementations". It has a search bar, a toolbar with "Check", "Delete implementation", "Copy implementation", "Rename implementation", "Application help", and "More", and a section for "Edit Implementation" with a radio button for "New BAdI". The "Enhancement implementation" field contains "Z_GTT_SOF_LE_SHP_DELIVERY_PROC", which is highlighted with a red box. The bottom window is titled "Enhancement Implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC Display". It also has a search bar, a toolbar with "Previous Object", "Next Object", "Display <> Change", "Other Object...", "Check", "Activate", "Where-Used List", "Display Object List", and "Fullscreen On/Off", and a "Properties" tab selected. The "Implementation Elements" tab is active, showing a table with one row for "Z_GTT_SOF_IM_LE_SHIPPING". The "Implementation" column shows "Implementation Class" and the "Description" column shows "Implementation: GTT - Enhancement to update the imputed sales orders' delivery list". The "Runtime Behavior" section includes 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 as same as the name that is defined in the corresponding model in the Manage Models application in GTT Version 2.

The screenshot shows the SAP GTT Model Details interface for maintaining Application Object Types. On the left, the 'Define Application Object Types' dialog is open, showing the 'Bus. Proc. Type' as 'TMS_TOR' and the 'Appl. Obj. Type' as 'ZGTT_SHP_ACC_HD'. The 'Appl. Obj. Type' field is highlighted with a red box. The main panel displays the 'Model Details' for the 'sof' model, which is active. The 'IDOC Integration' tab is selected, showing a tracked process mapping for 'Shipment'. The 'Tracked Process / Events (26)' table lists entries like 'ShipmentEvent' and 'LoadingStart'. The 'Fields' table on the right maps IDOC fields to SAP fields, such as 'shipmentNo' to 'E1EHPCT' and 'YN_SHP_NO'.

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCT	YN_SHP_NO
serviceAgentLbnId	E1EHPCT	YN_SHP_SA_LBN_ID
transportationMode	E1EHPCT	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPCT	YN_SHP_CONTAIN_DGOODS

2: Make the Customization Logic in the Function Modules and Assign Them to the Extractor Function

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

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

Function	Function Module	Description
ZSST_GTT_FO_HDR	ZSST_GTT_OTE_FO_HDR_REL	Extractor for relevance determination for Freight Order and Freight Booking

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

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

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

3: Sample Codes for the Track Shipments Template App

To support the Track Shipments template app, the sample codes cover the following cases by function group ZSST_GTT:

Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HDR	Function for control parameters of Freight Order and Freight Booking
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL	Actual Event of Arrival
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_COUPLING	Actual Event of Coupling
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_DECOUPLING	Actual Event of Decoupling
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_DEPARTURE	Actual Event of Departure
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_LOAD_END	Actual Event of Loading End
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_LOAD_START	Actual Event of Loading Start
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_POD	Actual Event of POD
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_POPU	Actual Event of POPU
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_END	Actual Event of Unloading End
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_START	Actual Event of Unloading Start
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_DELAY	Actual Event of Delay
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FU_DELAY	Actual Event of FU Delay
Event Data Extractors	TMS_TOR	ZSST_GTT_EXTR_EVT_FU_DELAY	Data Extractor for Event Delay of Freight Unit
Event Data Extractors	TMS_TOR	ZSST_GTT_EXTR_EVT_TU_DELAY	Data Extractor for Event Delay of Freight Unit
GTT relevance function of AOT	TMS_TOR	ZSST_GTT_OTE_FO_HDR_REL	Extractor for relevance determination for Freight Order and Freight Booking
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL_REL	Extractor for relevance determination for Arrival
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_COUPLING_REL	Extractor for relevance determination for Coupling
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_DECOUPLING_REL	Extractor for relevance determination for Decoupling
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_DEPARTURE_REL	Extractor for relevance determination for Departure
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_LOAD_END_REL	Extractor for relevance determination for Load End
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_LOAD_START_REL	Extractor for relevance determination for Load Start
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_POD_REL	Extractor for relevance determination for POD
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_POPU_REL	Extractor for relevance determination for POPU
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_END_REL	Extractor for relevance determination for Unload End
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_STRT_REL	Extractor for relevance determination for Unload Start
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_DELAY_REL	Extractor for relevance determination for FO Delay
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FU_DELAY_REL	Extractor for relevance determination for FU Delay
Planned Event Extractors	TMS_TOR	ZSST_GTT_EE_FO_HDR	Planned Event for Freight Order and Freight Booking
Tracking ID Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HEADER_TID	Function for setup of tracking IDs of Freight Order and Freight Booking

