



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

Logistics Business Network
March 2021

PUBLIC

Objectives



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

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

Agenda

A Prerequisites

B Configuration and Implementation - Basic

 B1 IDOC Configuration

 B2 Extractor Configuration

C Download ABAP Code from GitHub

D Configuration and Coding Guide - Advanced



A) Prerequisites



STEP 1: Check the SAP Version

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

1-2: The node “Interface to Global Track and Trace” in the IMG and the related GTT-specific versions of the IMG activities are available in the software component version SAP_BW 750 from SP 12 on. They cannot be downloaded as a correction via note assistant. We recommend upgrading to the service package level accordingly.

1-3: The following SAP Notes shall be implemented:

- 2370356 - SAP Global Track and Trace Application Interface
- 2937175 - Enhancement of IDOCs sent to GTT
- 2974952 - Error in Note 2937175

STEP 1: Check the SAP Version

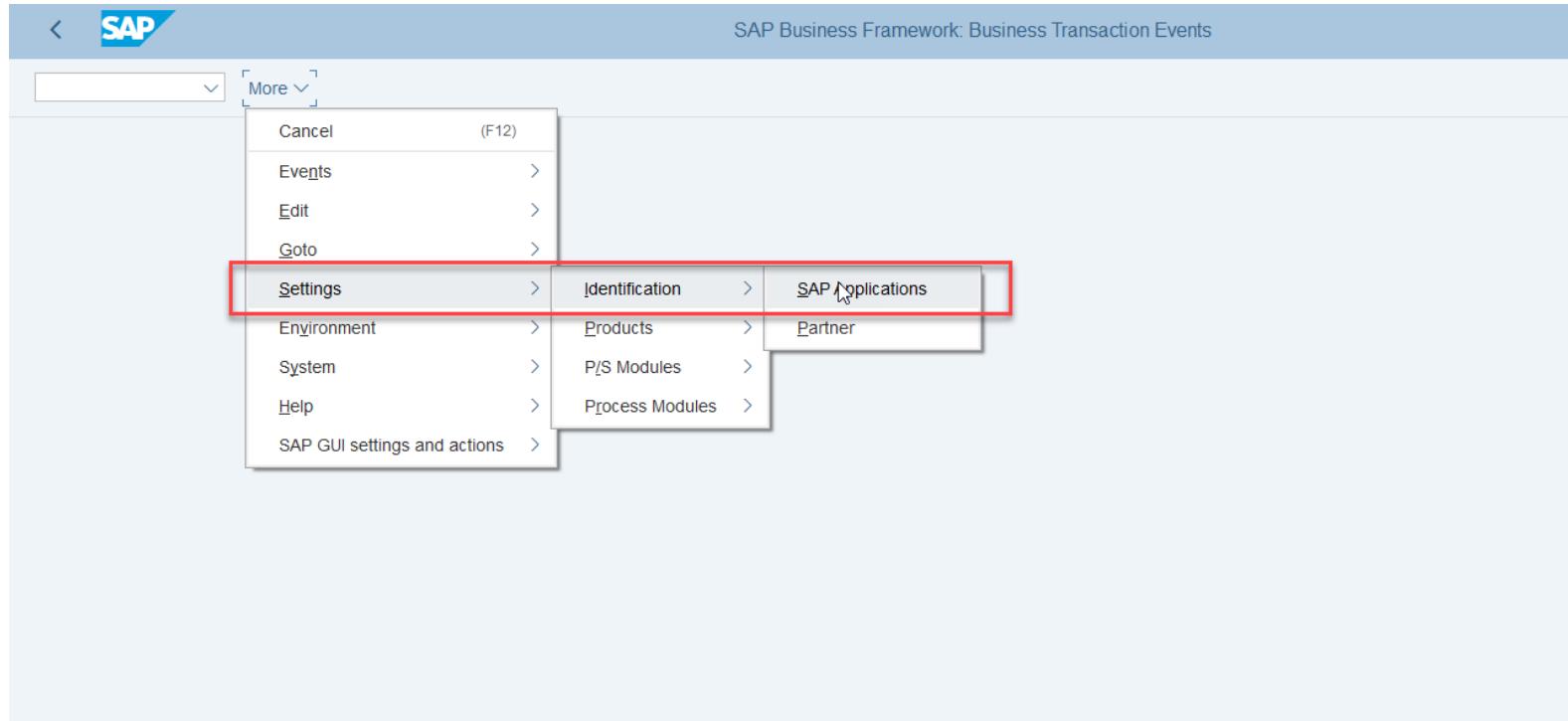
1-4: The ABAP codes on Github to support sample apps for GTT Version 2 shall be implemented in S4 HANA 1909 SP03 on premise, which is not validated in lower release, and not applicable for ECC series of products.

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>

STEP 2: Log on the Development Client to Configure BTE

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



STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**

The screenshot shows a SAP application interface titled "Change View 'BTE Application Indicator'. Overview". The main area is a table with three columns: "Appl.", "A", and "Text". The "Appl." column lists various application codes, and the "Text" column provides a brief description of each. The row for "PI-EM" is highlighted with a red box around the "Appl." cell. In the "A" column for "PI-EM", there is a checkbox that is checked, indicating the application is active. Other applications listed include PM, PM-BW, PM-EQM, PM-PAM, PMA-PC, PMAT, PMIPUR, PMPUSH, PP-BD, PP-DD, PP-MRP, PRICAT, PS-REP, PSRV, QBEXT, QBEXTP, QILPO, RDSVFI, and RDSVMD. The bottom right of the screen shows a dark footer bar with "Save" and "Cancel" buttons.

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

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



STEP 1: Define RFC Connection for GTT

1-1: Log on to the business client

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

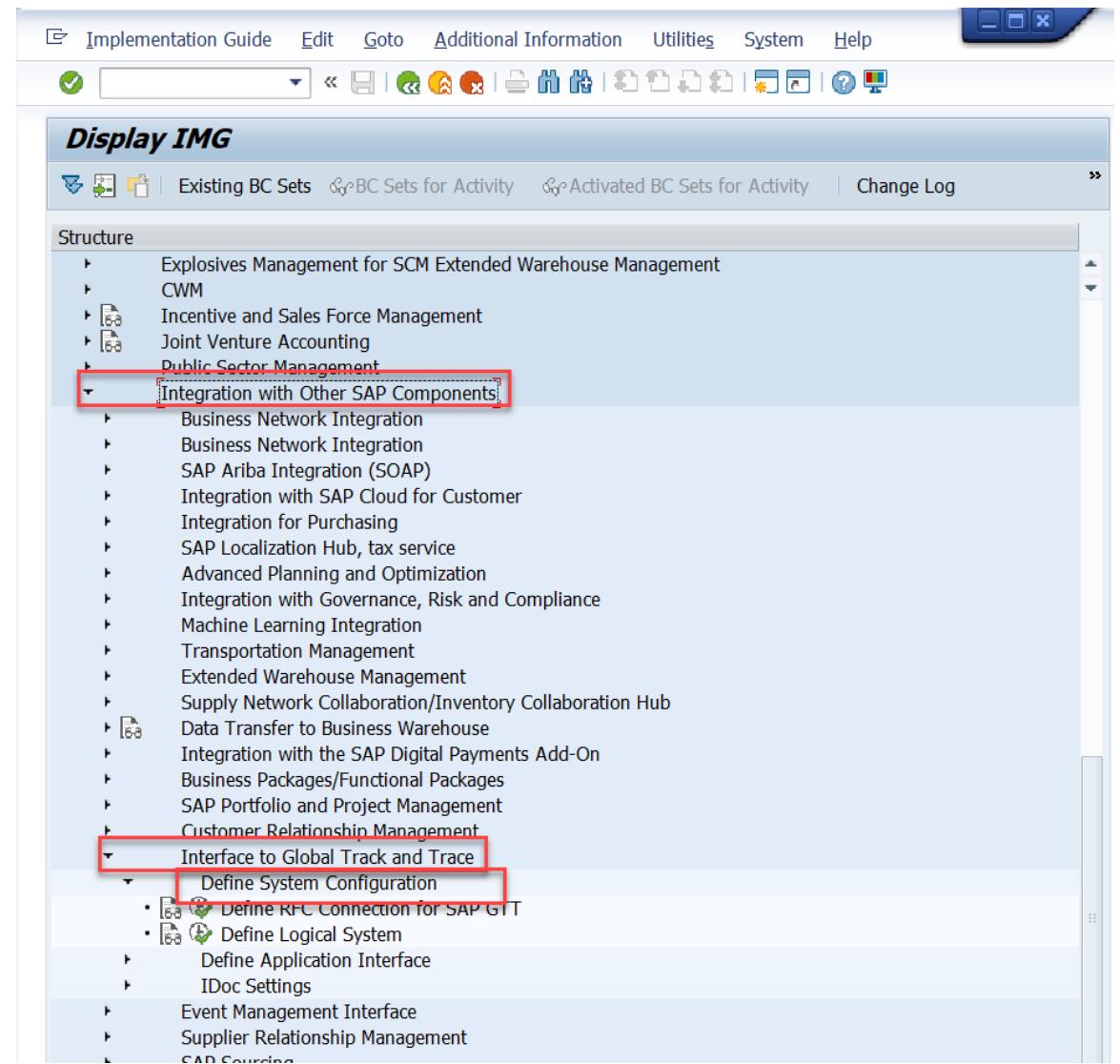
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

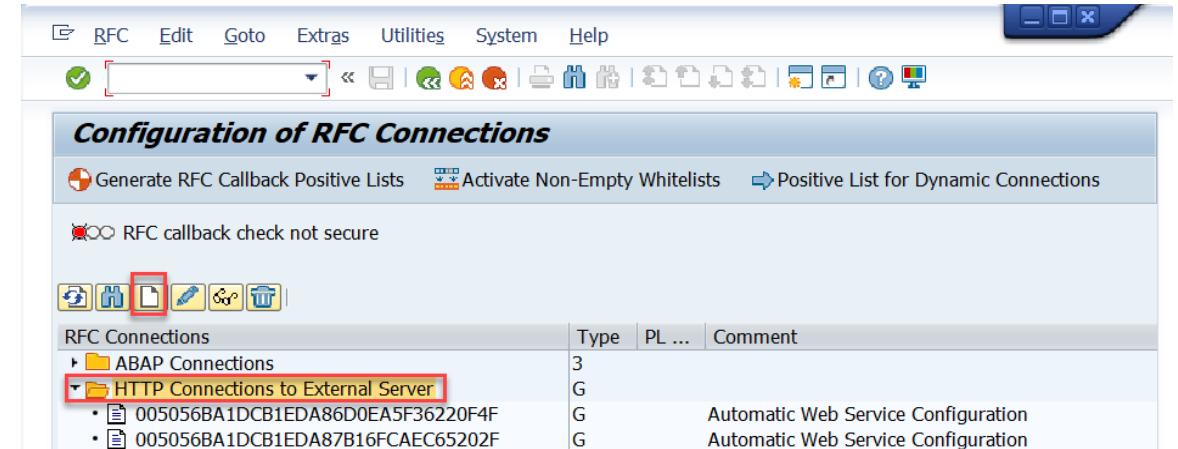
Define RFC Connection for SAP GTT



STEP 1: Define RFC Connection for GTT

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

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



STEP 1: Define RFC Connection for GTT

1-7: Enter a description

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

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

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

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

Port: `443`

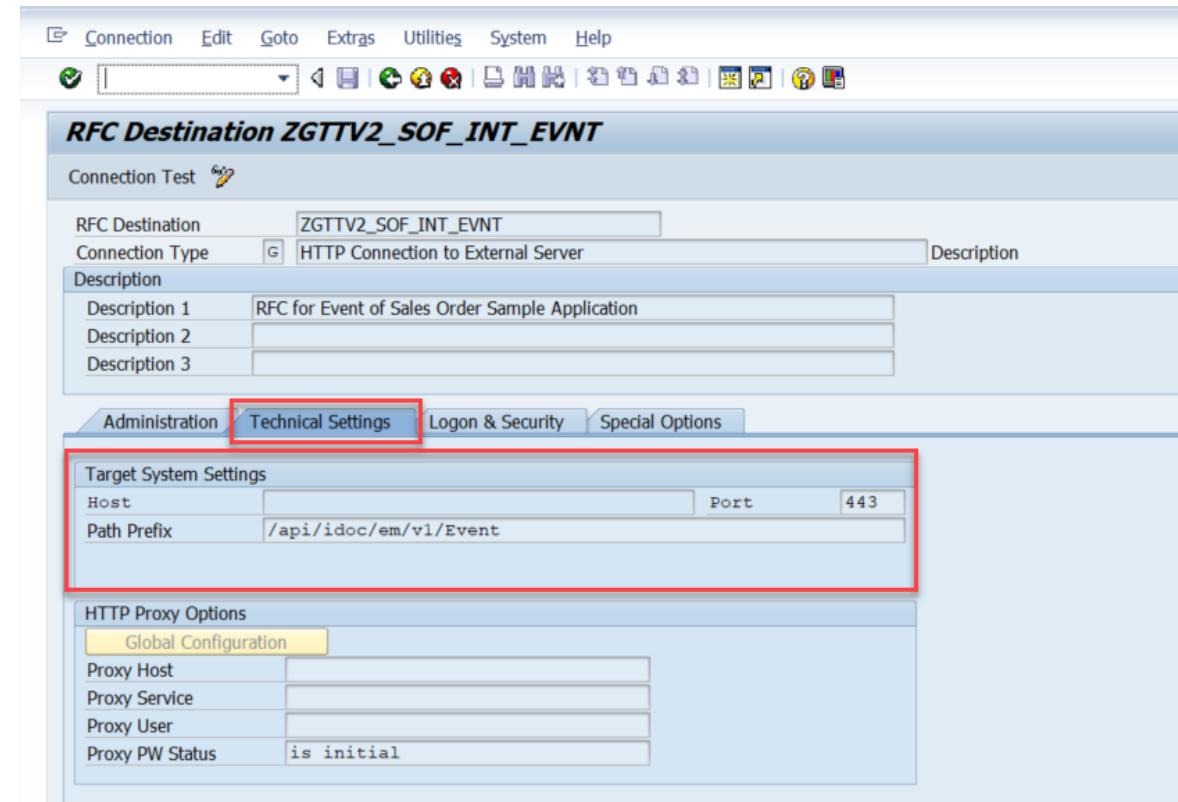
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

For the event:

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

For the tracked Process:

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



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

STEP 1: Define RFC Connection for GTT

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

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

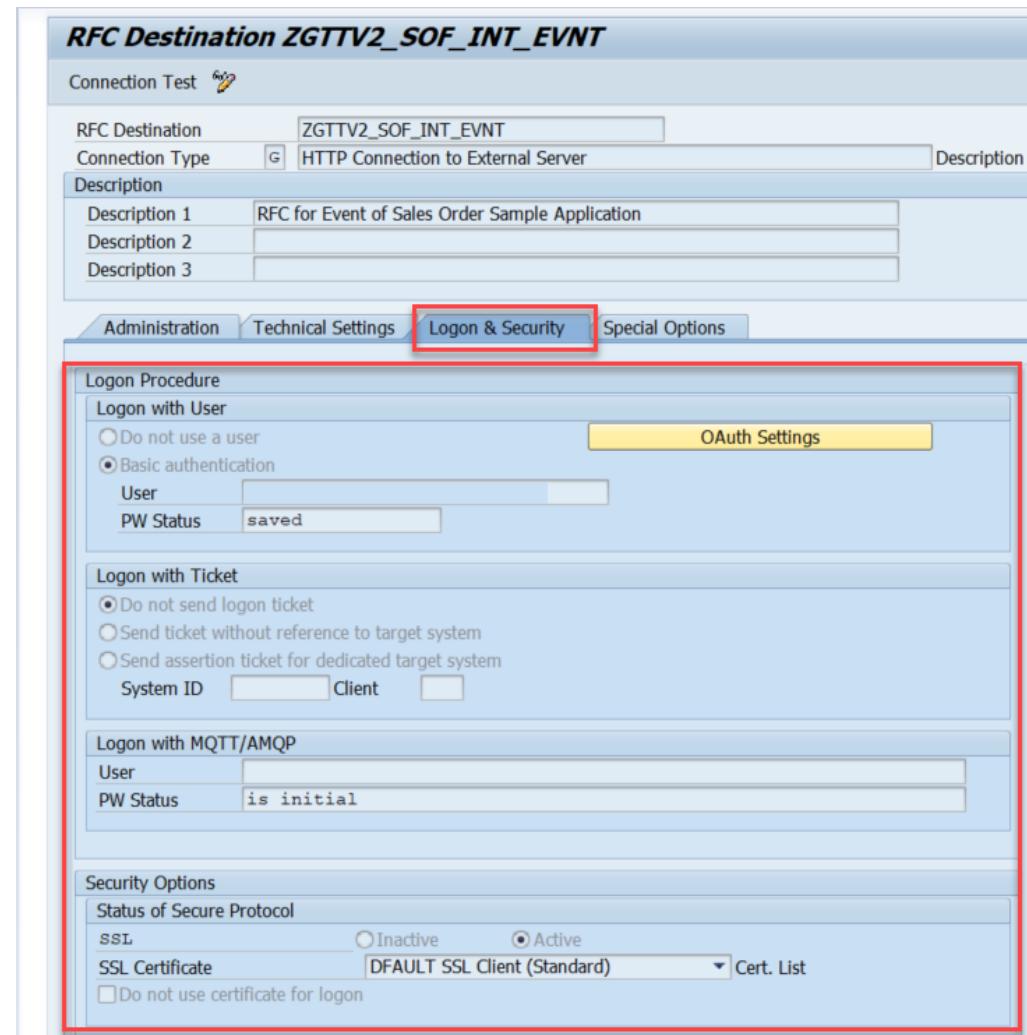
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

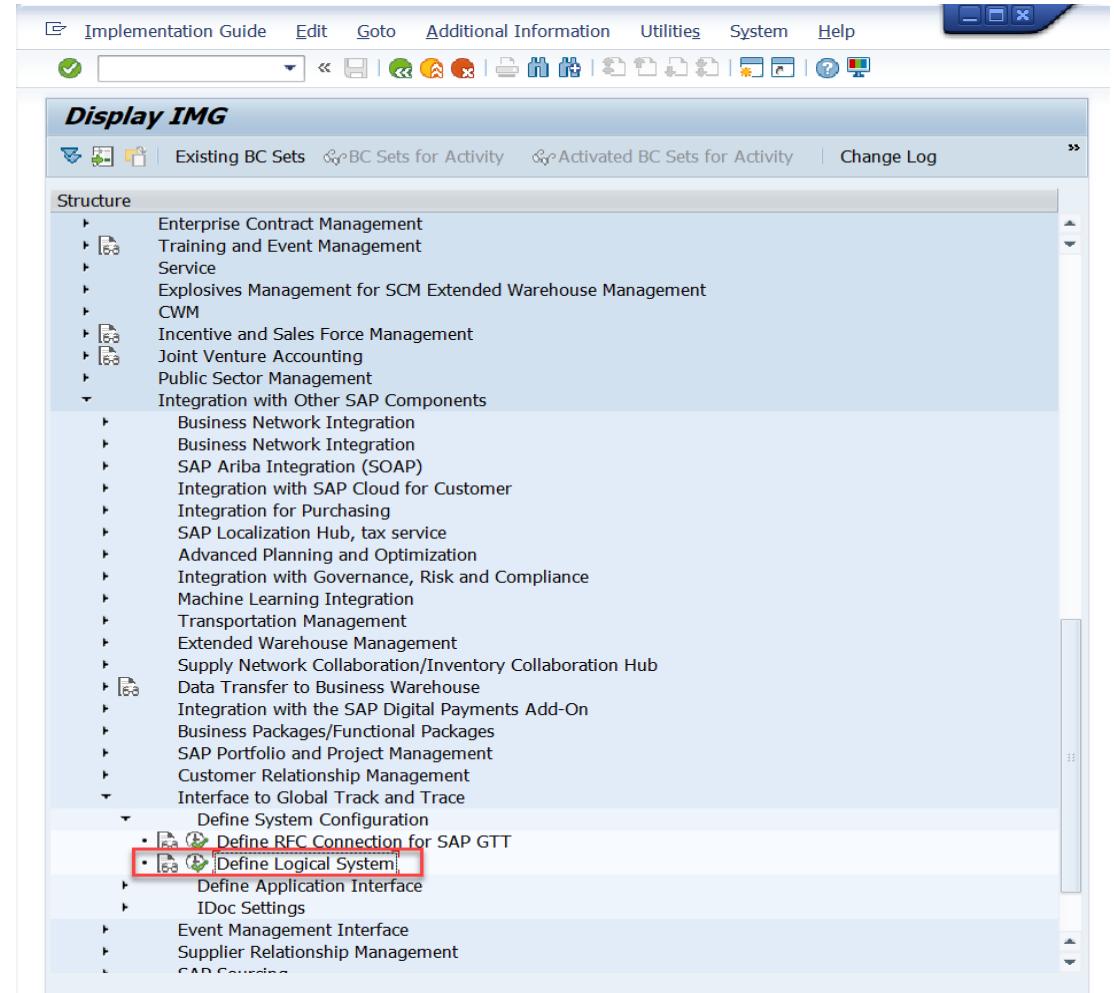
- one for event and
- the other for tracked process.