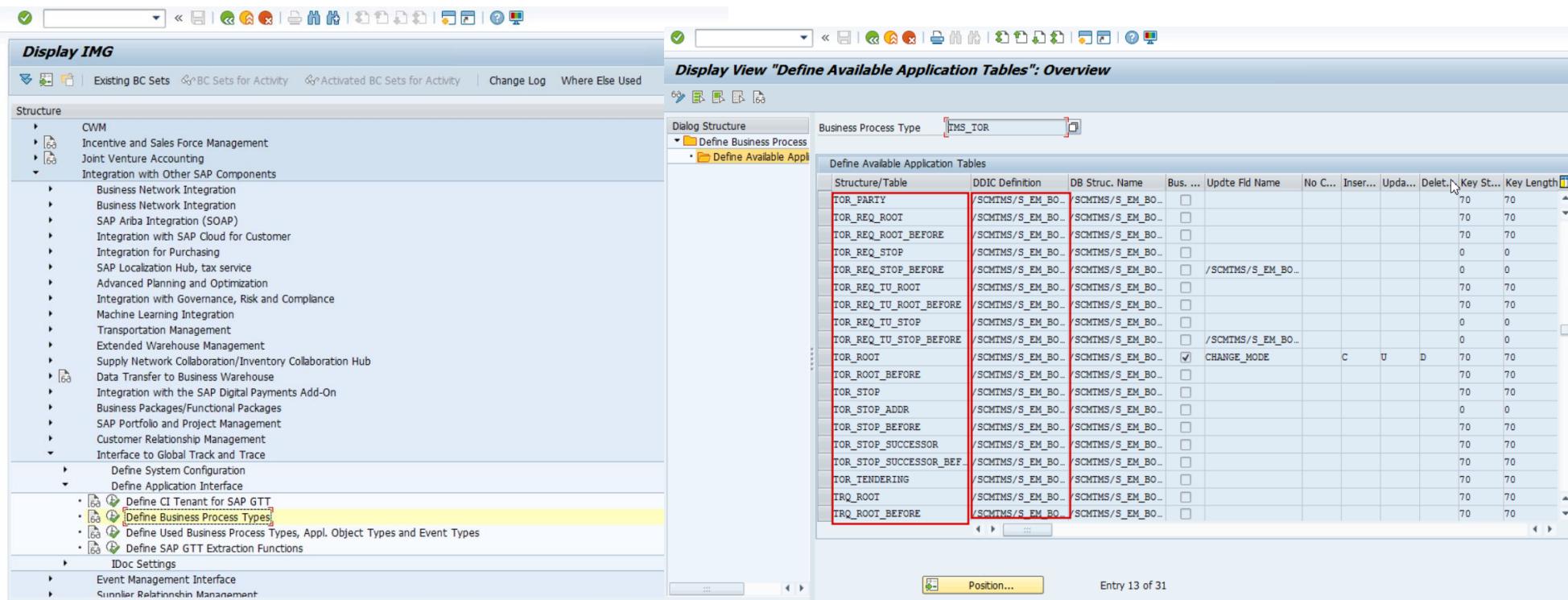
4: Available Contexts for the Extractors Modules

4-1: In **Display IMG** page, clic

Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

4-2: Choose activity **Define Business Process Types**

4-3: Select the Business Process Types to find all the context tables and their structure info



5: Coding Tips in the GTT Relevance Function Modules

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

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

See sample code of function module: *ZSST_GTT_OTE_FO_HDR_REL*

The screenshot displays the SAP ABAP Development Workbench interface with two main windows open:

- Function Builder: Display ZSST_GTT_OTE_FO_HDR_REL**: This window shows the source code for the *ZSST_GTT_OTE_FO_HDR_REL* function module. The code includes declarations for *lt_app_objects*, *lo_udm_message*, and *ls_bapiret*. It features a *TRY* block that calls *lcl_ef_performer->check_relevance* and handles errors using *CATCH cx_udm_message INTO lo_udm_message*. The code also includes *EXPORTING* and *IMPORTING* sections for *lo_udm_message* and *ls_bapiret*. A note at the bottom indicates that the function module is part of the *ZSST_GTT* package.
- ABAP Editor: Display Include LZSST_GTT_D20**: This window shows the source code for the *LZSST_GTT_D20* include module. The code defines a method *lif_bo_reader-check_relevance* that performs various checks and assignments. It includes logic for handling different event types and conditions, such as *is_app_object-maintabdef* and *is_app_object-update_indicator*. The code is heavily annotated with SAP-specific symbols (green squares, yellow triangles) indicating different types of logic or annotations.

6: Coding Tips in the Tracking ID Function Modules

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

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

See sample code of function module: *ZSST_GTT_OTE_FO_HEADER_TID*. Main logic for Freight Order and Freight Booking Tracking ID: *LCL_BO_FREIGHT_ORDER_READER* and *LCL_BO_FREIGHT_BOOKING_READER*, method *LIF_BO_READER~GET_TRACK_ID_DATA*

The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZSST_GTT_OTE_FO_HEADER_TID". The code editor displays the ABAP code for the function module ZSST_GTT_OTE_FO_HEADER_TID. The code handles tracking ID retrieval for a freight order header. It uses the LCL_BO_FACTORY class to get the track ID data and then stores it in a table named E_TRACKIDDATA. Error handling is included to manage failed lookups.

```
DATA: lo_udm_message TYPE REF TO cx_udm_message,
      ls_bapiret TYPE bapiret2.

TRY.
  lcl_ef_performer->get_track_id_data(
    EXPORTING
      is_definition = VALUE #(          maintab = lif_sst_constants->cs_tabledesc-fo_header_new )
      io_bo_factory = NEW lcl_bo_factory( )
      iv_appsyst = i_appsyst
      is_app_obj_types = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_type_cntl_tabs = i_app_type_cntl_tabs
      it_app_objects = i_app_objects
    IMPORTING
      et_track_id_data = e_trackidata[])
  .
  CATCH cx_udm_message INTO lo_udm_message.
  lcl_tools->get_errors_log( )
  EXPORTING
    io_udm_message = lo_udm_message
    iv_appsyst = i_appsyst
  IMPORTING
    es_bapiret = ls_bapiret .
  "
  add error message
  Scope:|FUNCTION ZSST_GTT_OTE_FO_HEADER_TID|TRY
```

The screenshot shows the SAP ABAP Editor interface with the title "ABAP Editor: Display Include LZSST_GTT_D20". The code editor displays the ABAP code for the include LZSST_GTT_D20, specifically the method lif_bo_reader~get_track_id_data. This method implements the LIF_BO_READER~GET_TRACK_ID_DATA interface. It retrieves the root node (lr_root) from the parameters and then performs a delta transport to update the track ID data based on the provided parameters.

```
METHOD lif_bo_reader~get_track_id_data.
  DATA: lr_item TYPE REF TO data,
        lr_item_old TYPE REF TO data,
        lt_track_id_data TYPE lif_ef_types->tt_enh_track_id_data,
        lt_track_id_data_old TYPE lif_ef_types->tt_enh_track_id_data,
        lr_root_new TYPE REF TO data,
        lr_root_old TYPE REF TO data.

  FIELD-SYMBOLS: <lt_item> TYPE ANY_TABLE,
                 <lt_item_old> TYPE ANY_TABLE,
                 <ls_root> TYPE /smmtms/s_em_bo_tor_root,
                 <lt_root_new> TYPE /smmtms/t_em_bo_tor_root,
                 <lt_root_old> TYPE /smmtms/t_em_bo_tor_root.

  ASSIGN is_app_object-maintabref->* TO <ls_root>.
  IF sy-subrc <> 0.
    RETURN.
  ENDIF.

  lr_root_new = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledesc-fo_header_new).

  lr_root_old = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledesc-fo_header_old).

  ASSIGN lr_root_new->* TO <lt_root_new>.
  IF sy-subrc <> 0.
    RETURN.
  ENDIF.

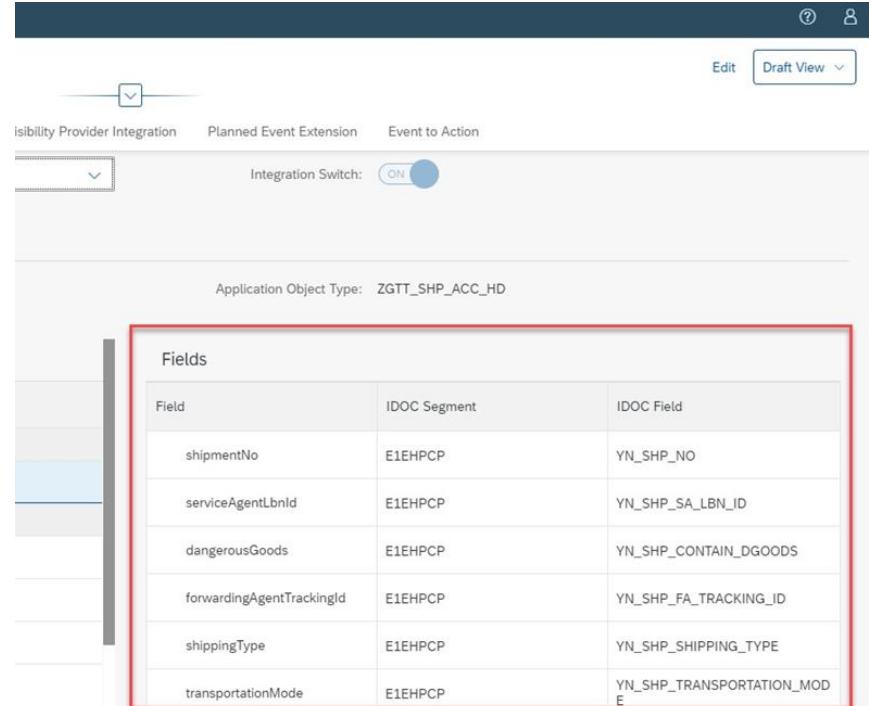
  Scope:|CLASS lcl_bo_freight_booking_reader|METHOD lif_bo_reader~get_trac...|ABAP | Ln 1636 Col 76
```