STEP 2: Define Logical System

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

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

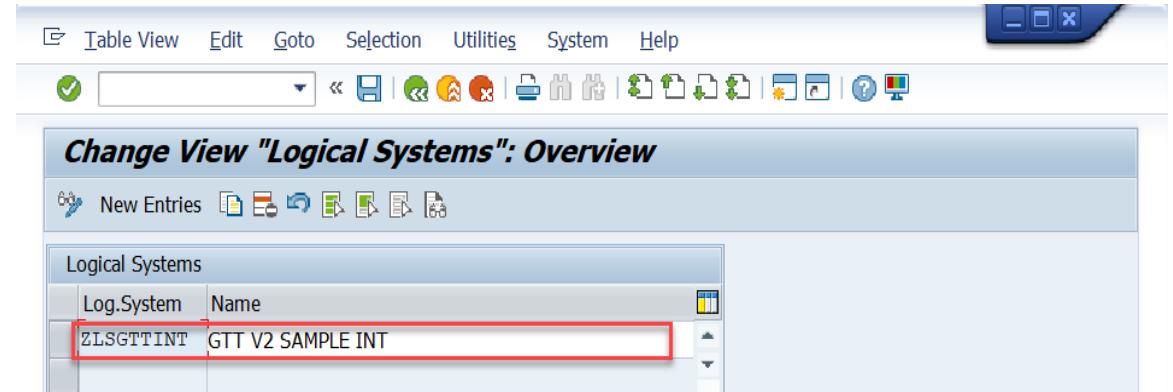


STEP 2: Define Logical System

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

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

2-4: Save the configuration

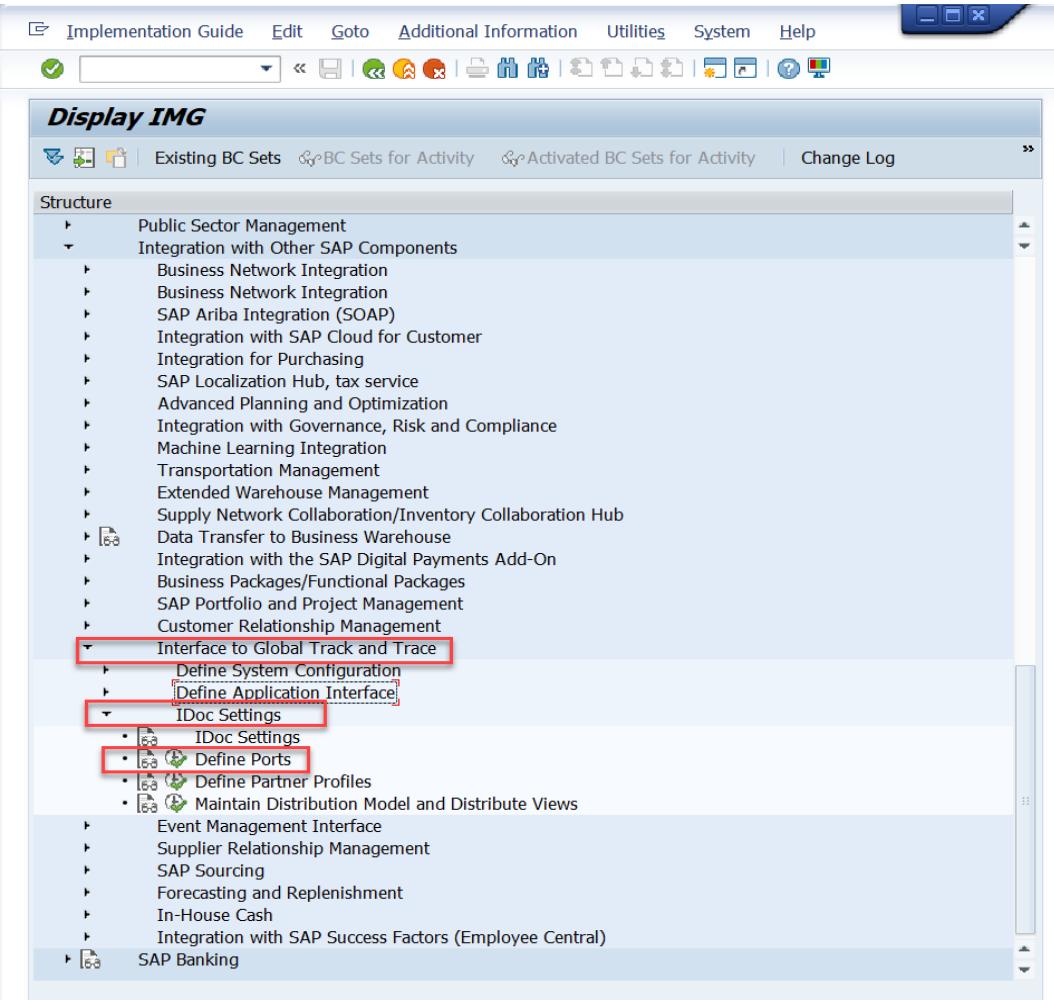


Log.System	Name
ZLSGTTINT	GTT V2 SAMPLE INT

STEP 3: Define Ports

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

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



STEP 3: Define Ports

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

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

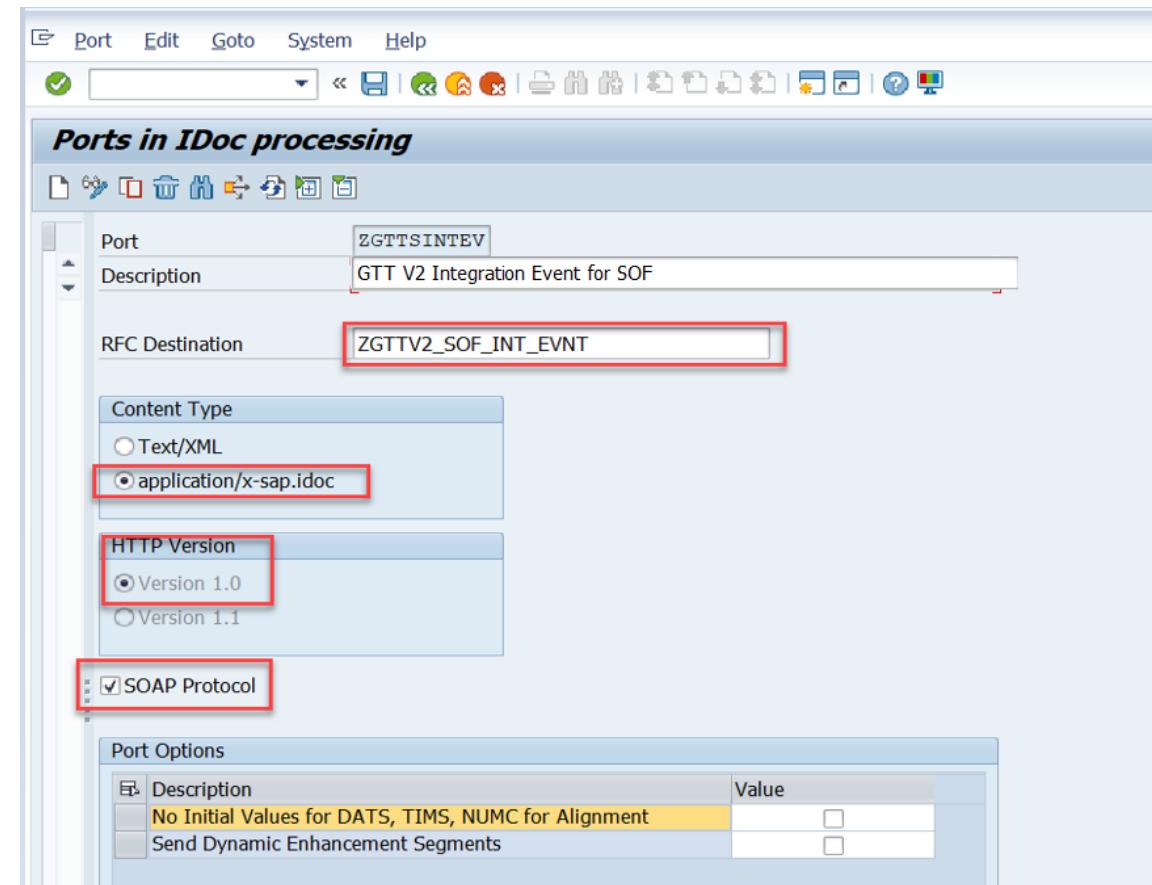
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

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

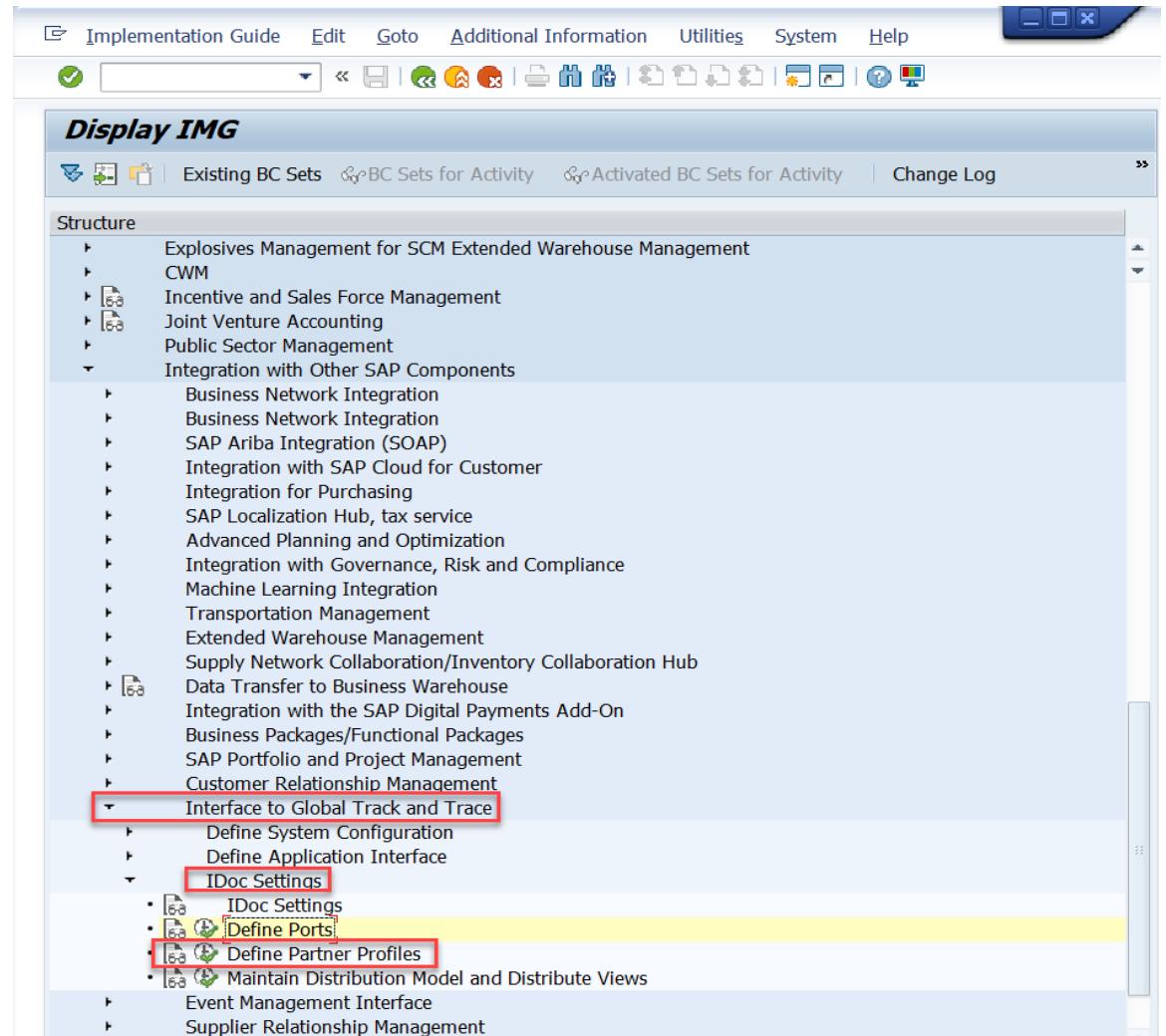


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTSINTEV	GTT V2 Integration Event for SOF	ZGTTV2_SOF_INT_EVNT	application/x-sap.idoc	Version 1.0	Checked
ZGTTSINTTP	GTT V2 Integration Tracked Process for SOF	ZGTTV2_SOF_INT_TP	application/x-sap.idoc	Version 1.0	Checked

STEP 4: Define Partner Profiles

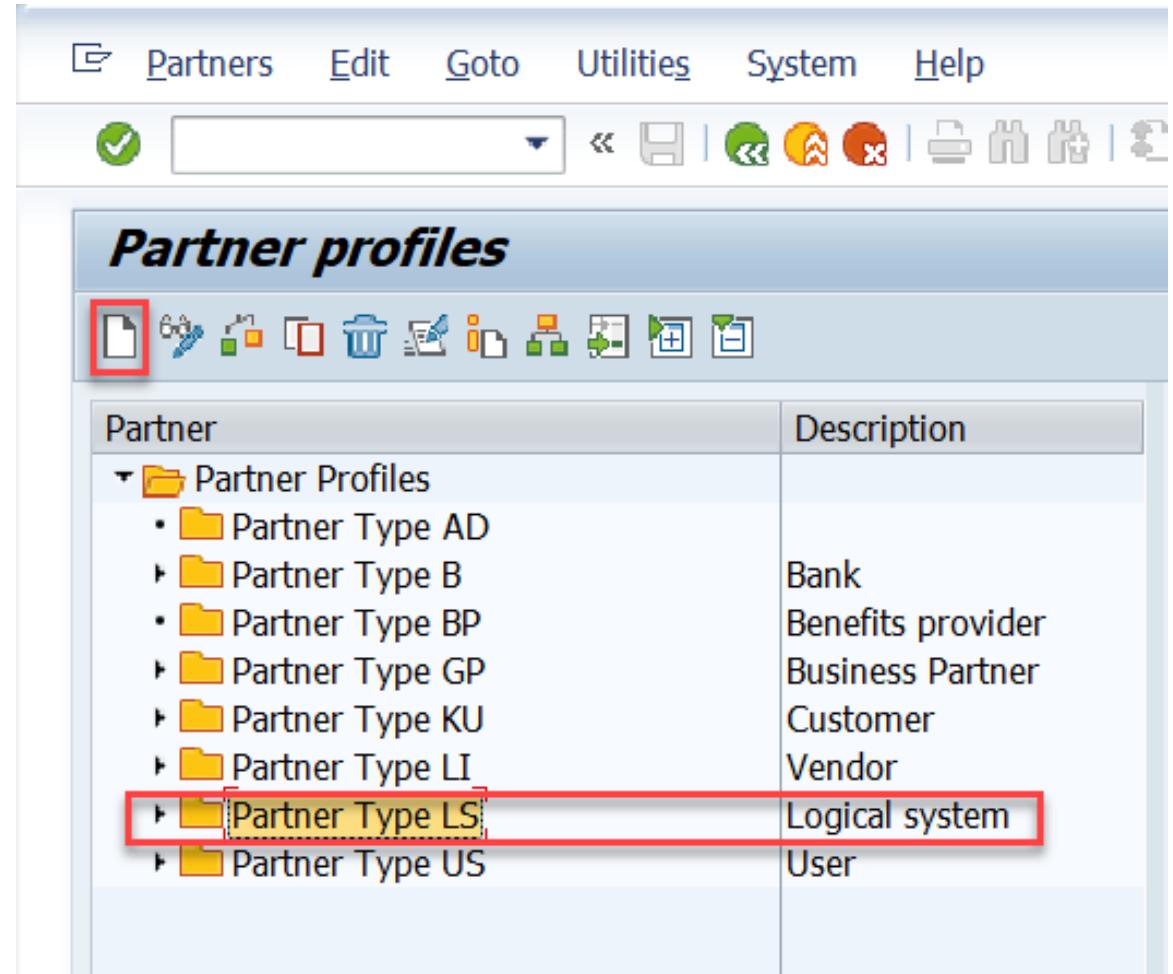
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components -> Interface to Global Track and Trace -> IDoc Settings**

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



STEP 4: Define Partner Profiles

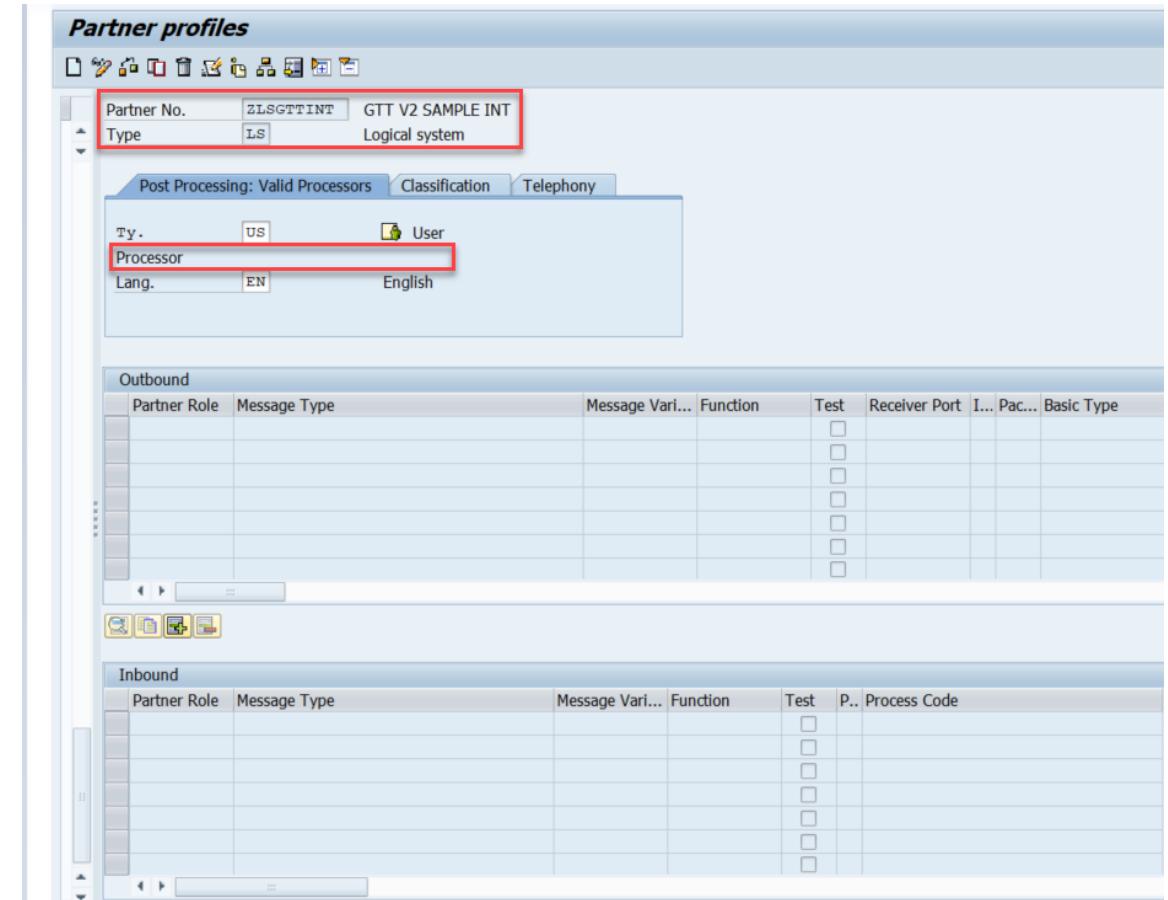
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

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

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



STEP 4: Define Partner Profiles

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

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

STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

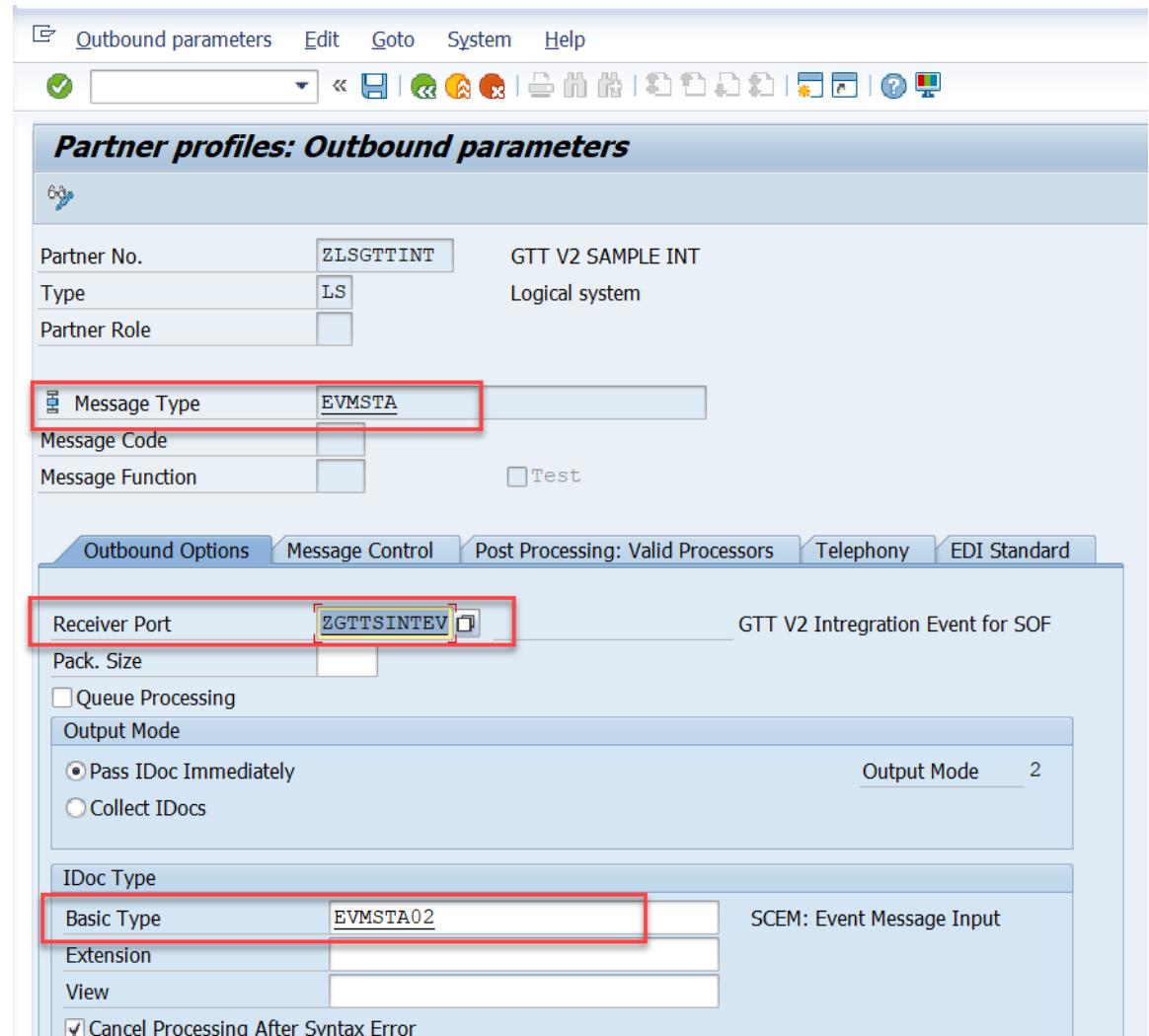
For the tracked Process:

Message Type: AOPOST

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

4-9: Save the configuration

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



STEP 4: Define Partner Profiles

4-10: Fill in the Message Type.

For the tracked Process:

Message Type: AOPOST

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

4-12: Save the configuration

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

Partner profiles: Outbound parameters

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

Message Type AOPOST AOPOST

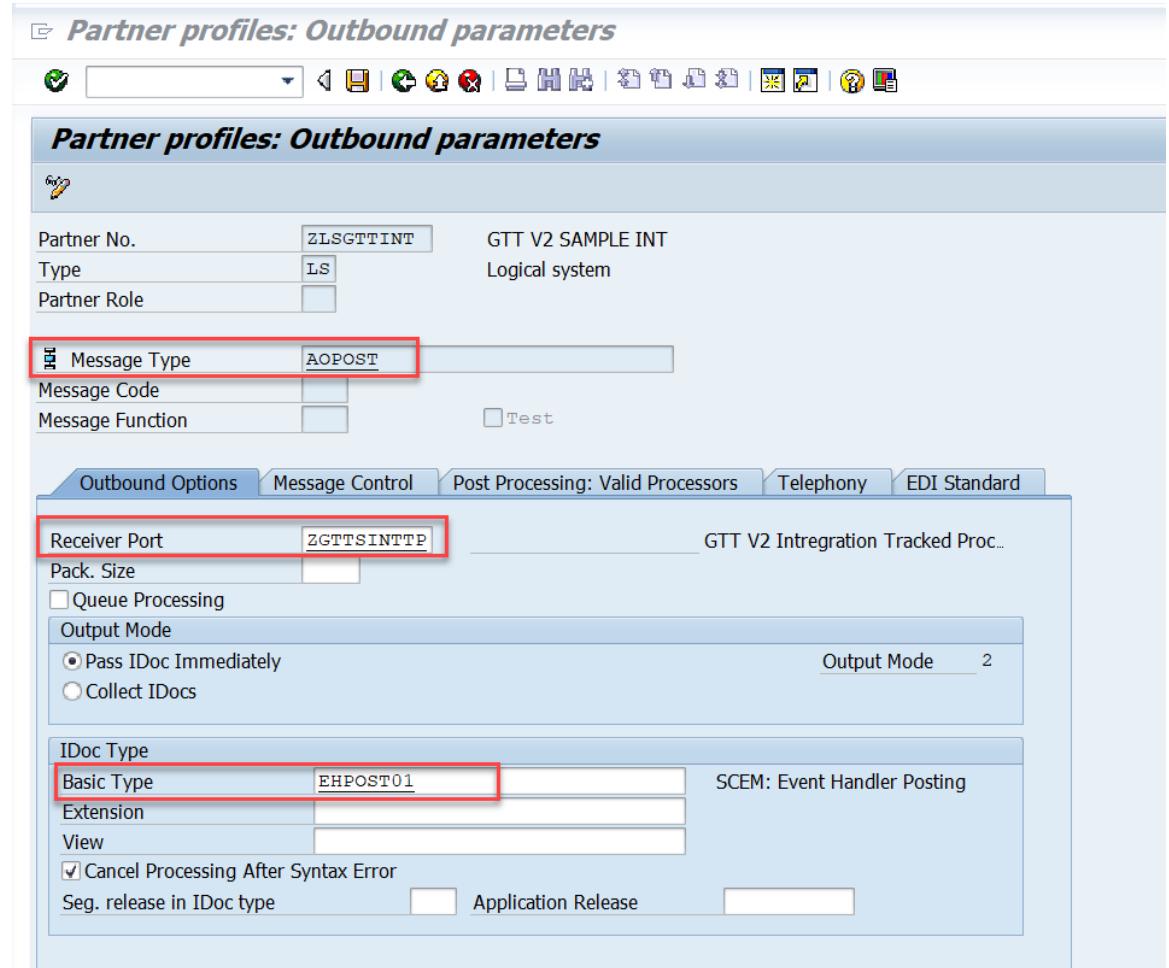
Message Code
Message Function Test

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

Receiver Port ZGTTTSINTTP ZGTTTSINTTP
GTT V2 Intregation Tracked Proc...