7: Coding Tips in the Control Parameter Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_CONTROL_DATA*.
3. GTT v2 asks for full transport for all the control parameters, which means that all the fields need to be extracted in all cases, no matter whether their values have been changed.
4. To fill up the composition (table) fields defined in Manage Model applications, use single field table types for all fields in composition, *PARAMINDEX* will be incremented automatically. If the field is empty, 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 application. It's not recommended to fill a code list type field to clear a composition even if it's a key field.
6. The field with fixed name '*ACTUAL_BUSINESS_DATETIME*' and '*ACTUAL_BUSINESS_TIMEZONE*' are mandatory fields to be transported for event handling sequencing in GTT V2.
7. In Manage Model application, click tab *IDOC Integration* to map the parameter names and model field names.
8. For DATE or DATETIME fields, when the source value is initial like '00000000' '0000000000000000', then please ensure to only enable *PARAMNAME* and *PARAMINDEX* in the extractor code, not enable *VALUE* for IDOC sending.
9. For Amount field which has reference currency, please ensure to call BAPI '*BAPI_CURRENCY_CONV_TO_EXTERNAL*' using the reference currency to make the amount tracked correctly by GTT v2. The BAPI will output the conversion result in 4 decimals as fixed, which needs additional rounding in the extractor if the corresponding field defined in the tracking model is of less than 4 decimals.

See sample code of function module: *ZSST_GTT_OTE_FO_HDR*. Main logic for Freight Order and Freight Booking Control parameters: *LCL_BO_FREIGHT_ORDER_READER* and *LCL_BO_FREIGHT_BOOKING_READER*, method *LIF_BO_READER~GET_DATA*



The screenshot shows the SAP ALE Integration Switch configuration interface. At the top, there are tabs for 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below these, there is a section labeled 'Integration Switch' with a toggle switch set to 'ON'. The application object type is specified as 'ZGTT_SHP_ACC_HD'. A red box highlights a table titled 'Fields' under the 'Event to Action' tab, which maps IDOC segments to IDOC fields. The table columns are 'Field', 'IDOC Segment', and 'IDOC Field'. The mapped fields are:

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCP	YN_SHP_NO
serviceAgentLbnId	E1EHPCP	YN_SHP_SA_LBN_ID
dangerousGoods	E1EHPCP	YN_SHP_CONTAIN_DGOODS
forwardingAgentTrackingId	E1EHPCP	YN_SHP_FA_TRACKING_ID
shippingType	E1EHPCP	YN_SHP_SHIPPING_TYPE
transportationMode	E1EHPCP	YN_SHP_TRANSPORTATION_MODE

7: Coding Tips in the Control Parameter Function Modules

ABAP Editor: Display Include LSSST_GTT_D20

```

840 *      cx_udm_message.
841 ENDCLASS.
842
843 CLASS lcl_bo_freight_order_reader IMPLEMENTATION.
844
845 METHOD lif_bo_reader~get_data.
846
847   DATA: lr_fo TYPE REF TO data.
848   FIELD-SYMBOLS: <ls_freight_order> TYPE ts_fo_header,
849                  <ls_fo>          TYPE any,
850                  <ls_maintabref>  TYPE any,
851                  <lt_maintabref>  TYPE ANY TABLE.
852
853   DATA(lr_maintabref) = get_maintabref( is_app_object ).
854
855   rr_data = NEW ts_fo_header( ).
856   ASSIGN rr_data->* TO <ls_freight_order>.
857   IF sy-subrc <> 0.
858     MESSAGE e010(zsst_gtt) INTO DATA(lv_dummy).
859     lcl_tools->throw_exception( ).
860   ENDIF.
861
862   get_data_from_root(
863     EXPORTING
864       iv_old_data = iv_old_data
865       ir_root     = lr_maintabref
866     CHANGING
867       cs_fo_header = <ls_freight_order> .
868   IF <ls_freight_order> IS INITIAL.
869     RETURN.
870   ENDIF.

```

Function Builder: Display ZSST_GTT_OTE_FO_HDR

```

19 DATA: lo_udm_message TYPE REF TO cx_udm_message,
20      ls_bapiret    TYPE bapiret2.
21
22 TRY.
23   lcl_ef_performer->get_control_data(
24     EXPORTING
25       is_definition      = VALUE #(
26         maintab = lif_sst_co-
27         io_bo_factory     = NEW lcl_tor_factory( )
28         iv_appsyst        = i_appsyst
29         is_app_obj_types  = i_app_obj_types
30         it_all_appl_tables = i_all_appl_tables
31         it_app_type_cntl_tabs = i_app_type_cntl_tabs
32         it_app_objects    = i_app_objects
33     CHANGING
34       ct_control_data   = e_control_data[] ).
35
36 CATCH cx_udm_message INTO lo_udm_message.
37   lcl_tools->get_errors_log( )
38
39 EXPORTING
40   io_udm_message = lo_udm_message
41   iv_appsyst    = i_appsyst
42
43 IMPORTING
44   es_bapiret    = ls_bapiret .
45
46 APPEND ls_bapiret TO e_logtable.
47
48 CASE lo_udm_message->textid.

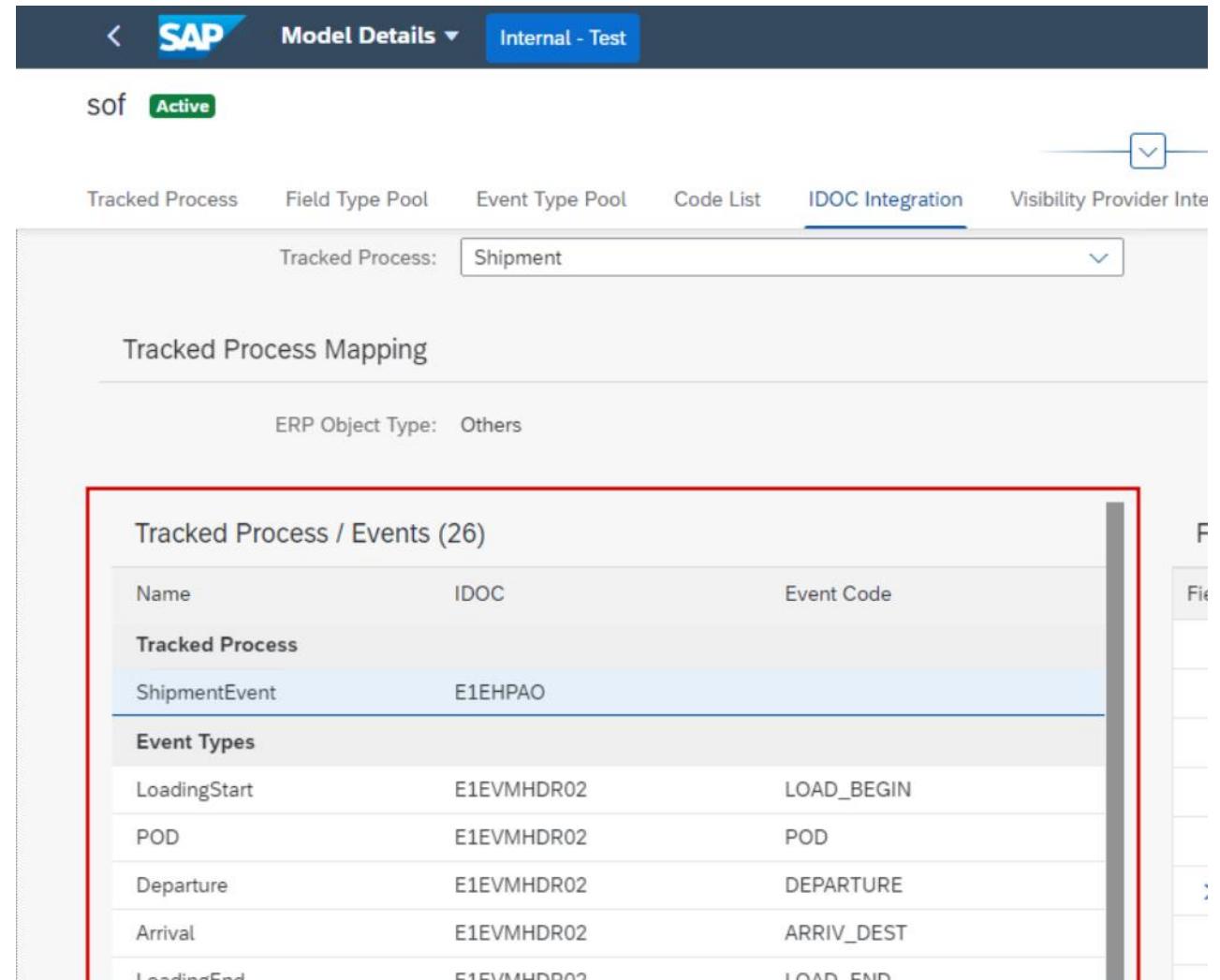
```