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

IDoc Type
Basic Type EHPOST01 EHPOST01 SCEM: Event Handler Posting
Extension
View
 Cancel Processing After Syntax Error
Seg. release in IDoc type Application Release



B) Configuration and Implementation

- Basic

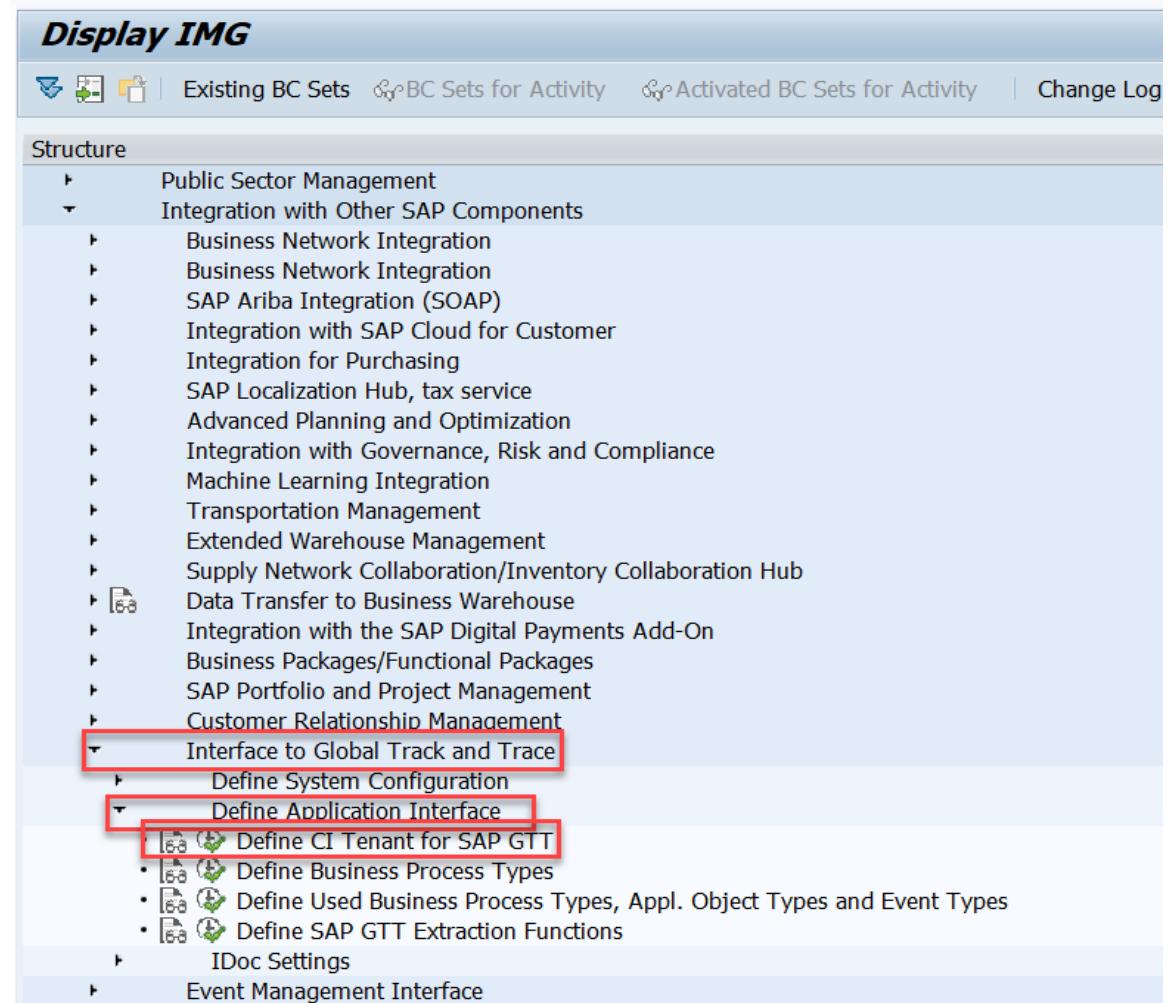
B2. Extractor Configuration



STEP 5: Define CI Tenant for GTT

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

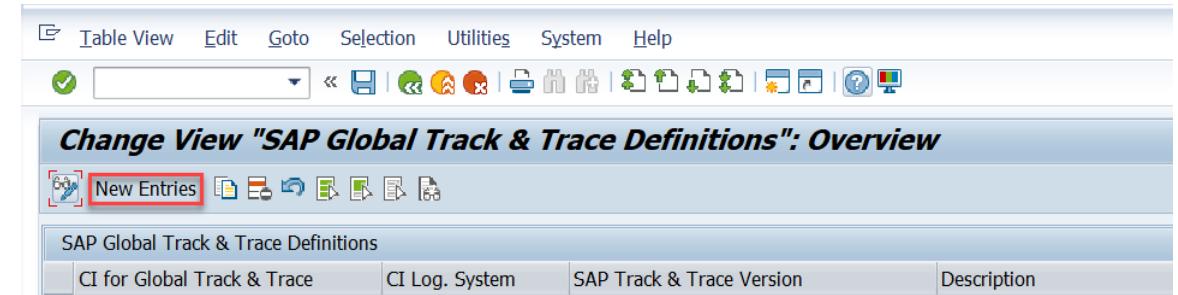
5-2: Choose activity
Define CI Tenant for SAP GTT



STEP 5: Define CI Tenant for GTT

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

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



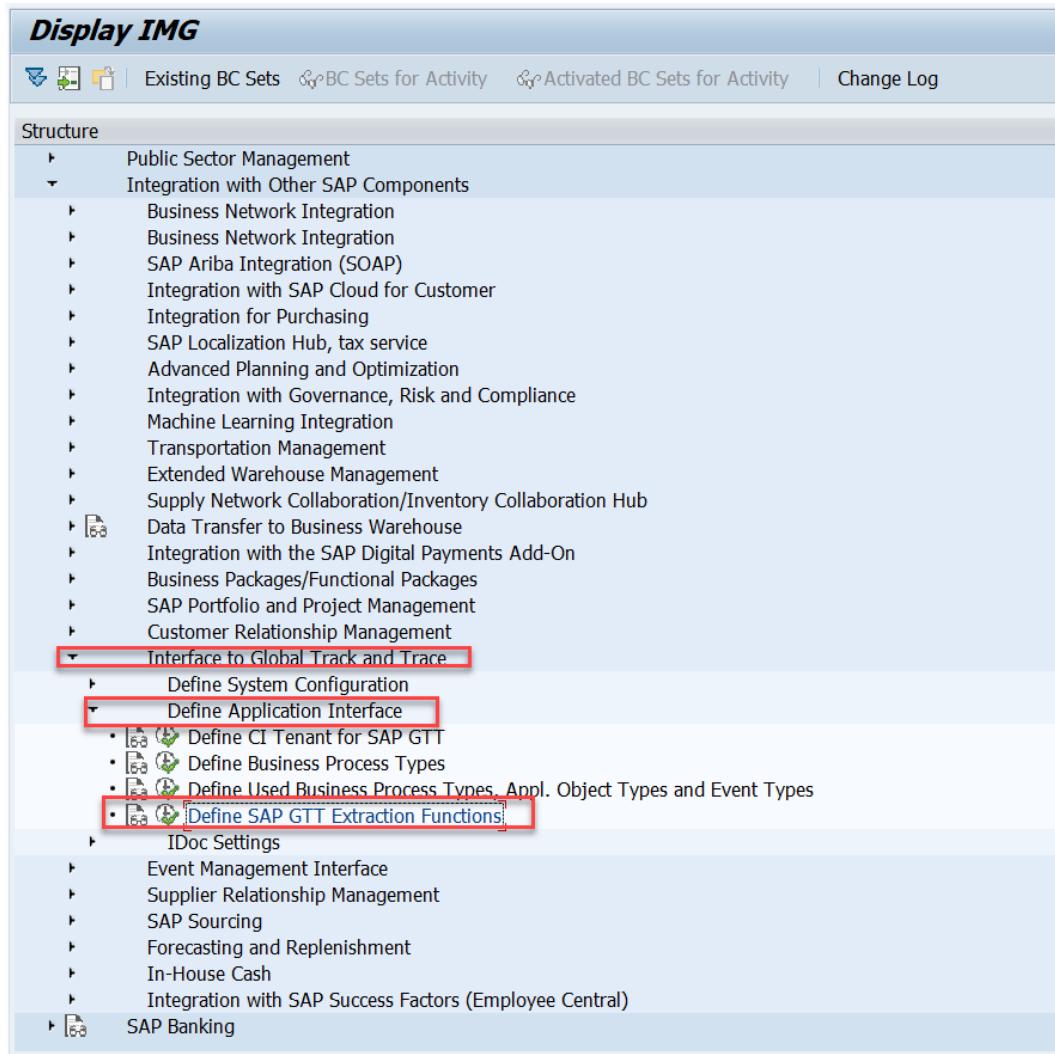
The screenshot shows the SAP GUI interface with the title bar "Display View 'SAP Global Track & Trace Definitions': Overview". Below the title bar is a toolbar with various icons. The main area displays a table titled "SAP Global Track & Trace Definitions" with four columns: CI for Global Track & Trace,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

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

STEP 6: Define GTT Extraction Functions

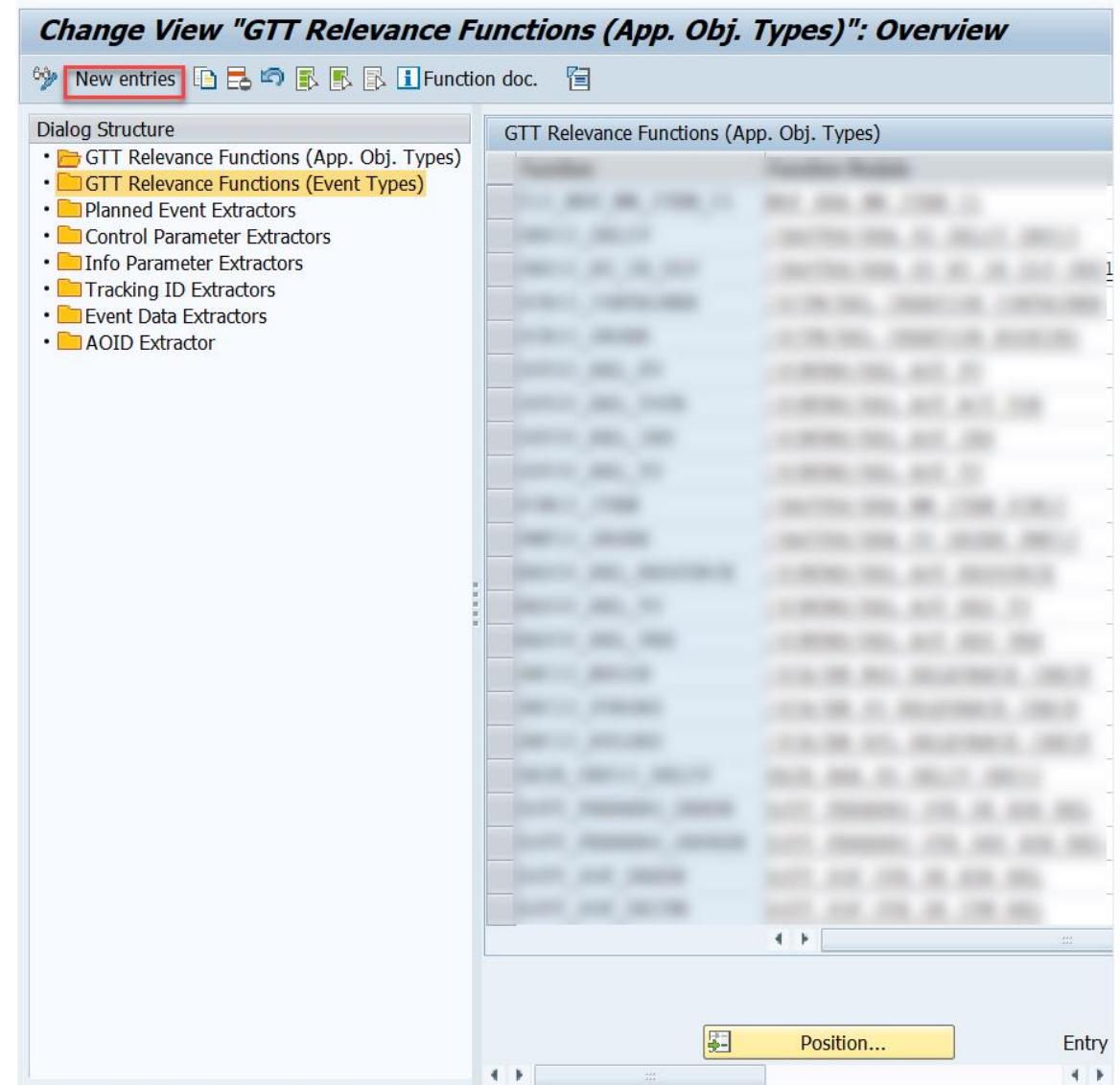
6-1: In **Display IMG** page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

6-2: Choose activity
Define SAP GTT Extraction Functions



STEP 6: Define GTT Extraction Functions

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



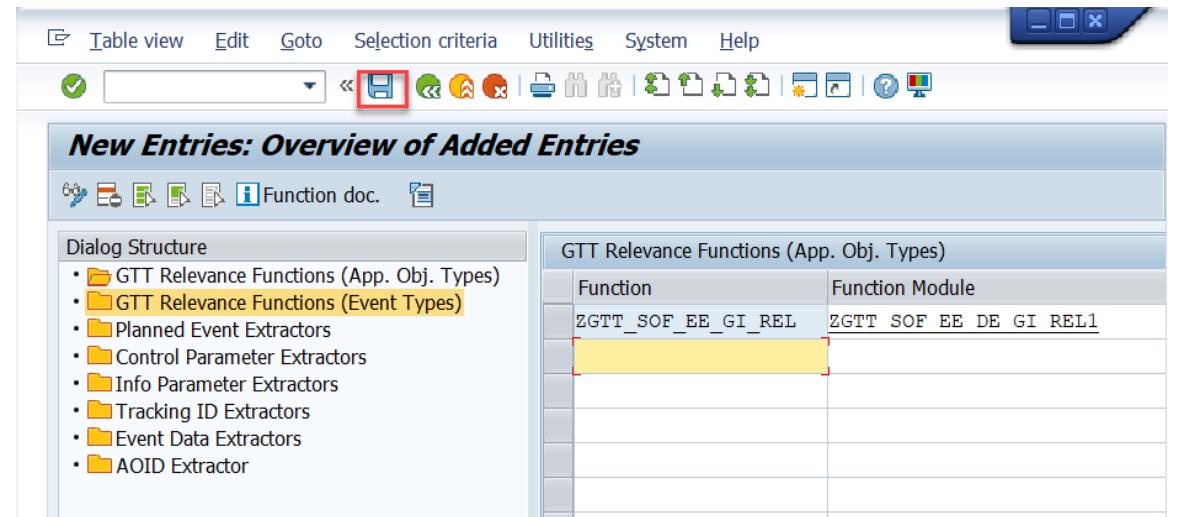
STEP 6: Define GTT Extraction Functions

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

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

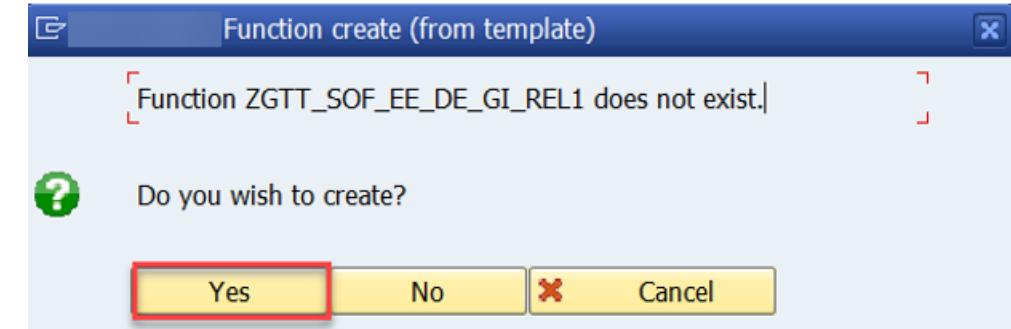
STEP 6: Define GTT Extraction Functions

6-5: Click Save



STEP 6: Define GTT Extraction Functions

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



STEP 6: Define GTT Extraction Functions

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

6-8: Click **Copy**



STEP 6: Define GTT Extraction Functions

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

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

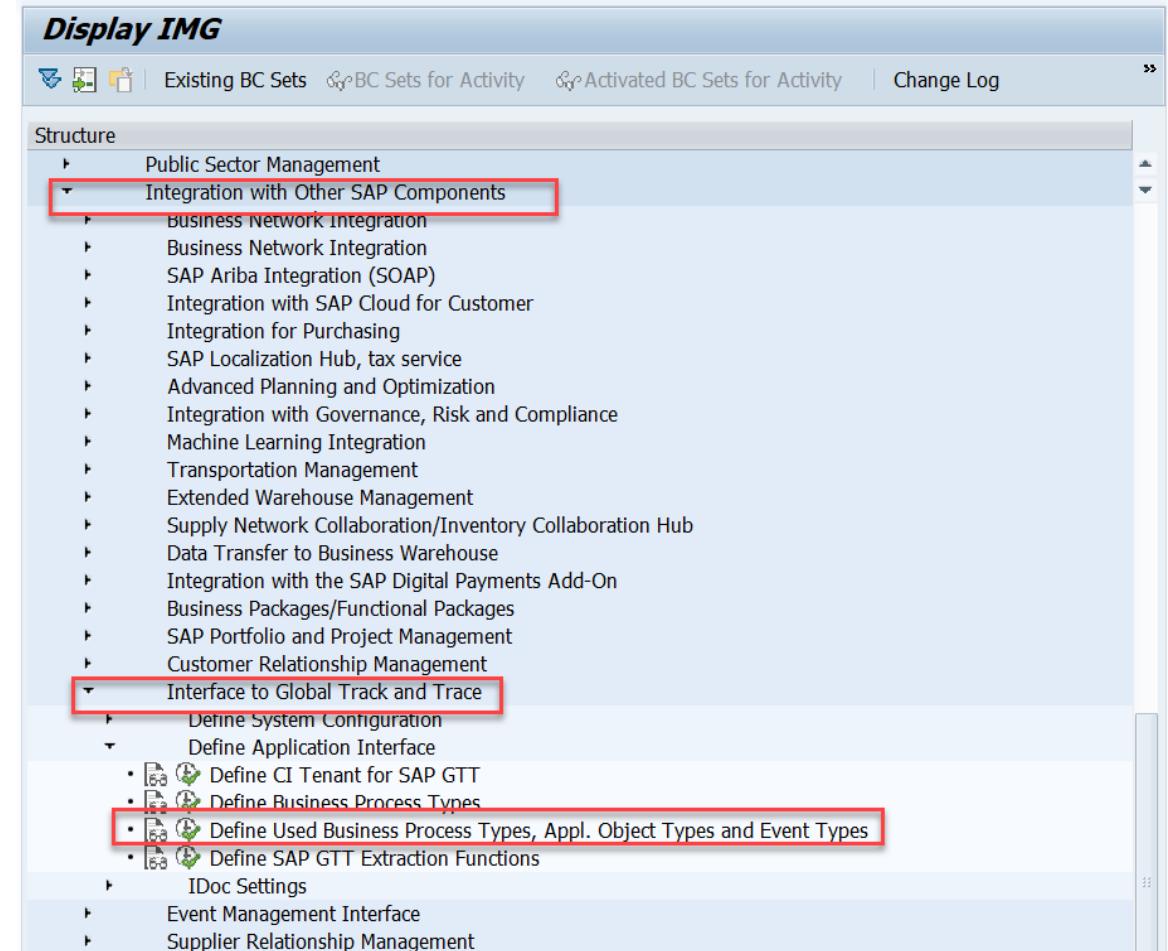
The screenshot shows the SAP Function Builder interface. In the top navigation bar, the title is "Function Builder: Display ZGTT_SOEE_DE_GI_REL1". Below the title, there are tabs for "Attributes", "Import", "Export", "Changing", "Tables", "Exceptions", and "Source Code". The "Source Code" tab is selected, displaying the ABAP code for the function module. The code defines a function named ZGTT_SOEE_DE_GI_REL1 with various parameters and return types. It also includes sections for IMPORTING, EXPORTING, TABLES, and EXCEPTIONS. The code is annotated with SAP comments starting with four asterisks (*"). The left side of the screen shows a "Repository Browser" pane with a "Function Group" dropdown set to "ZGTT_SOEE" and a list of function modules under "ZGTT_SOEE". One specific function module, "ZGTT_SOEE_DE_GI_REL1", is highlighted with a red box. The bottom right corner of the interface shows the text "ABAP" and "Ln 13 Col 48".

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

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

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

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



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

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

In the following:

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

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

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

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

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

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

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

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

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

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

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

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

7-8: Check **GTT Relevant**

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

General Data Control Tables Global Track & Trace Relevance

Sequencing / Destination	
Seq. No.	10
HCI for GTT	ZGTTSOFIN CI For GTT V2 Integration system Sales Order Sa

Data Setup	
Event Function	ZGTT_SOF_EVNT_CHIN

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

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

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

Caution:

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

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

Top Screenshot (Business Process Type: ESC_SHIPMT):

- General Data:**
 - Bus. Proc. Type: ESC_SHIPMT
 - Event Type: ZGTT_SOF_CHIN_INT
 - Text: Check In Event
- Data Source for Events:**
 - Main Obj. Table: SHIPMENT_HEADER_NEW
 - Master Table: (empty)
 - Old Main Obj. Table: SHIPMENT_HEADER_OLD
 - Old Master Table: (empty)
- Reference Between Main and Master Table:**
 - First Field Reference from Main to Master Table: (empty)
 - Second Field Reference from Main to Master Table: (empty)

A red box highlights the "SHIPMENT_HEADER_NEW" entry in the Main Obj. Table field, with the annotation "Event on Header level" pointing to it.

Bottom Screenshot (Business Process Type: ESC_DELIV):

- General Data:**
 - Bus. Proc. Type: ESC_DELIV
 - Event Type: ZGTT_SOF_PICKING_INT
 - Text: Picking Event
- Data Source for Events:**
 - Main Obj. Table: DELIVERY_ITEM_NEW
 - Master Table: DELIVERY_HEADER_NEW
 - Old Main Obj. Table: DELIVERY_ITEM_OLD
 - Old Master Table: DELIVERY_HEADER_OLD
- Reference Between Main and Master Table:**
 - First Field Reference from Main to Master Table:
 - Uplink Field: VBELN
 - Uplink Mode: R
 - Uplink Target Fld: VBELN
 - Uplink Const: (empty)
 - Second Field Reference from Main to Master Table:
 - Uplink Field: (empty)
 - Uplink Mode: (empty)
 - Uplink Target Fld: (empty)
 - Uplink Const: (empty)

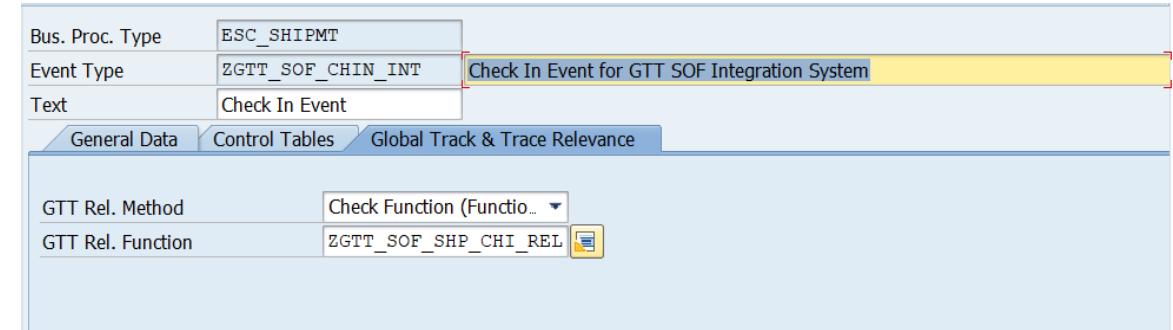
A red box highlights the "DELIVERY_ITEM_NEW" and "DELIVERY_HEADER_NEW" entries in the Main Obj. Table field, with the annotation "Event on Item level" pointing to them.

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

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

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

Click **Save**.



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

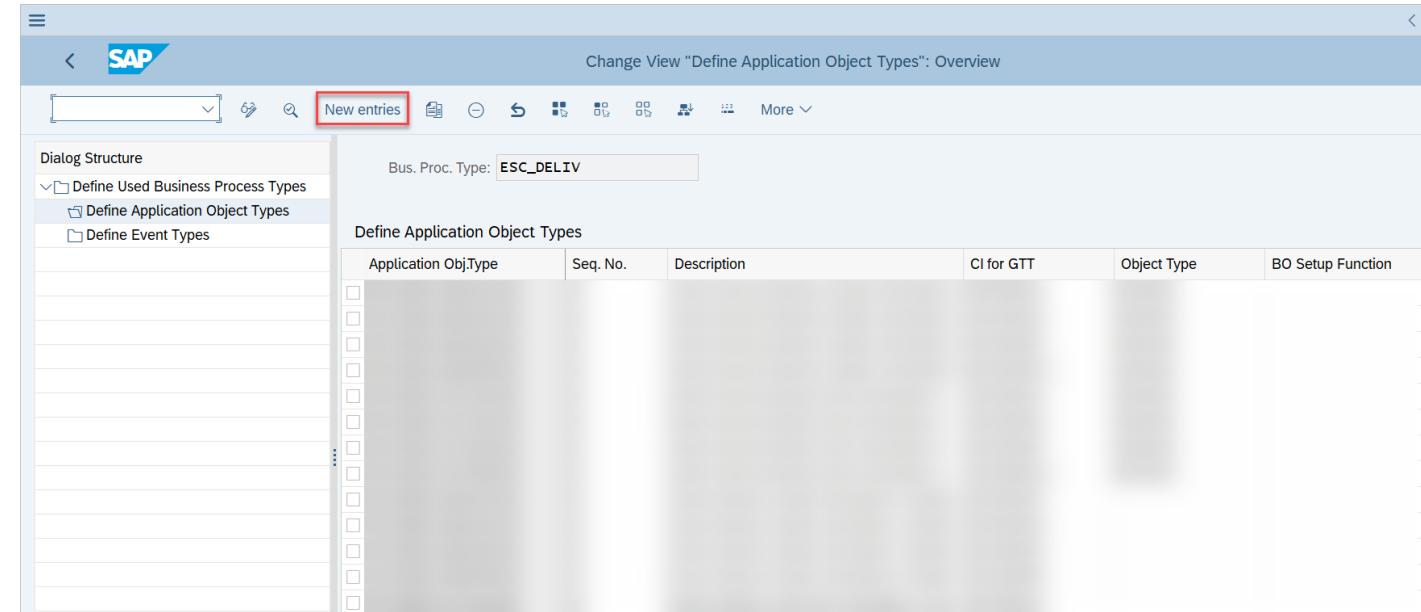
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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

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



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

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

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

7-16: Check **GTT Relevant**

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

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

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

The General Data tab contains several sections:

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

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

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

Caution:

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

The screenshot shows the SAP Fiori interface for defining business process types, application object types, and event types. It is divided into two main sections: ESC_SHIPMT (top) and ESC_DELIV (bottom).

ESC_SHIPMT Configuration:

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

ESC_DELIV Configuration:

- General Data:** Bus. Proc. Type: ESC_DELIV; Appl. Obj. Type: ZGTT_DE_INT_ITEM; Text: Extract delivery order item information to Global Track and Trace Integration.
- Control Tables:** Main Obj. Table: DELIVERY_ITEM_NEW (highlighted with a red box); Master Table: DELIVERY_HEADER_NEW (highlighted with a red box). AOT on Item Level is checked.
- Data Source for Deleted Objects:** Del.Obj. Table: DELIVERY_ITEM_OLD (highlighted with a red box).
- Reference Between Main and Master Table:** First Field Reference from Main to Master Table. Uplink Field: VBELN; Uplink Mode: R; Uplink Target Fld: VBELN; Uplink Const: (empty).

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

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

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

The screenshot shows the SAP Fiori interface for defining business process types, application object types, and event types. The top section displays basic configuration details:

Bus. Proc. Type:	ESC_DELIV	
Appl. Obj. Type:	ZGTT_DE_INT_ITEM	Extract delivery order item information to Global Track and Trace Integration
Text:	Delivery Item	

The interface includes tabs for General Data, Control Tables, Object Identification (which is selected), Global Track & Trace Relevance, and Parameter Setup. In the Object Identification tab, the 'Method for determination of AOID' section shows 'AOID Method: Determine from Field' selected. Below this, the 'Application Object ID Source' section is configured for two fields:

- First Field to Build Appl. Obj. ID:
 - Cntrl Tab. Type: 1 Main Object Table
 - AO ID Field: VBELN
- Second Field to Build Appl. Obj. ID:
 - Cntrl Tab. Type: 1 Main Object Table
 - AO ID Field: POSNR

At the bottom, the 'Determine AOID By Function' section contains a text input field for 'AOID Function'.

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

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

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

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

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

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

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

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

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

Click **Save**.

The screenshot shows the SAP Fiori interface for defining business process parameters. At the top, there are fields for 'Bus. Proc. Type' (ESC_DELIV), 'Appl. Obj. Type' (ZGTT_DE_INT_ITEM) with a tooltip 'Extract delivery order item information to Global Track and Trace Integration', and 'Text' (Delivery Item). Below these are tabs for General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The Parameter Setup tab is active. Under 'Tracking ID Setup', the 'TrkID Method' dropdown is set to 'A Determine by Function', and the 'Trk.ID Function' dropdown is set to 'ZGTT_TID_DE_ITEM'. A 'Tracking ID Fld:' input field is also present. Under 'Parameter Setup', the 'Ctrl Data Function' dropdown is set to 'ZGTT_OTE_DE_ITEM', and the 'Info Data Function' and 'Planned Event Function' dropdowns are set to 'ZGTT_EE_DE_ITM'. All three function dropdowns have small edit icons next to them. The entire screenshot area is enclosed in a red border.

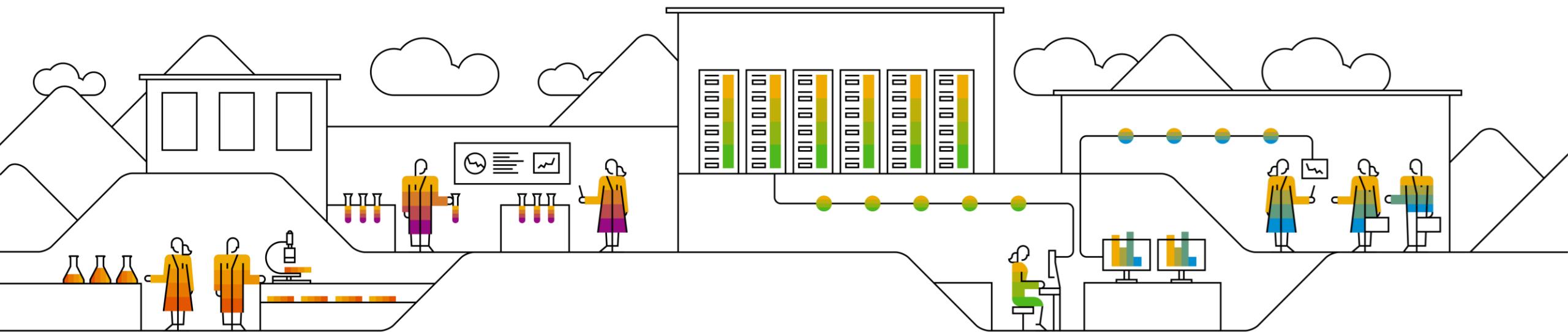
C) Download ABAP Code from GitHub

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

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

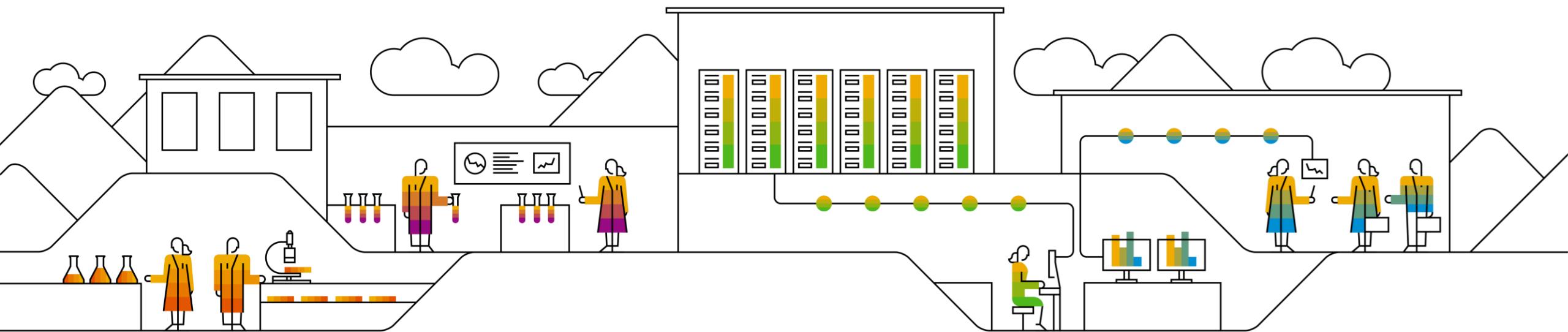
C3. Download Another ABAP Code from GitHub (Only for TPOF)

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



C) Download ABAP Code from GitHub

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



STEP 1: Install ABAPGit

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

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

 [Improve this page](#)

Summary #

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

- The standalone version is targeted at users. It 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

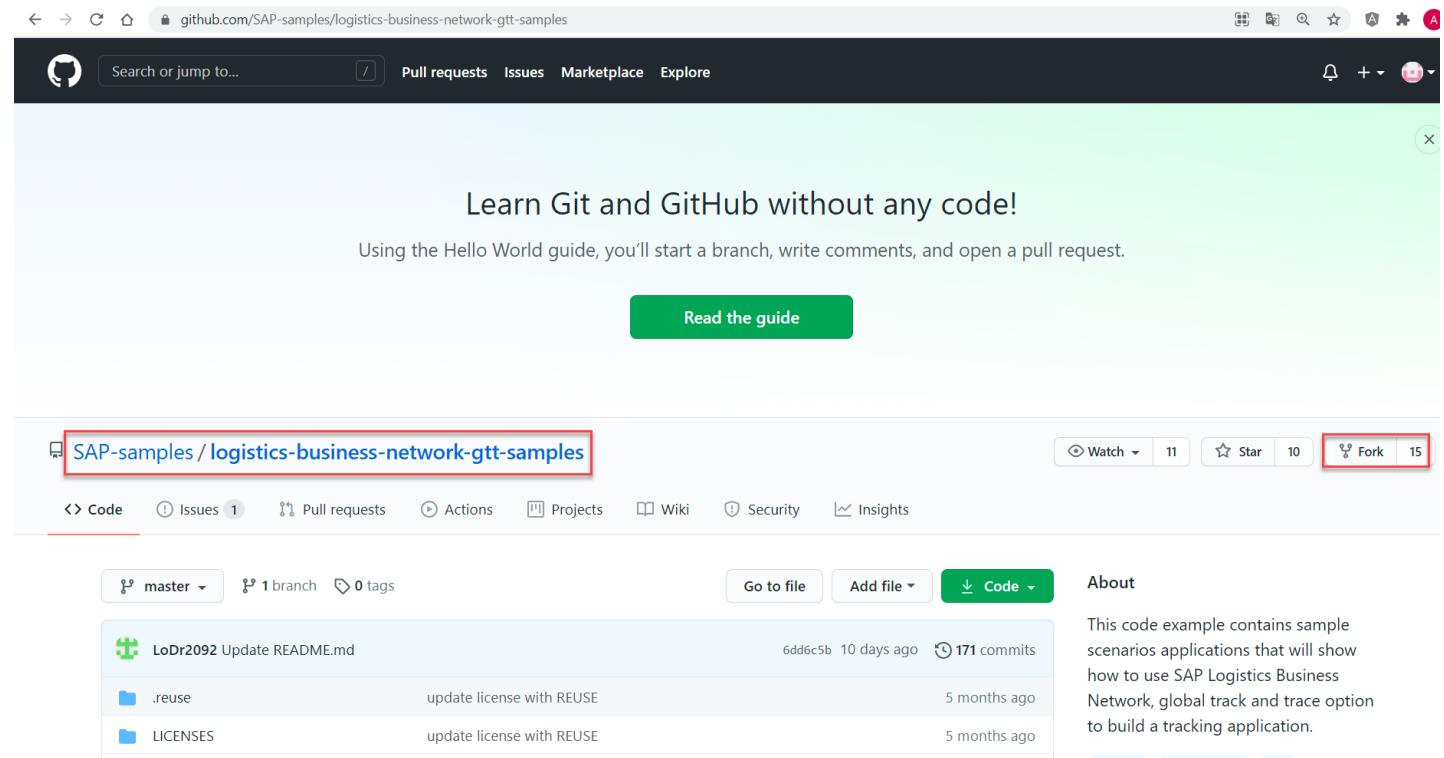
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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 the user’s own repository.



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

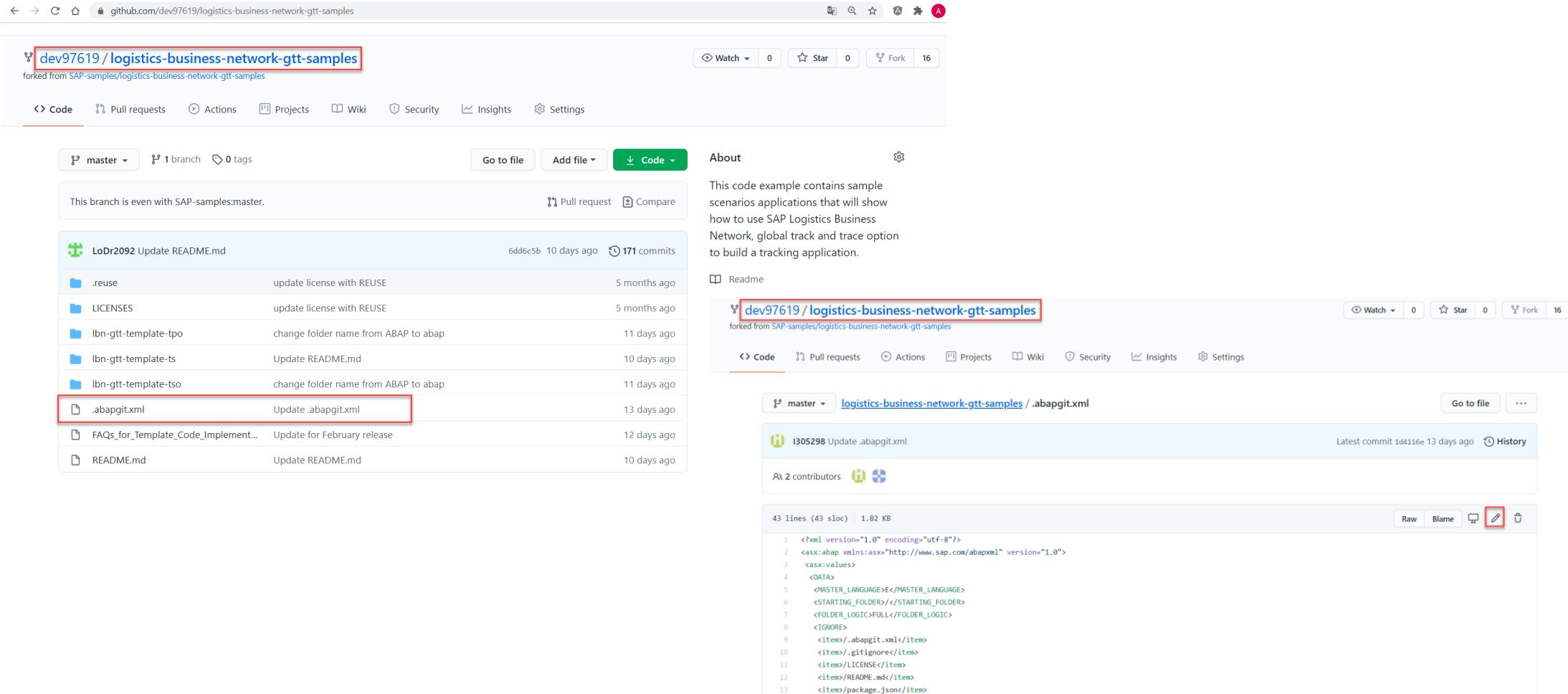
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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

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

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

3-2: Click  button to edit the file.



The screenshot shows a GitHub repository page for `dev97619/logistics-business-network-gtt-samples`. The repository has 0 stars, 16 forks, and 16 commits. The master branch has 1 branch and 0 tags. A pull request is open. The repository description states: "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." Below the description is a list of commits:

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

The commit `.abapgit.xml` is highlighted with a red box. The commit message is "Update .abapgit.xml". The file content is displayed below:

```
43 lines (43 sloc) 1.82 KB
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE></MASTER_LANGUAGE>
6       <STARTING_FOLDER></STARTING_FOLDER>
7       <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8     <IGNORE>
9       <item>/.abapgit.xml</item>
10      <item>/.gitignore</item>
11      <item>/LICENSE</item>
12      <item>/README.md</item>
13      <item>/package.json</item>
```

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

3-3: Replace the line "<STARTING_FOLDER>/</STARTING_FOLDER>" with

"<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>" as follows.

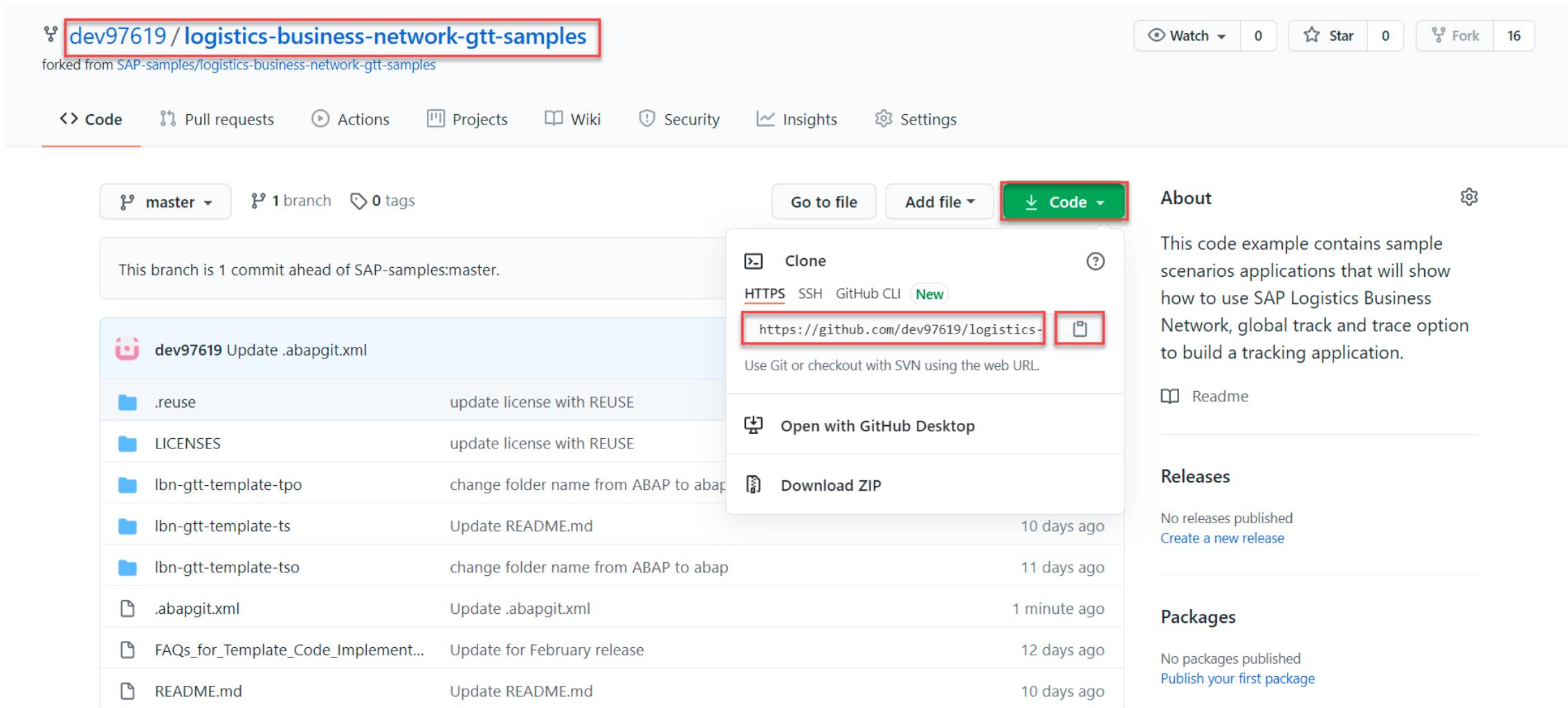
3-4: Commit change.