8: Coding Tips in the Planned Event Function Modules

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

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. GTT v2 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.
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* application in GTTV2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function module: *ZSST_GTT_EE_FO_HDR*,
Main logic for Freight Order and Freight Booking Control parameters:
LCL_PE_FILLER_FO_HEADER, method
LIF_PE_FILLER~GET_PLANED_EVENTS



The screenshot shows the SAP Model Details interface for a model named 'sof' (status: Active). The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process' is set to 'Shipment'. In the 'Tracked Process / Events' section, a table lists tracked processes and their corresponding IDOCs and event codes. The table has columns: Name, IDOC, and Event Code. It includes rows for 'Tracked Process' (ShipmentEvent, E1EHPAO), 'Event Types' (LoadingStart, E1EVMHDR02, LOAD_BEGIN; POD, E1EVMHDR02, POD; Departure, E1EVMHDR02, DEPARTURE; Arrival, E1EVMHDR02, ARRIV_DEST; LoadingEnd, E1EVMHDR02, LOAD_END).

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

8: Coding Tips in the Planned Event Function Modules

ABAP Editor: Display Include LZSST_GTT_D30

```

METHOD lif_pe_filler~get_planed_events.

  DATA: lv_tor_id  TYPE /scmtms/tor_id,
        lv_tor_cat  TYPE /scmtms/tor_category,
        lr_stop     TYPE REF TO data,
        lr_loc_addr TYPE REF TO data,
        ls_loc_addr TYPE REF TO /scmtms/s_em_bo_loc_addr.

  FIELD-SYMBOLS: <lt_stop>      TYPE /scmtms/t_em_bo_tor_stop,
                  <lt_loc_addr> TYPE /scmtms/t_em_bo_loc_addr.

  lv_tor_id   = lcl_tools->get_field_of_structure(
                ir_struct_data = is_app_objects-maintabref
                iv_field_name  = 'TOR_ID' ).

  SHIFT lv_tor_id LEFT DELETING LEADING '0'.

  lv_tor_cat  = lcl_tools->get_field_of_structure(
                ir_struct_data = is_app_objects-maintabref
                iv_field_name  = 'TOR_CAT' ).

  lr_stop     = mo_ef_parameters->get_appl_table(
                iv_tabledef = lif_sst_constants->cs_tabledef-fo_stop_new ).

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

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