The screenshot shows a GitHub repository page for '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>/</STARTING_FOLDER>', is highlighted with a red box. To the right, a 'Commit changes' dialog is displayed. It contains fields for 'Update .abapgit.xml' and 'Add an optional extended description...', both currently empty. Below these fields are two radio button options: one selected for 'Commit directly to the master branch.' and another for 'Create a new branch for this commit and start a pull request.' A 'Commit changes' button is at the bottom of the dialog, and a 'Cancel' button is also present.

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

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

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



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

dev97619 / logistics-business-network-gtt-samples

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

Ibn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

FAQs_for_Template_Code_Implement... Update for February release

README.md Update README.md

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI New

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

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published Create a new release

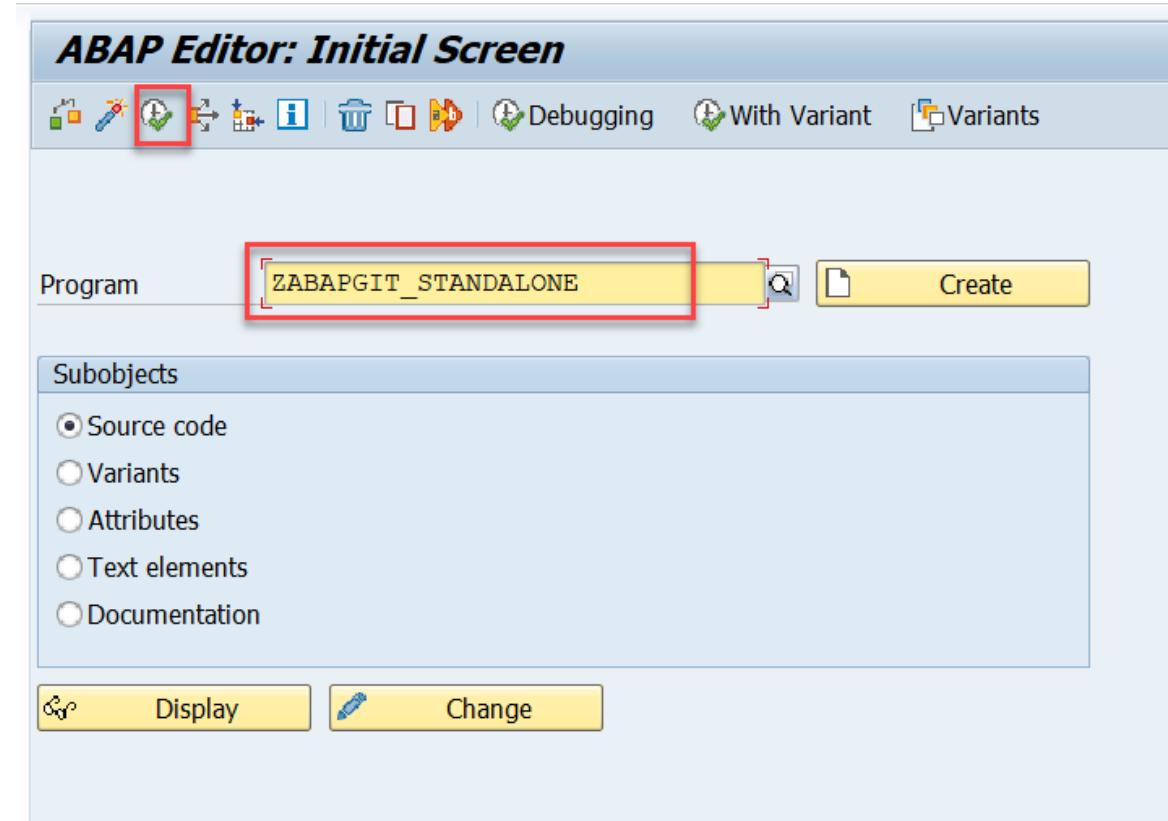
Packages

No packages published Publish your first package

STEP 4: Download ABAP Code from GitHub

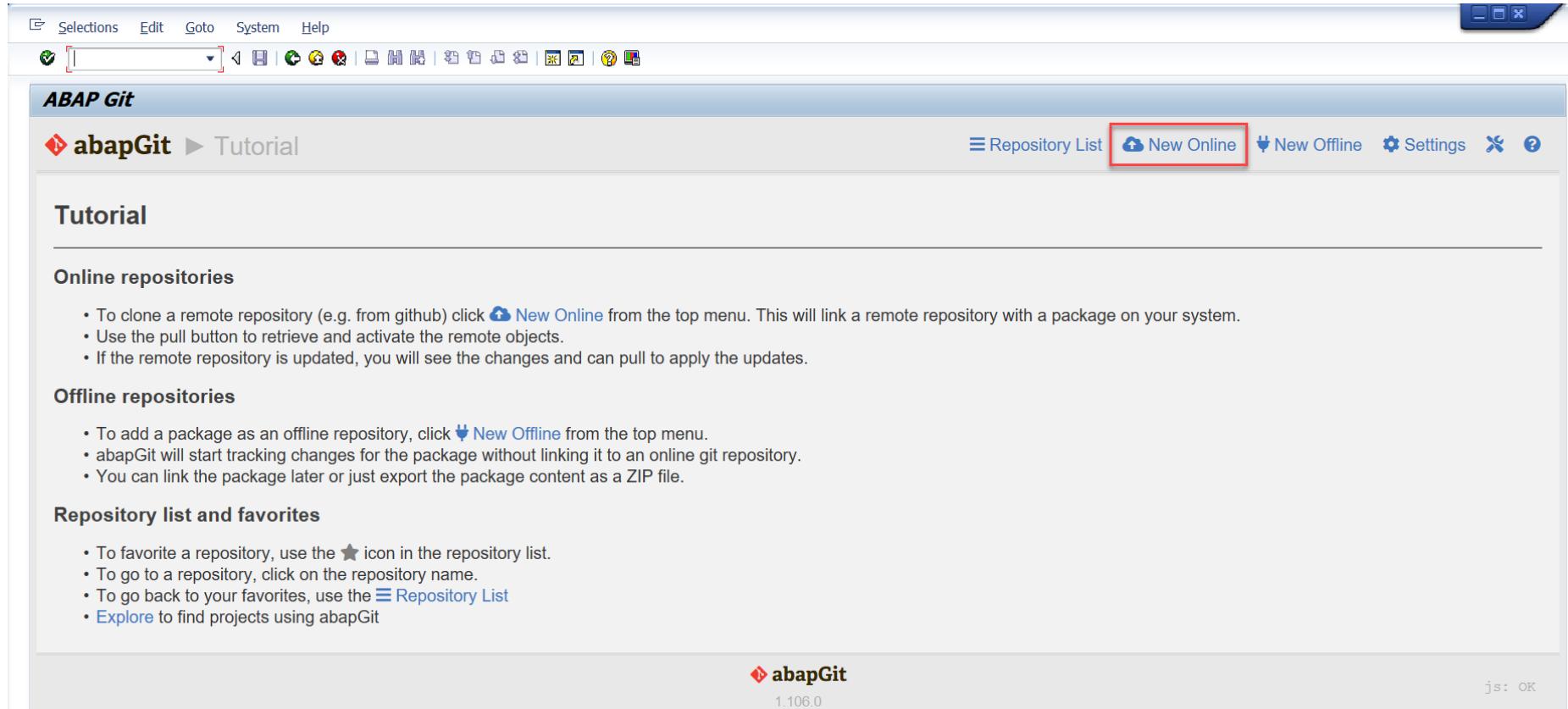
4-1: Enter T-code *SE38* and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

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

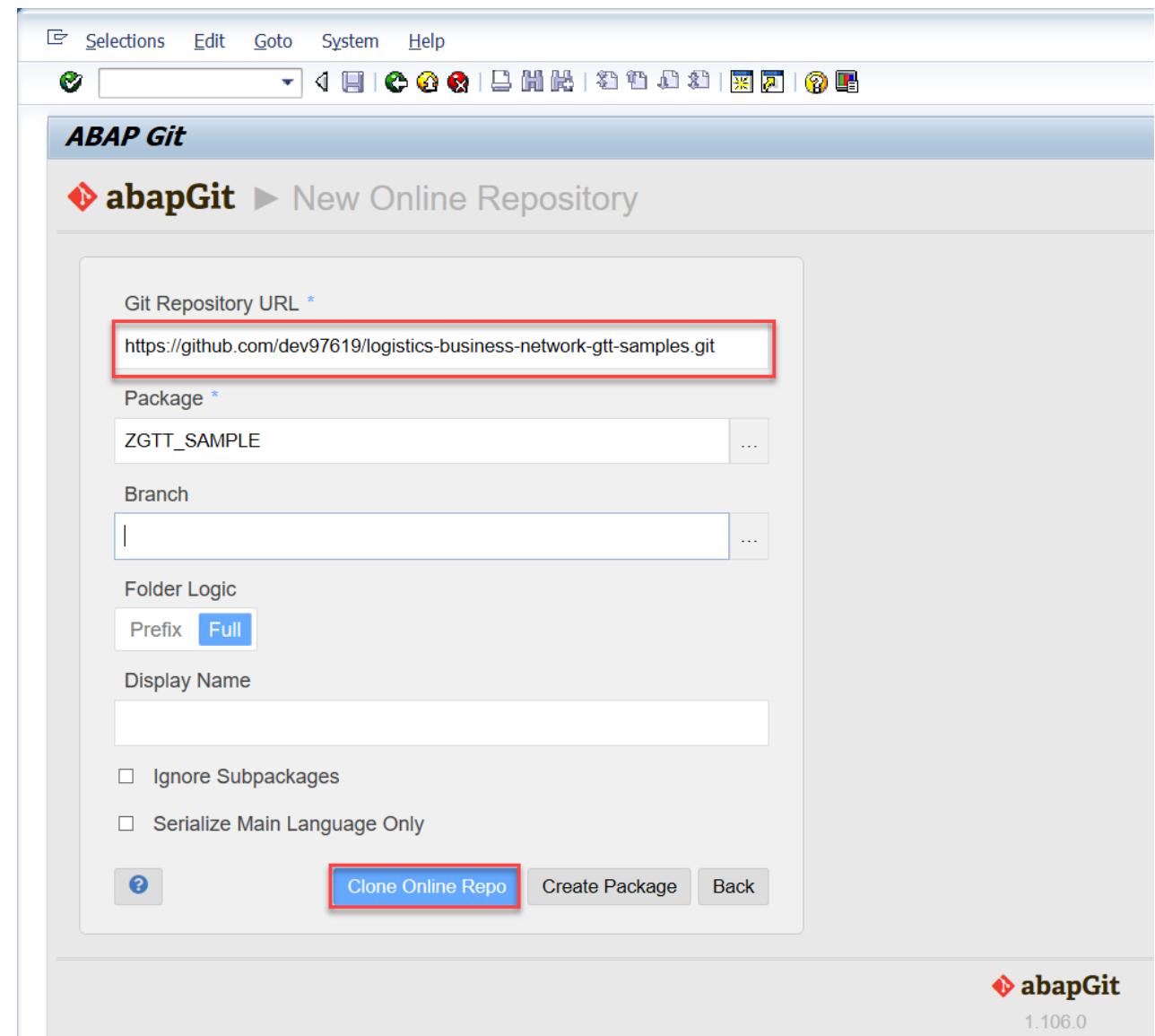
Caution:

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

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

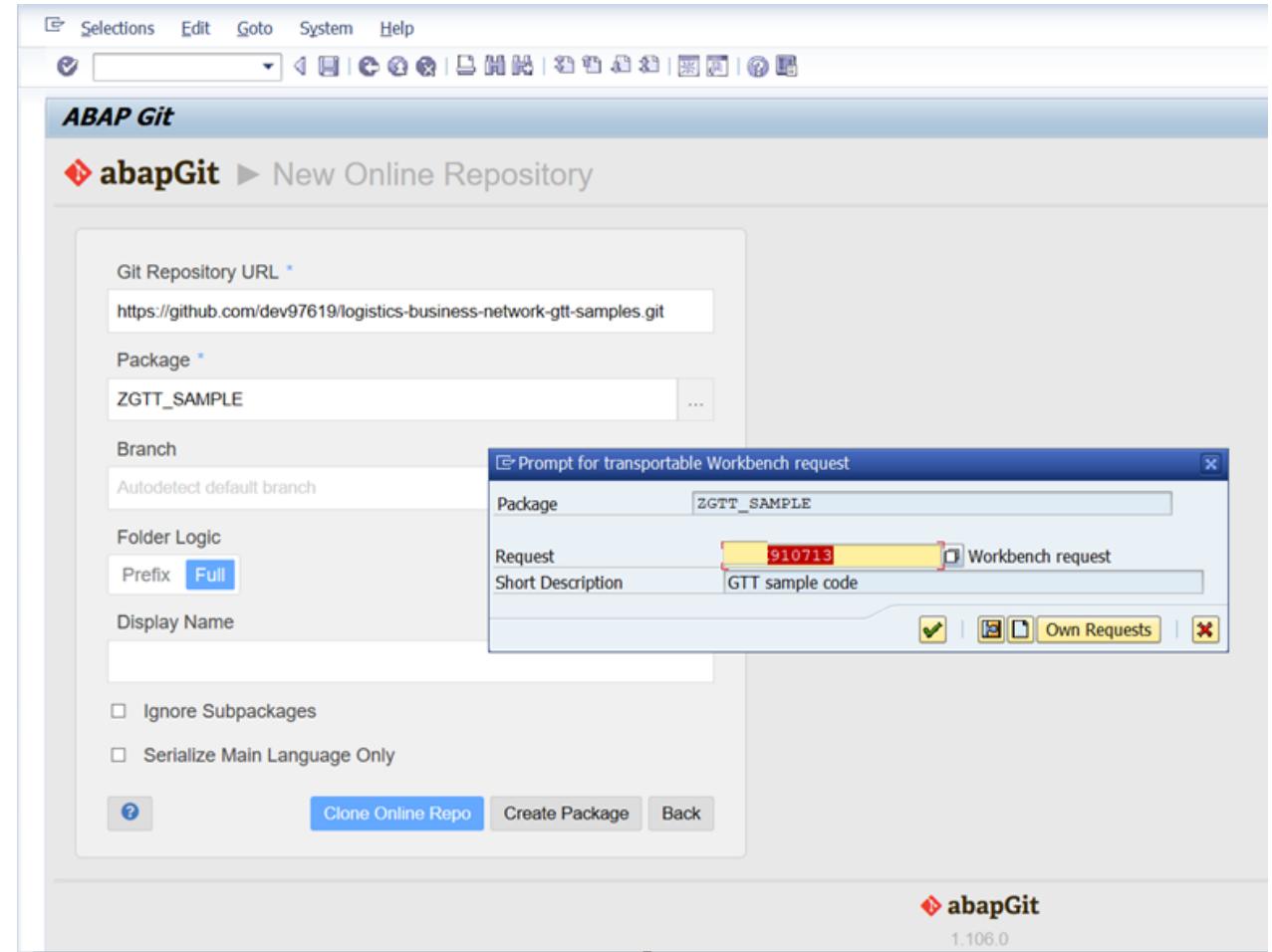
4-6: Set *Full* for **Folder Logic**

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



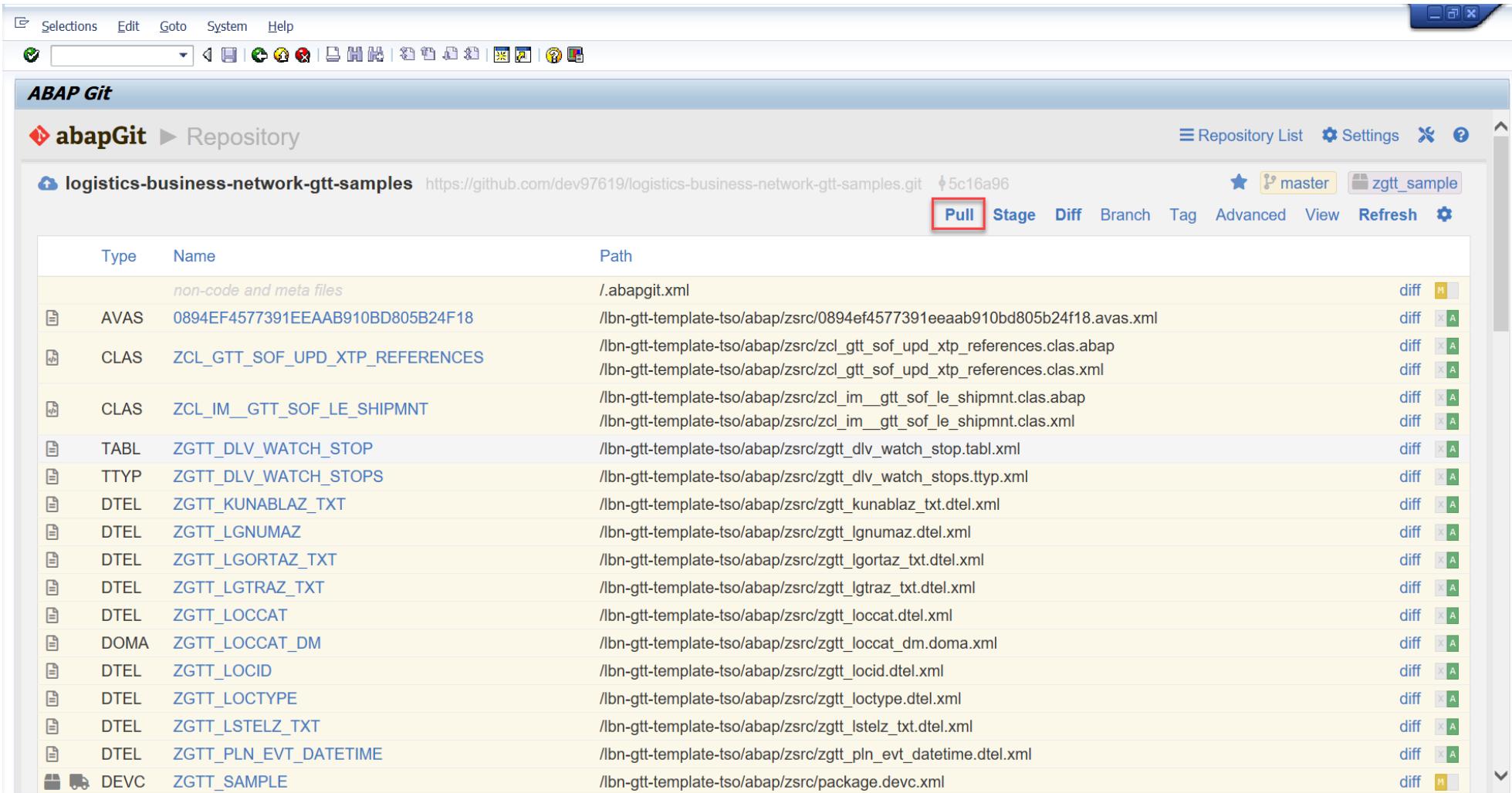
STEP 4: Download ABAP Code from GitHub

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



STEP 4: Download ABAP Code from GitHub

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

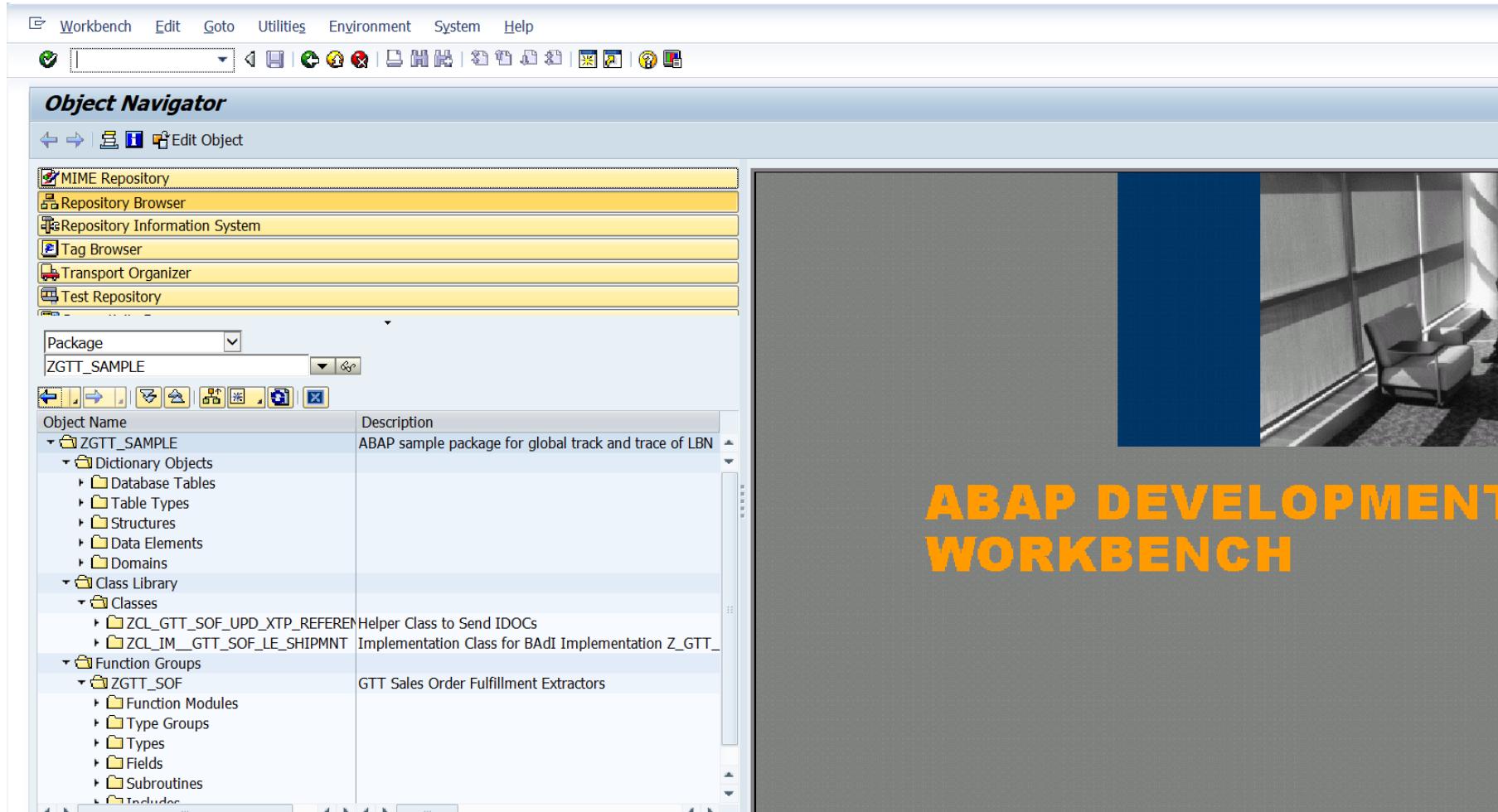


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons for file operations like Selections, Edit, Goto, System, Help, and navigation. Below the toolbar is a header bar with the title "ABAP Git" and a sub-header "abapGit ► Repository". The main area displays a table of files in the "logistics-business-network-gtt-samples" repository. The table has columns for Type, Name, and Path. The "Pull" button in the toolbar is highlighted with a red box. The repository URL is https://github.com/dev97619/logistics-business-network-gtt-samples.git and the commit hash is 5c16a96. The master branch is selected. The table lists several ABAP classes and templates, such as ZCL_GTT_SOF_UPD_XTP_REFERENCES, ZCL_IM_GTT_SOF_LE_SHIPMNT, and ZGTT_DLV_WATCH_STOP, along with their corresponding XML paths and diff links.