```

Scope: \CLASS lcl_pe_filler_fo_header\METHOD lif_pe_filler~get_planed_events | ABAP | Ln 581 Col 67

Function Builder: Display ZSST_GTT_EE_FO_HDR

```

DATA: lo_udm_message TYPE REF TO cx_udm_message,
      ls_bapiret     TYPE bapiret2.

CLEAR e_logtable[].

LOOP AT i_app_objects ASSIGNING FIELD-SYMBOL(<ls_app_objects>) WHERE maindbtabdef IS NOT INITIAL.

TRY.
  lcl_ef_performer->get_planned_events(
    EXPORTING
      is_definition  = maintab  = lif_sst_constants->cs_tabledef-fo_header_new
      io_factory     = NEW lcl_tor_factory()
      iv_apps     = i_apps
      is_app_obj_types = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
      it_app_objects = i_app_objects
    CHANGING
      ct_expeventdata = e_expeventdata[]
      ct_measrmntdata = e_measrmntdata[]
      ct_infodata = e_infodata[]
    IMPORTING
      CATCH cx_udm_message INTO lo_udm_message.
      lcl_tools->get_errors_log(
        EXPORTING
          io_udm_message = lo_udm_message
          iv_apps     = i_apps
        IMPORTING
          )
  ).

```

Scope: \FUNCTION ZSST_GTT_EE_FO_HDR\LOOP\TRY | ABAP | Ln 37 Col 19

8: Coding Tips in the Planned Event Function Modules

For customers who implemented before February release 2021 and are still using SAP S/4HANA 1909 SP00 – SP01, to extract planned events, you need to apply the following Postal Address data method `get_postal_address()` of class `Icl_tools`.

```
Include LZSST_GTTD10 Active

540 METHOD get_postal_address.
541   DATA(lo_tor_srv_mgr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_tor_c=>sc_bo_key).
542   DATA(lo_loc_srv_mgr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
543
544   lo_tor_srv_mgr->retrieve_by_association(
545     EXPORTING
546       iv_node_key      = /scmtms/if_tor_c=>sc_node-root
547       it_key          = VALUE #( ( key = iv_node_id ) )
548       iv_association = /scmtms/if_tor_c=>sc_association-root-stop
549     IMPORTING
550       et_target_key   = DATA(lt_stop_target_key) .
551
552   IF lt_stop_target_key IS NOT INITIAL.
553     lo_tor_srv_mgr->retrieve_by_association(
554       EXPORTING
555         iv_node_key      = /scmtms/if_tor_c=>sc_node-stop
556         it_key          = CORRESPONDING #( lt_stop_target_key )
557         iv_association = /scmtms/if_tor_c=>sc_association-stop-bo_loc_log
558       IMPORTING
559         et_key_link     = DATA(lt_loc_log_key_link) .
560
561   IF lt_loc_log_key_link IS NOT INITIAL.
562     lo_loc_srv_mgr->retrieve_by_association(
563       EXPORTING
564         iv_node_key      = /scmtms/if_location_c=>sc_node-root
565         it_key          = CORRESPONDING #( lt_loc_log_key_link MAPPING key = target_key )
566         iv_association = /scmtms/if_location_c=>sc_association-root-address
567       IMPORTING
568         et_key_link     = DATA(lt_address_key_link) .
569
570   IF lt_address_key_link IS NOT INITIAL.
571     TRY.
572       DATA(lr_bo_conf) = /bobf/cl_frw_factory->get_configuration(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
573       CATCH /bobf/cx_frw.
574         MESSAGE e011(zsst_gtt) INTO DATA(lv_dummy).
575         lcl_tools->throw_exception( ).
576     ENDTRY.
577
578     DATA(lv_postal_ass_key) = lr_bo_conf->get_content_key_mapping(
579       iv_content_cat    = /bobf/if_conf_c=>sc_content_ass
580       iv_do_content_key = /bofu/if_addr_constants=>sc_association-root-postal_address
581       iv_do_root_node_key = /scmtms/if_location_c=>sc_node-/bofu/address ) .
582
```

9: Coding Tips in the Event Data Function Modules

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

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

See sample code of function module: *ZSST_GTT_EE_FO_ARRIVAL*.
Relevance function module: *ZSST_GTT_EE_FO_ARRIVAL_REL*.