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

STEP 4: Download ABAP Code from GitHub

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



C) Download ABAP Code from GitHub

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

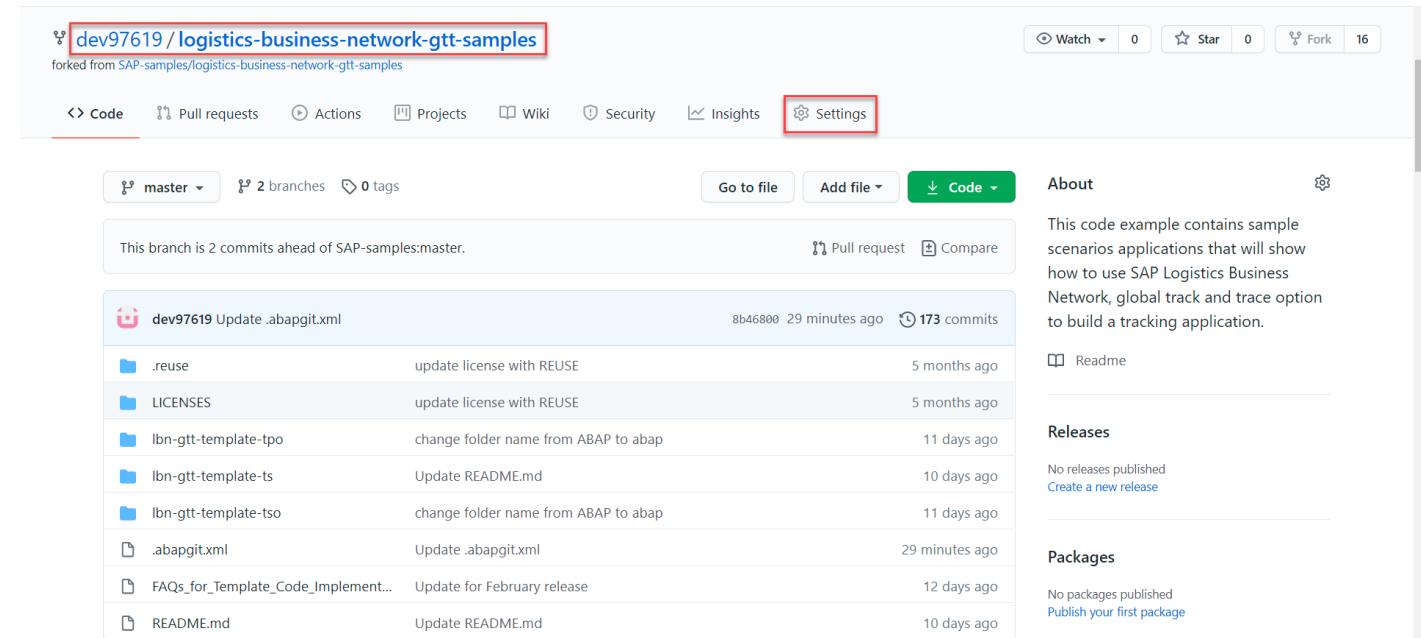


STEP 1: Delete the User's Account Repository

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

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

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



STEP 1: Delete the User's Account Repository

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

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

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

None Save

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

Choose a theme

Danger Zone

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

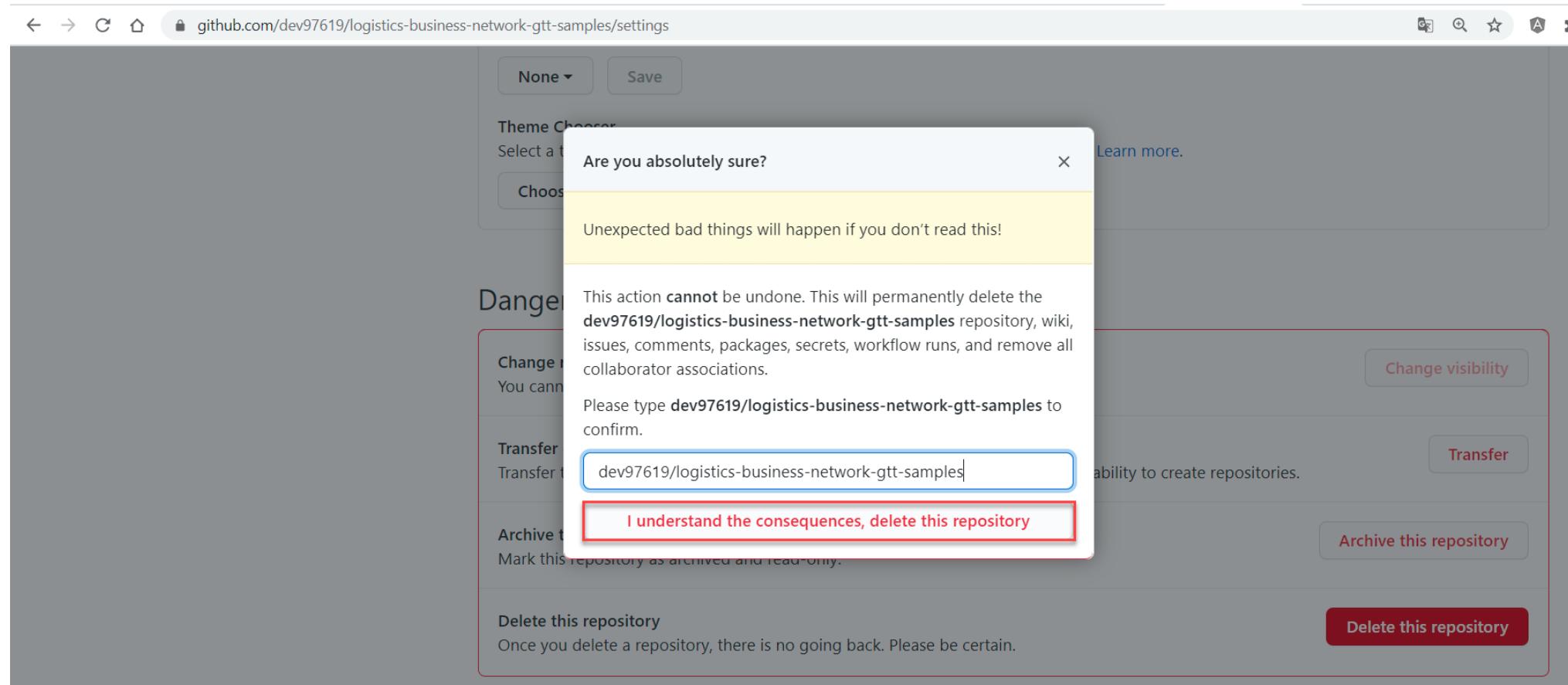
Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. Transfer

Archive this repository
Mark this repository as archived and read-only. Archive this repository

Delete this repository
Once you delete a repository, there is no going back. Please be certain. Delete this repository

STEP 1: Delete the User's Account Repository

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



STEP 1: Delete the User's Account Repository

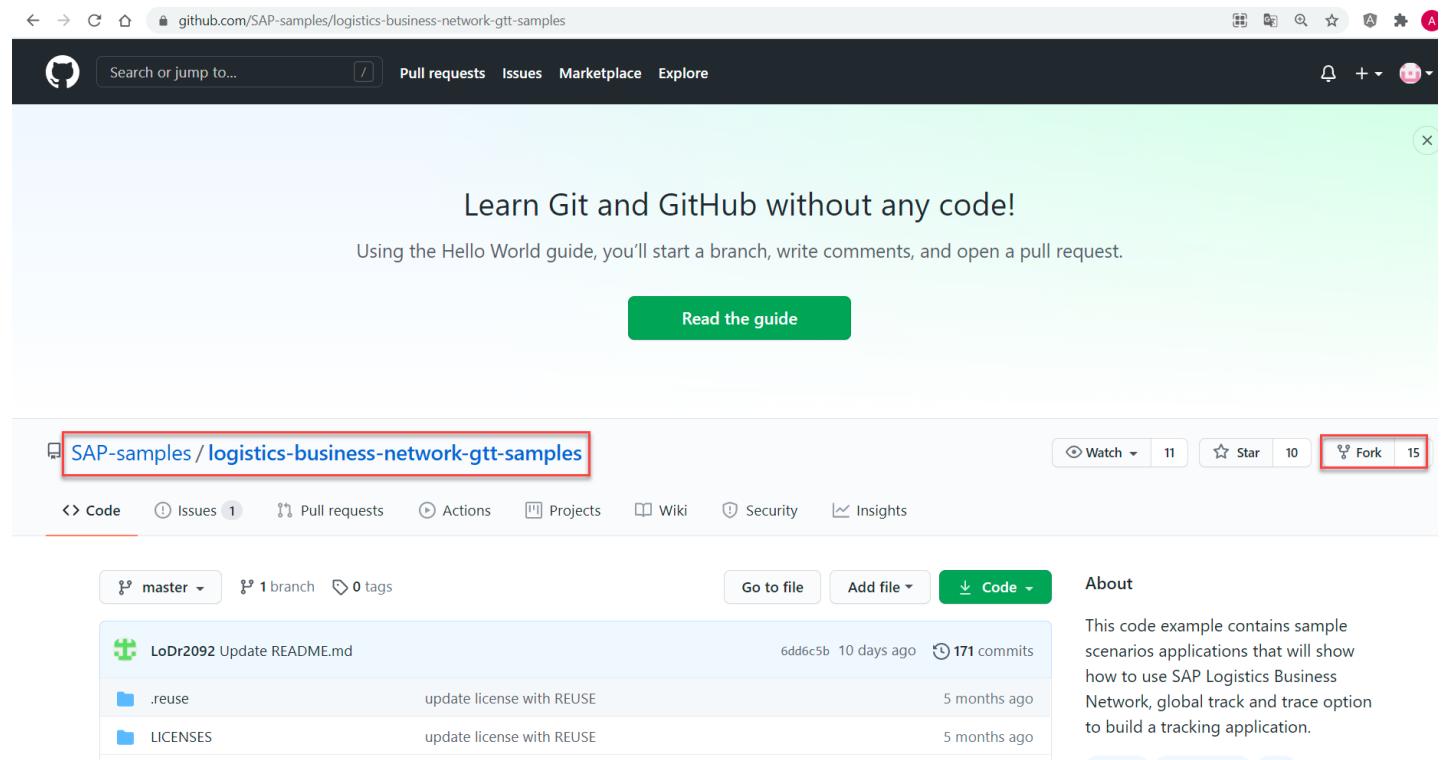
1-5: The user's account repository is deleted.

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

STEP 2: Fork Sample Code Repository

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

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



STEP 2: Fork Sample Code Repository

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

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with SAP-samples:master.

Go to file Add file ▾ Code ▾

LoDr2092 Update README.md 6dd6c5b 10 days ago 171 commits

.reuse update license with REUSE 5 months ago

LICENSES update license with REUSE 5 months ago

Ibn-gtt-template-tpo change folder name from ABAP to abap 11 days ago

Ibn-gtt-template-ts Update README.md 10 days ago

Ibn-gtt-template-tso change folder name from ABAP to abap 11 days ago

.abapgit.xml Update .abapgit.xml 13 days ago

FAQs_for_Template_Code_Implement... Update for February release 12 days ago

README.md Update README.md 10 days ago

About

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

Readme

Releases

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

Packages

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

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

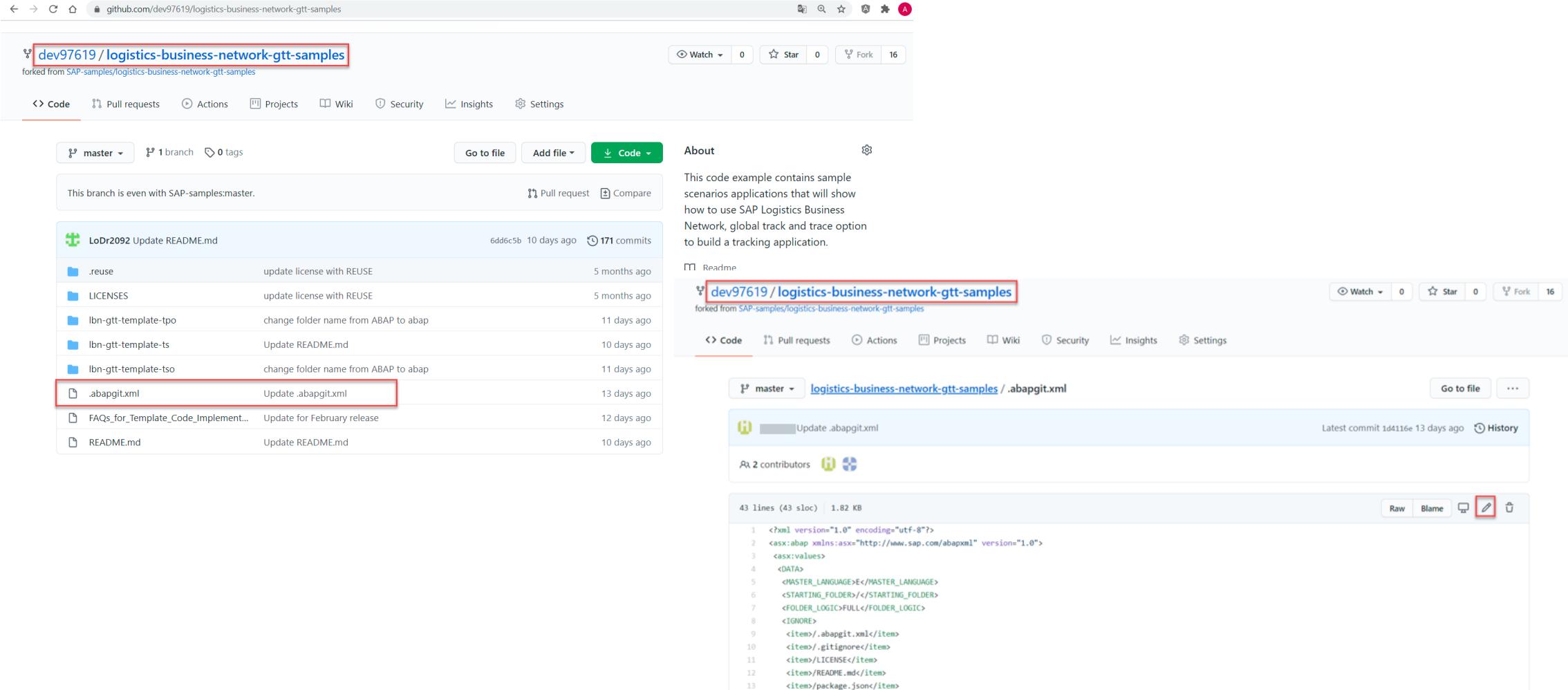
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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

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

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

3-2: Click  button to edit the file.



The screenshot shows a GitHub repository page for 'dev97619 / logistics-business-network-gtt-samples'. The repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected, showing the master branch with 1 branch and 0 tags. A commit by LoDr2092 titled 'Update README.md' is visible. Below it, a commit by the current user titled 'Update .abapgit.xml' is highlighted with a red box. The repository has 171 commits in total. The 'About' section describes the code example as containing sample scenarios for SAP Logistics Business Network, global track and trace options. The '.abapgit.xml' file content is displayed in a code editor, showing XML configuration for ABAP git. The file has 43 lines and 1.82 KB size. The code includes elements like <?xml version="1.0" encoding="utf-8"?>, <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">, and various logic and ignore definitions for file types like .abapgit.xml, .gitignore, .mdc, and LICENSE.

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

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

3-3: Replace the line "<STARTING_FOLDER>/</STARTING_FOLDER>" with

"<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>" as follows.

3-4: Commit change.

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

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

Commit changes

Update .abapgit.xml

Add an optional extended description...

-o- Commit directly to the master branch.

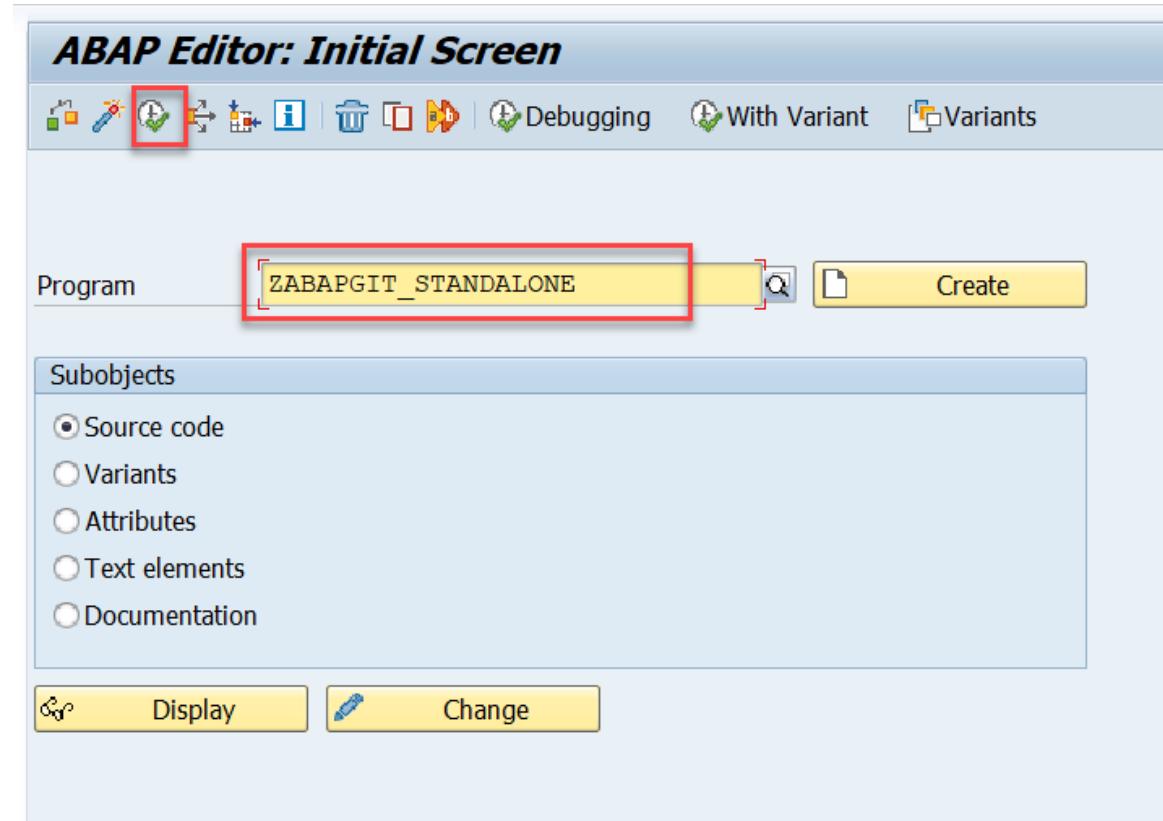
! Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit changes Cancel

STEP 4: Update ABAP Code from GitHub

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

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



STEP 4: Update ABAP Code from GitHub

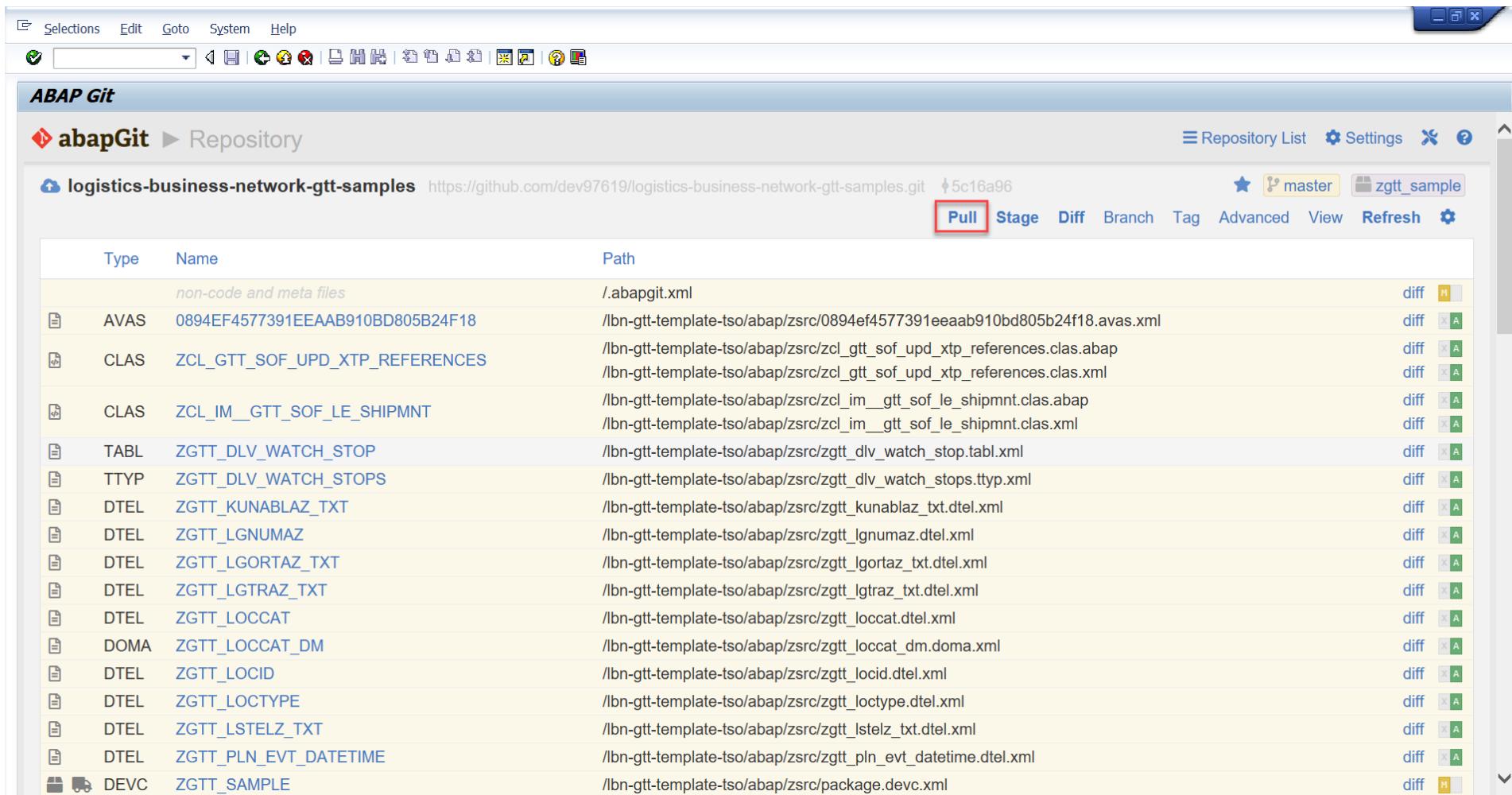
4-3: Check if the URL is changed or not after your recreation of repository copy. Access the TSOF Repository by clicking  button.