The screenshot shows the SAP Model Details interface for a tracked process named 'Shipment'. The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', it shows 'ERP Object Type: Others'. A red box highlights the 'Tracked Process / Events (26)' section, which lists the following data:

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

9: Coding Tips in the Event Data Function Modules

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL

Function Module: ZSST_GTT_EE_FO_ARRIVAL active

Attributes Import Export Changing Tables Exceptions Source Code

```

58     CALL FUNCTION '/SCMTMS/EXTR_EVT_TO_ARRIVAL'
59     EXPORTING
60         i_applsys          = i_applsys
61         i_event_type       = i_event_type
62         i_all_appl_tables = i_all_appl_tables
63         i_event_type_cntl_tabs = i_event_type_cntl_tabs
64         i_events            = i_events
65     TABLES
66         ct_trackingheader = ct_trackingheader
67         ct_tracklocation  = ct_tracklocation
68         ct_trackaddress   = ct_trackaddress
69         ct_trackparameters = ct_trackparameters
70     CHANGING
71         c_eventid_map      = c_eventid_map
72     EXCEPTIONS
73         parameter_error    = 1
74         event_data_error   = 2
75         stop_processing    = 3
76         OTHERS              = 4.
77     CASE sy-subrc.
78     WHEN 1.
79         RAISE parameter_error.
80     WHEN 2.
81         RAISE event_data_error.
82     WHEN 3.
83         RAISE stop_processing.
84     ENDIF.
    
```

Scope: \FUNCTION zsst_gtt_ee_fo_arrival\{CASE ABAP

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL_REL

Function Module: ZSST_GTT_EE_FO_ARRIVAL_REL active

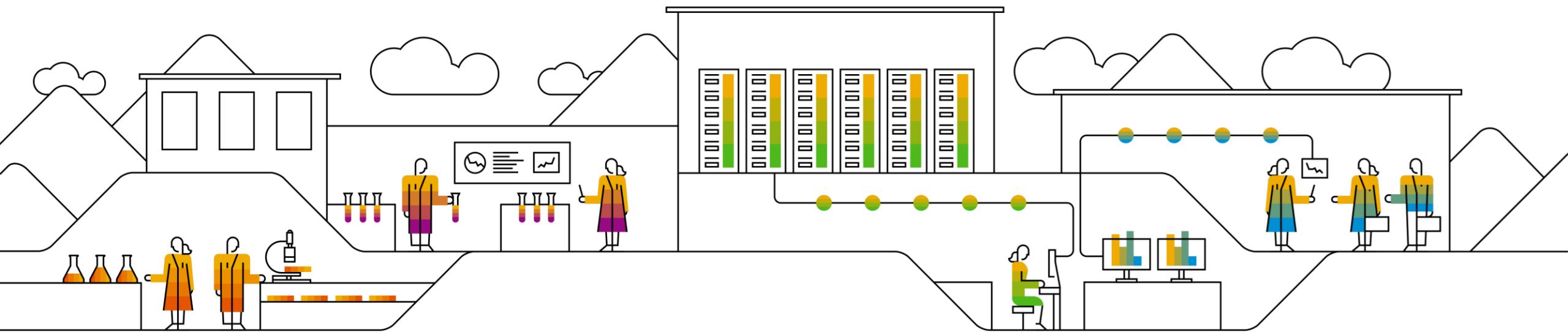
Attributes Import Export Changing Tables Exceptions Source Code

```

1  FUNCTION zsst_gtt_ee_fo_arrival_rel.
2
3  *"-- Local Interface:
4  *-- IMPORTING
5  *--   REFERENCE(I_APPLSYS) TYPE /SAPTRX/APPLSYSTEM
6  *--   REFERENCE(I_EVENT_TYPES) TYPE /SAPTRX/EVTYPES
7  *--   REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER
8  *--   REFERENCE(I_EVENTTYPE_TAB) TYPE TRXAS_EVENTTYPE_TABS_WA
9  *--   REFERENCE(I_EVENT) TYPE TRXAS_EVT_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 TRY.
20     lcl_actual_event->get_tor_actual_event_class( i_event )->check_event_relevance(
21     EXPORTING
22         i_all_appl_tables = i_all_appl_tables
23         iv_event_code    = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
24         i_event           = i_event
25     IMPORTING
26         e_result          = e_result .
27     CATCH cx_udm_message INTO DATA(lo_udm_message).
    
```

Scope: \FUNCTION zsst_gtt_ee_fo_arrival_rel\TRY ABAP | Ln 27 Col 52

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[®]