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

STEP 4: Update ABAP Code from GitHub

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



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

Type	Name	Path	diff
	non-code and meta files	/abapgit.xml	M
AVAS	0894EF4577391EEAAB910BD805B24F18	//lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	A
CLAS	ZCL_GTT_SOUPD_XTP_REFERENCES	//lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	A A
CLAS	ZCL_IM_GTT_SOFL_SHIPMNT	//lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.abap //lbn-gtt-template-tso/abap/zsrc/zcl_im_gtt_sof_le_shipmnt.clas.xml	A A
TABL	ZGTT_DLW_WATCH_STOP	//lbn-gtt-template-tso/abap/zsrc/zgtt_dlw_watch_stop.tabl.xml	A
TTYP	ZGTT_DLW_WATCH_STOPS	//lbn-gtt-template-tso/abap/zsrc/zgtt_dlw_watch_stops.ttyp.xml	A
DTEL	ZGTT_KUNABLAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_kunabla_z_txt.dtel.xml	A
DTEL	ZGTT_LGNUMAZ	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgnumaz.dtel.xml	A
DTEL	ZGTT_LGORATAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgorataz_txt.dtel.xml	A
DTEL	ZGTT_LGTRAZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lgtraz_txt.dtel.xml	A
DTEL	ZGTT_LOCCAT	//lbn-gtt-template-tso/abap/zsrc/zgtt_locat.dtel.xml	A
DOMA	ZGTT_LOCCAT_DM	//lbn-gtt-template-tso/abap/zsrc/zgtt_locat_dm.doma.xml	A
DTEL	ZGTT_LOCID	//lbn-gtt-template-tso/abap/zsrc/zgtt_locid.dtel.xml	A
DTEL	ZGTT_LOCTYPE	//lbn-gtt-template-tso/abap/zsrc/zgtt_loctype.dtel.xml	A
DTEL	ZGTT_LSTELZ_TXT	//lbn-gtt-template-tso/abap/zsrc/zgtt_lstelz_txt.dtel.xml	A
DTEL	ZGTT_PLN_EVT_DATETIME	//lbn-gtt-template-tso/abap/zsrc/zgtt_pln_evt_datetime.dtel.xml	A
DEV	ZGTT_SAMPLE	//lbn-gtt-template-tso/abap/zsrc/package.devcl.xml	M

C) Download ABAP Code from GitHub

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



STEP 1: Fork Sample Code Repository

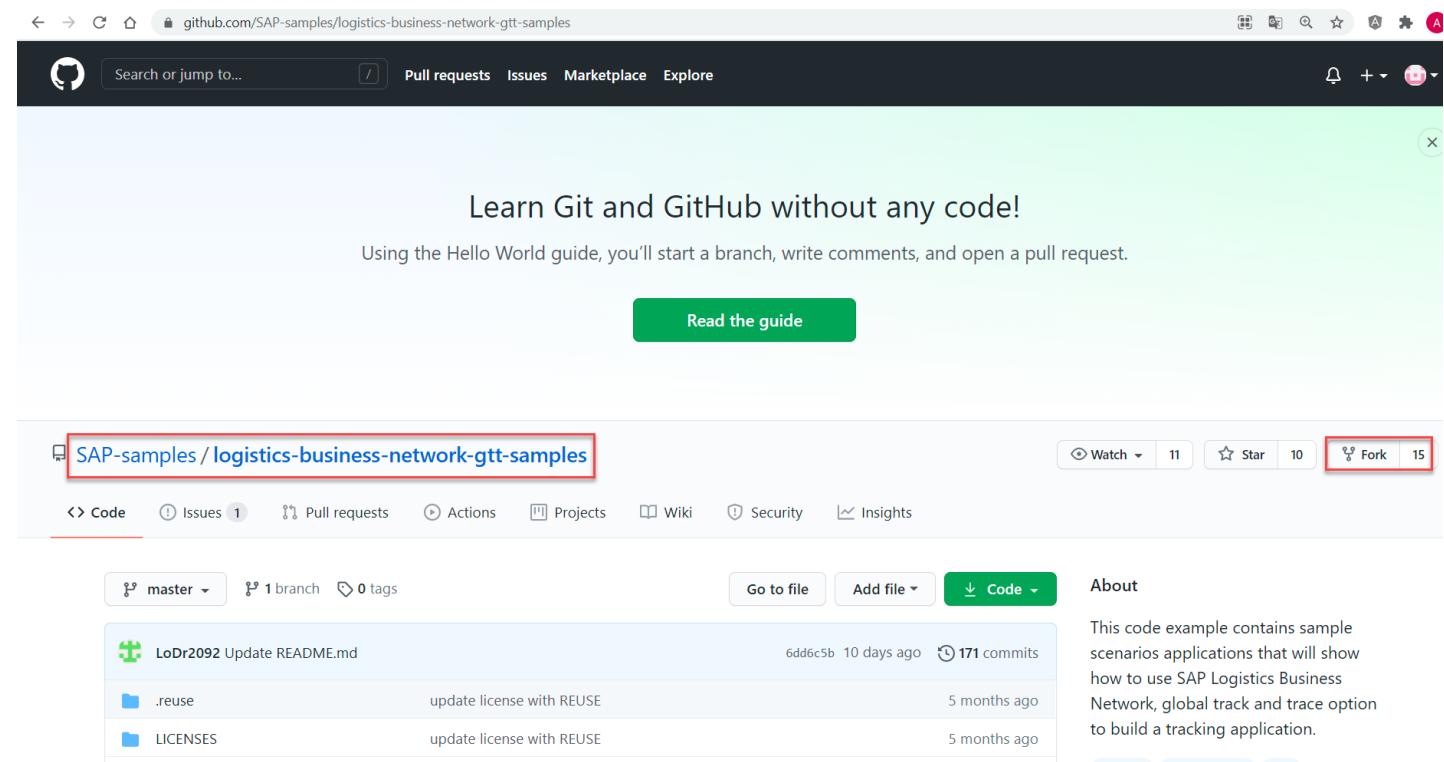
Prerequisite:

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

To install the TPOF do the following:

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

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



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

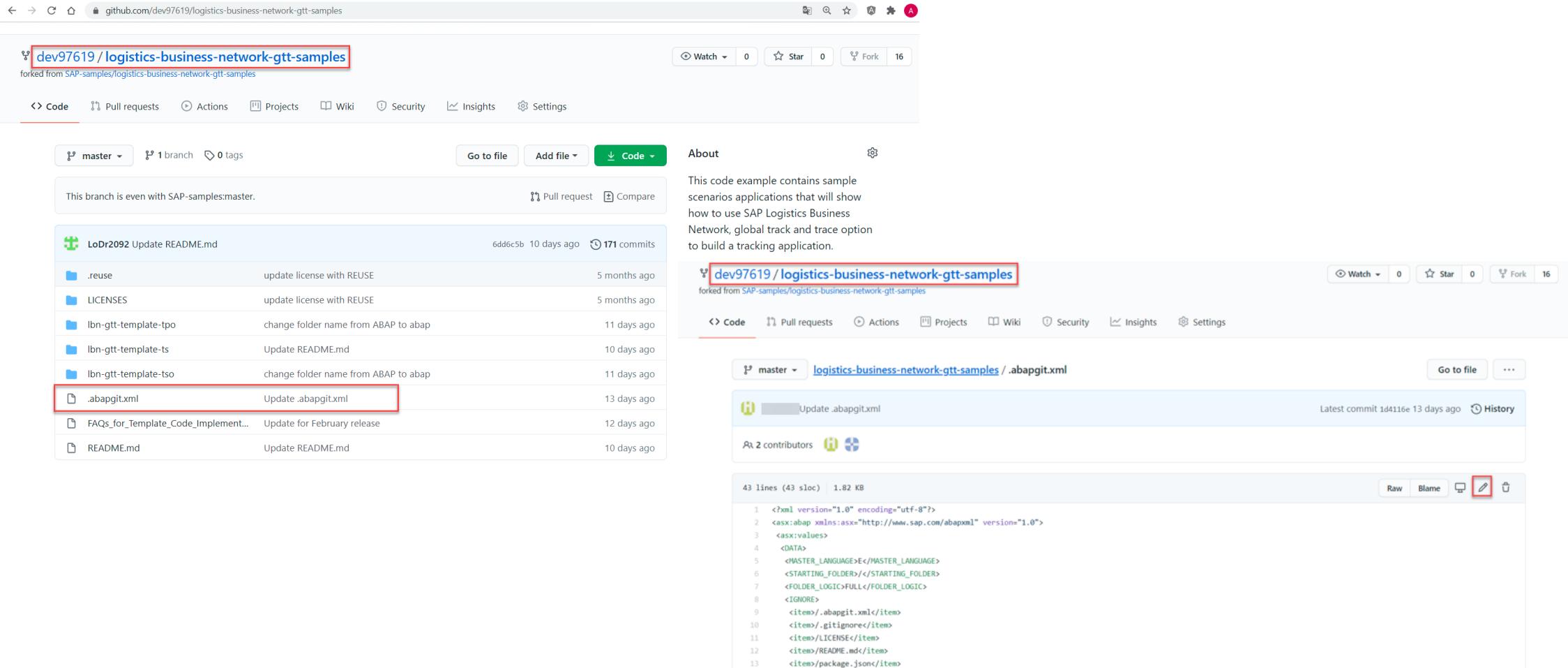
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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

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

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

2-2: Click  button to edit the file.



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

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

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

2-3: Replace the line "<STARTING_FOLDER>/</STARTING_FOLDER>" with

"<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>" as follows.

2-4: Commit change.

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

.abapgit.xml Content:

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

Commit changes:

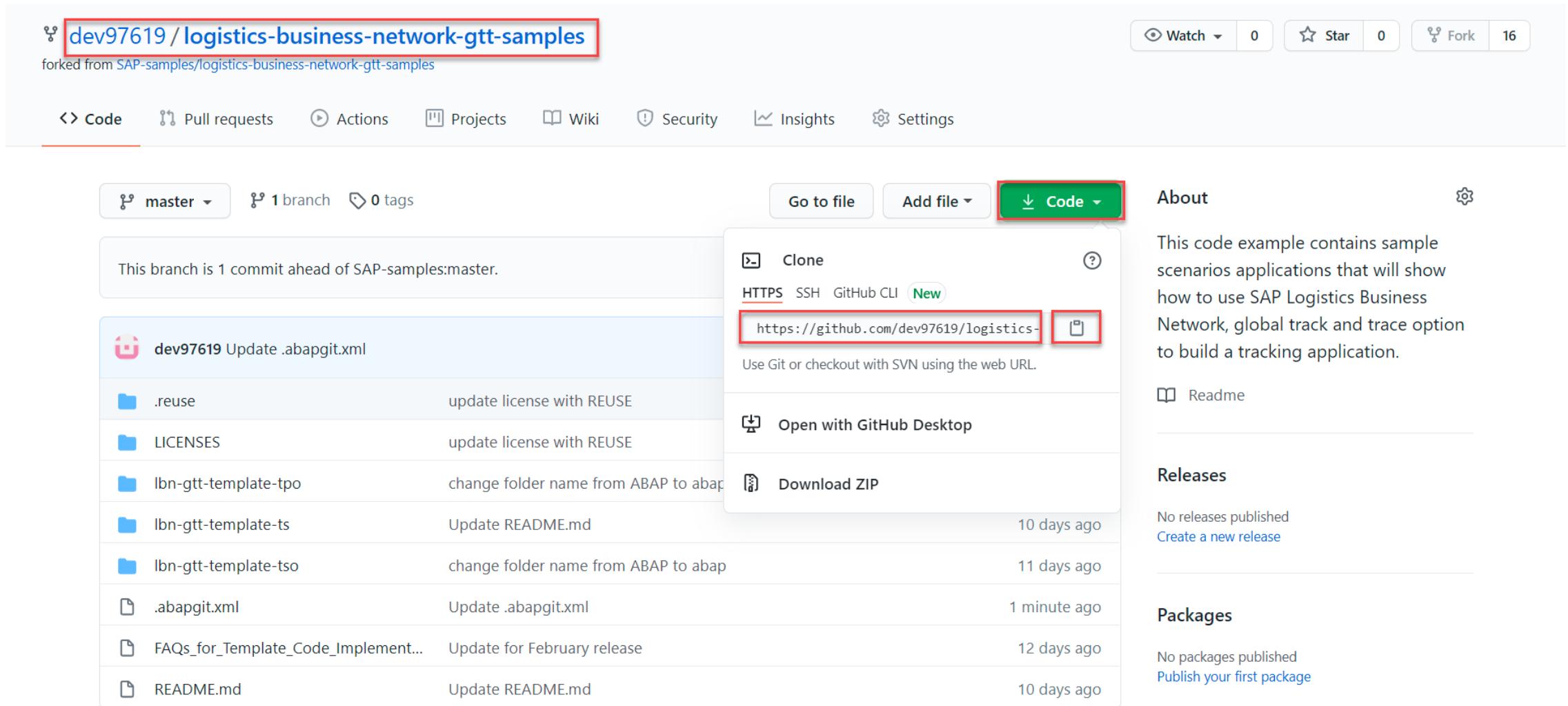
Update .abapgit.xml
Add an optional extended description...

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

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

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

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



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

dev97619 / logistics-business-network-gtt-samples

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

Watch 0 Star 0 Fork 16

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

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

dev97619 Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

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

Ibn-gtt-template-ts Update README.md

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

.abapgit.xml Update .abapgit.xml

FAQs_for_Template_Code_Implement... Update for February release

README.md Update README.md

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI New

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

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

10 days ago

11 days ago

1 minute ago

12 days ago

10 days ago

About

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

Readme

Releases

No releases published Create a new release

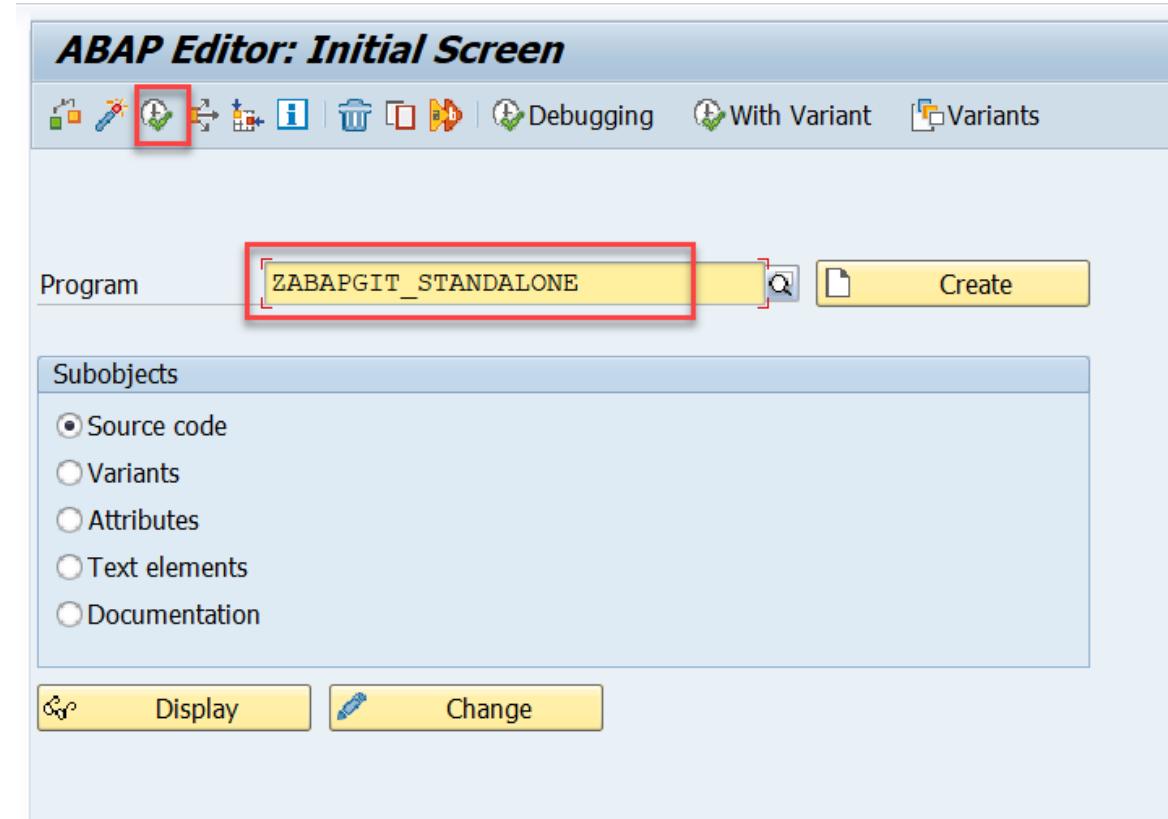
Packages

No packages published Publish your first package

STEP 3: Remove TSOF Repository in ABAPGit

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

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



STEP 3: Remove TSOF Repository in ABAPGit

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

The screenshot shows the ABAP Git interface within a SAP application. The top menu bar includes Selections, Edit, Goto, System, and Help. Below the menu is a toolbar with various icons. The main title is "ABAP Git". The sub-title is "abapGit ► Repository List". On the right side of the header are buttons for New Online, New Offline, Settings, and Help. Below the header is a search bar labeled "Filter:" and checkboxes for "Only Favorites" and "Detail". The main area displays a table with columns: Name, Url, Package, Branch, and Action. A single repository row is shown:

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

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

STEP 3: Remove TSOF Repository in ABAPGit

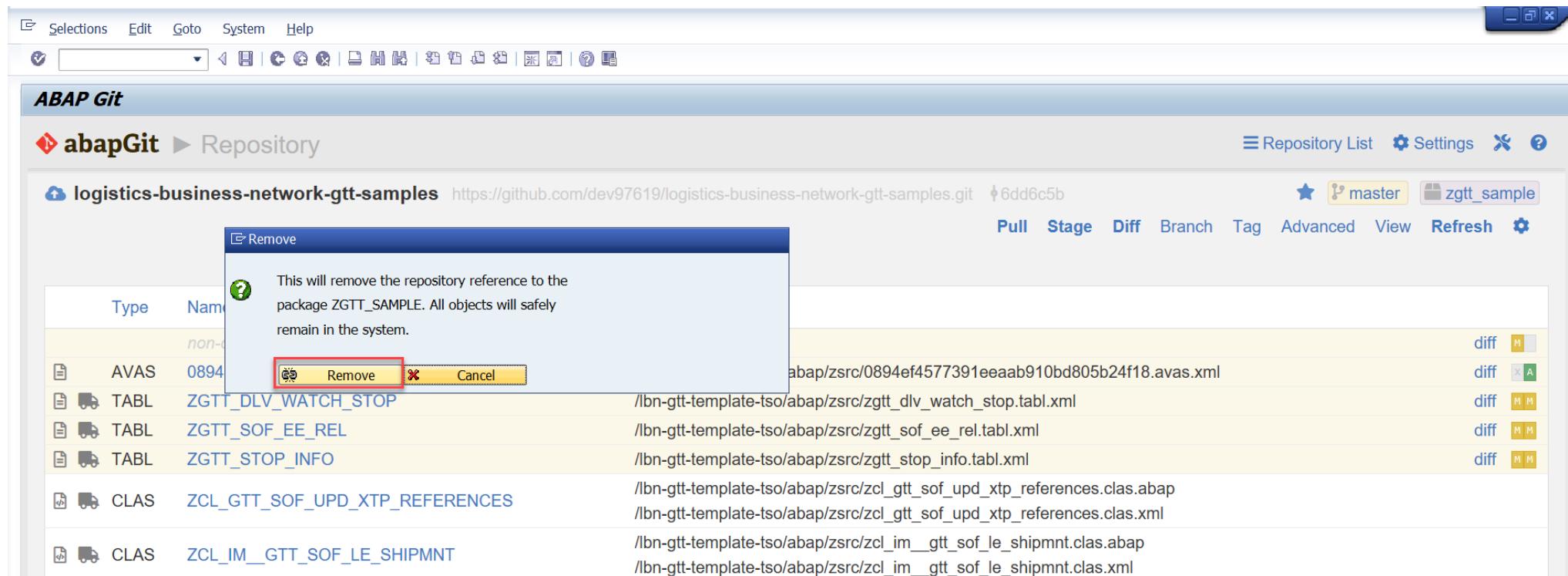
3-4: Under the “Advanced” menu, choose and click “Remove”.

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

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

STEP 3: Remove TSOF Repository in ABAPGit

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

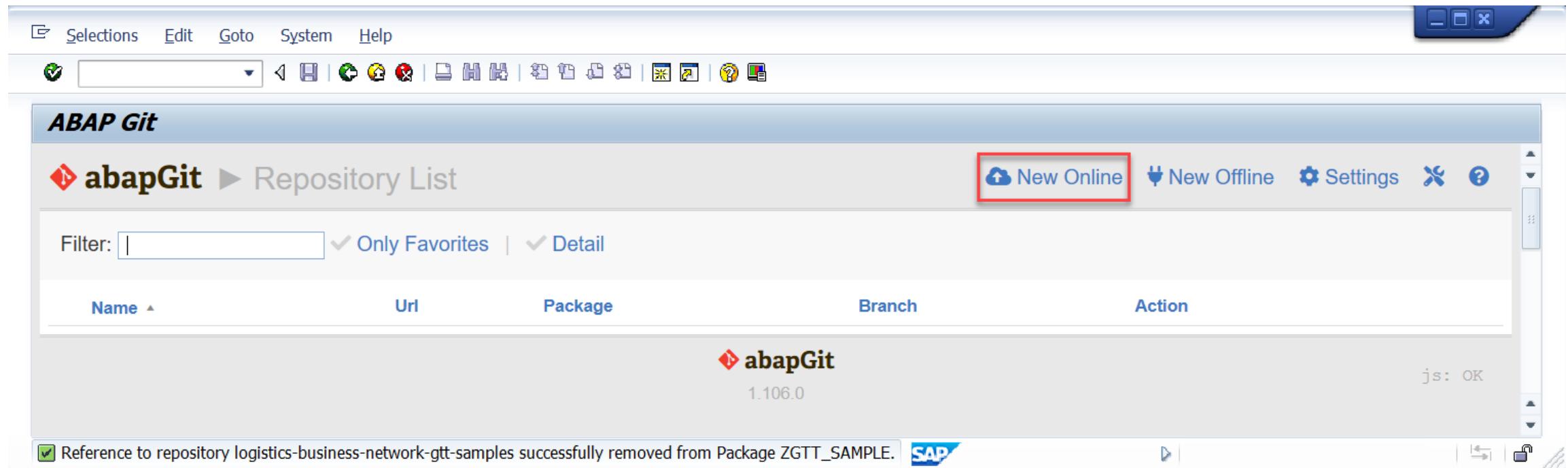


3-6: After repository removal you will see the following message:



STEP 4: Download TPOF Code from GitHub

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



STEP 4: Download TPOF Code from GitHub

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

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

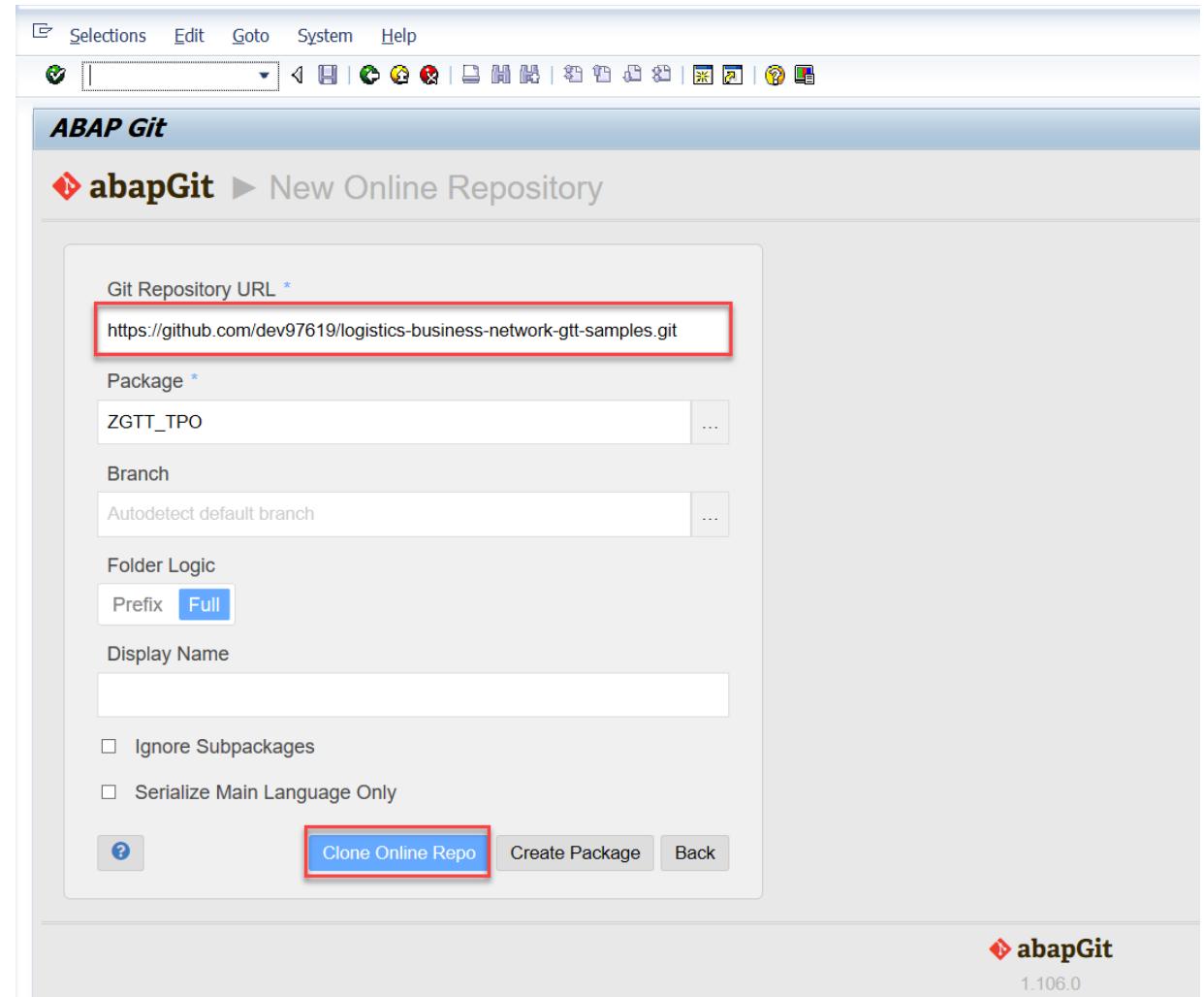
Caution:

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

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

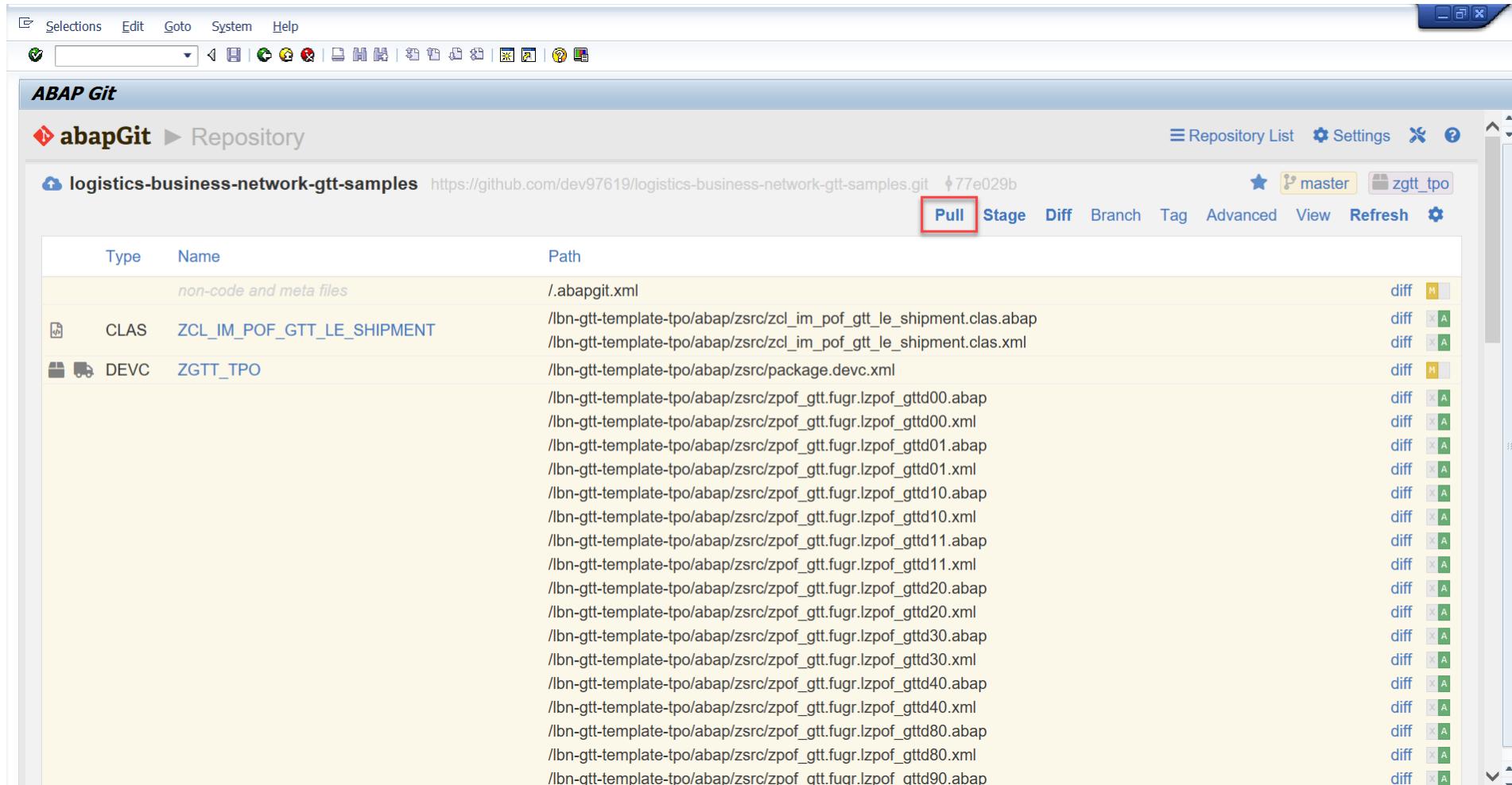
4-4: Set *Full* for **Folder Logic**

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



STEP 4: Download ABAP Code from GitHub

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

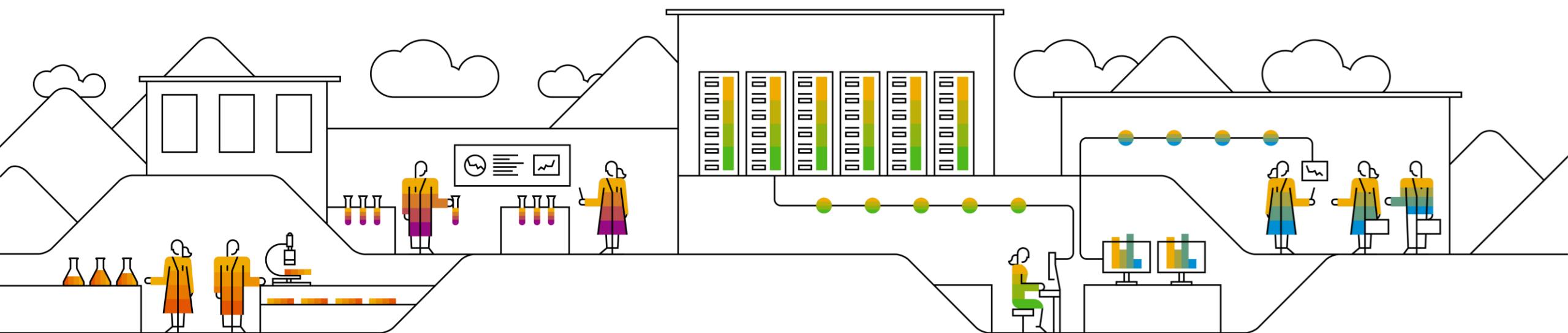


The screenshot shows the SAP ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation "abapGit > Repository". The main area displays a table of files under the repository "logistics-business-network-gtt-samples". The table has columns for Type, Name, Path, and Diff. A red box highlights the "Pull" button in the toolbar above the table. The "Path" column lists file paths such as ".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 "Diff" column shows status indicators like "diff N" or "diff A".

Type	Name	Path	Diff
non-code and meta files			
		/.abapgit.xml	diff N
CLAS	ZCL_IM_POF_GTT_LE_SHIPMENT	/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zcl_im_pof_gtt_le_shipment.clas.xml	diff A
DEVC	ZGTT_TPO	/lbn-gtt-template-tpo/abap/zsrc/package.devcl.xml	diff N
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt40.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt40.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt80.abap	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt80.xml	diff A
		/lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt90.abap	diff A

C) Download ABAP Code from GitHub

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



STEP 1: Install ABAPGit

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

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

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

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

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

 [Improve this page](#)

Summary #

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

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

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

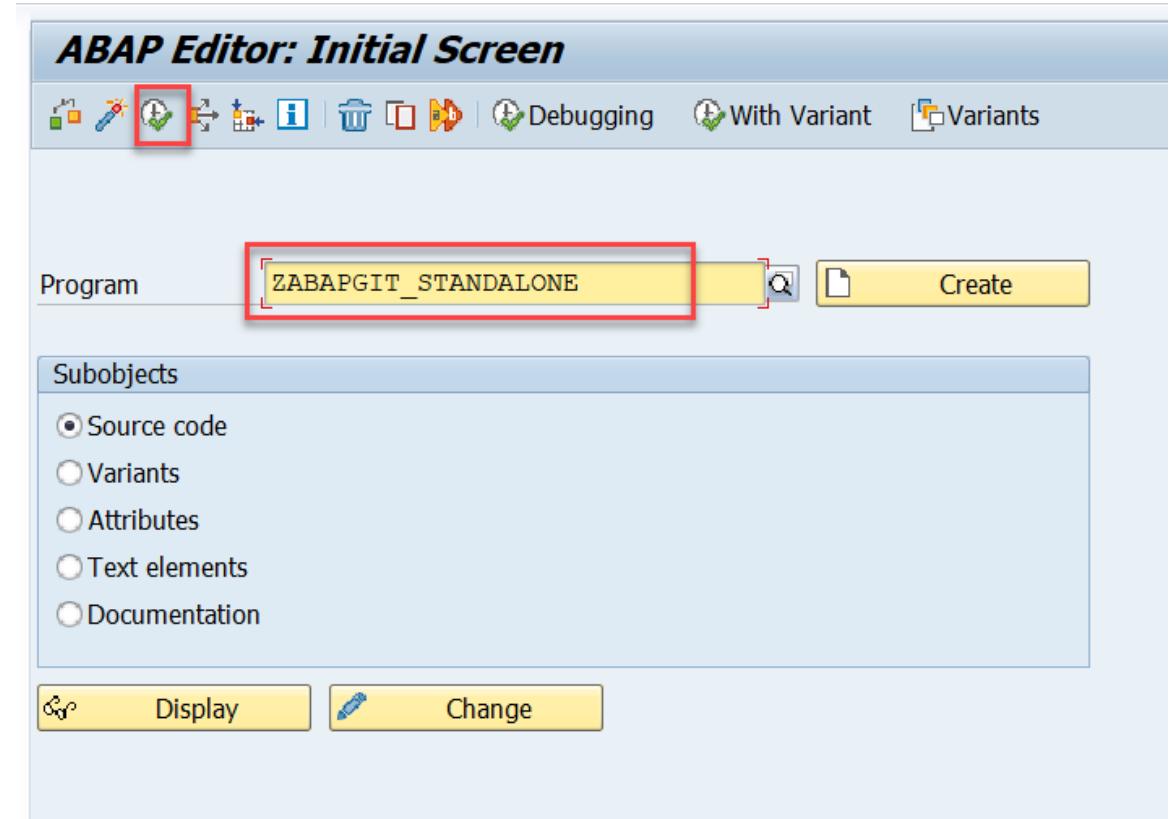
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

STEP 2: Download ABAP Code

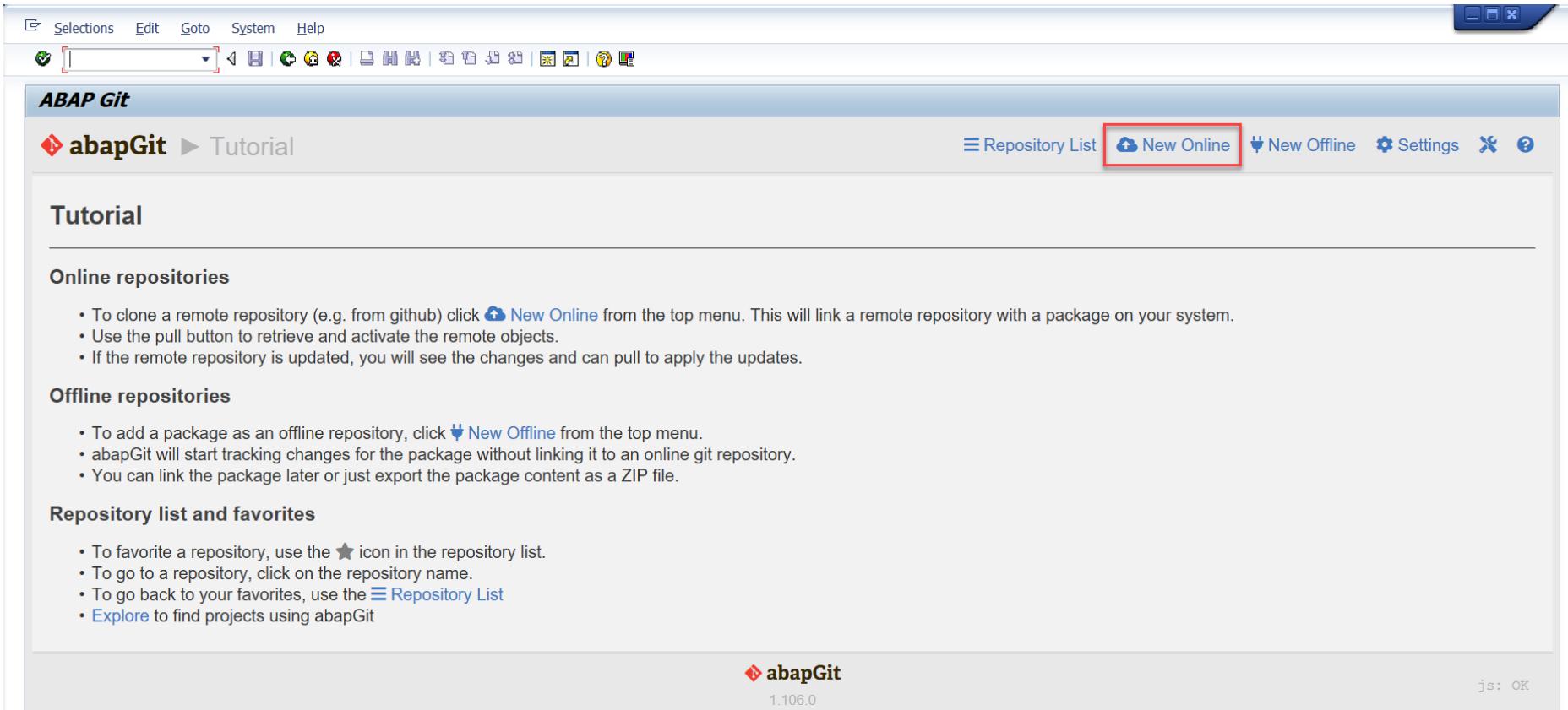
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 2: Download ABAP Code

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



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

Online repositories

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

Offline repositories

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

Repository list and favorites

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

STEP 2: Download ABAP Code

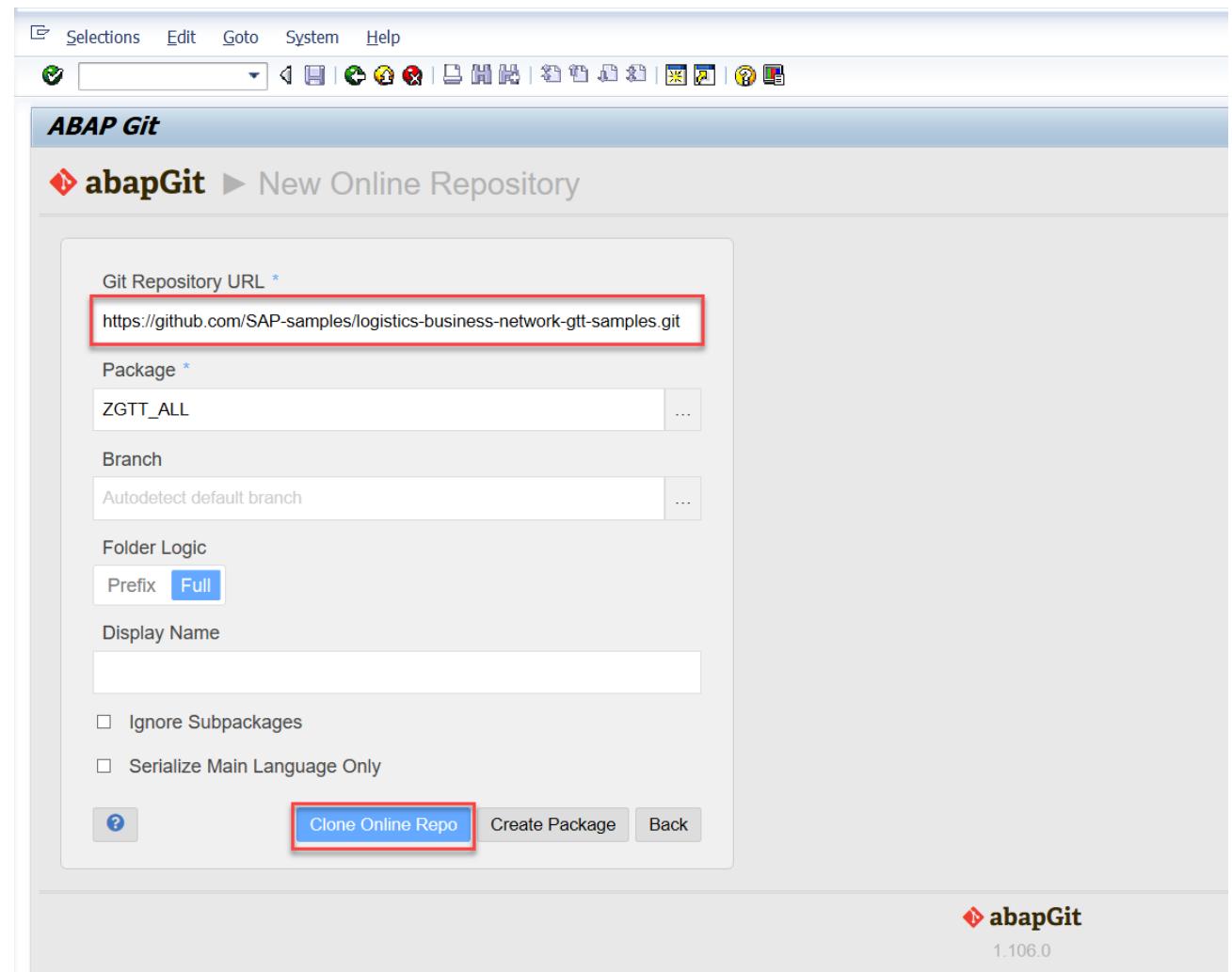
2-4: Fill in the **Git Repository URL**:

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

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

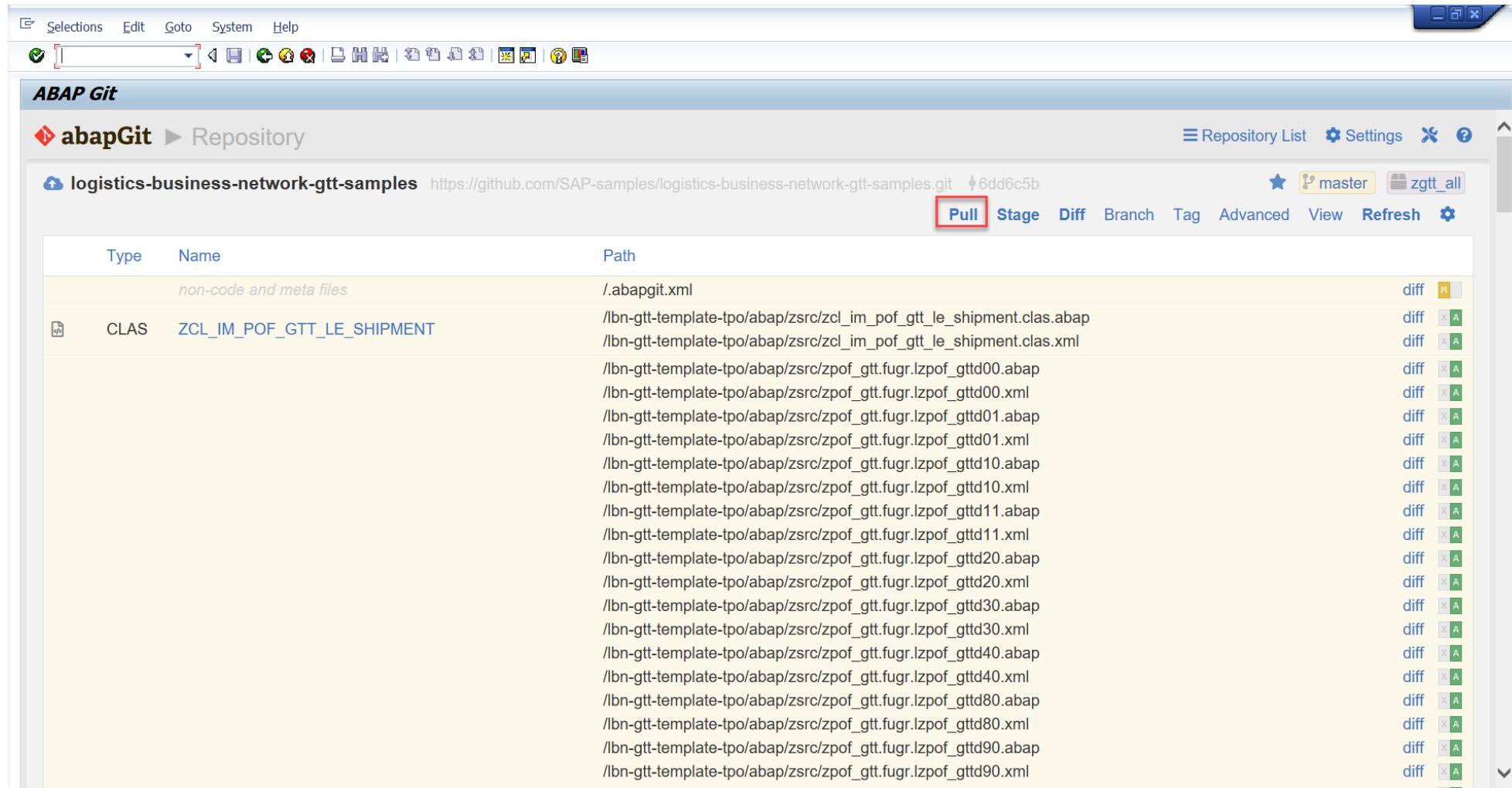
2-6: Set *Full* for **Folder Logic**

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



STEP 2: Download ABAP Code

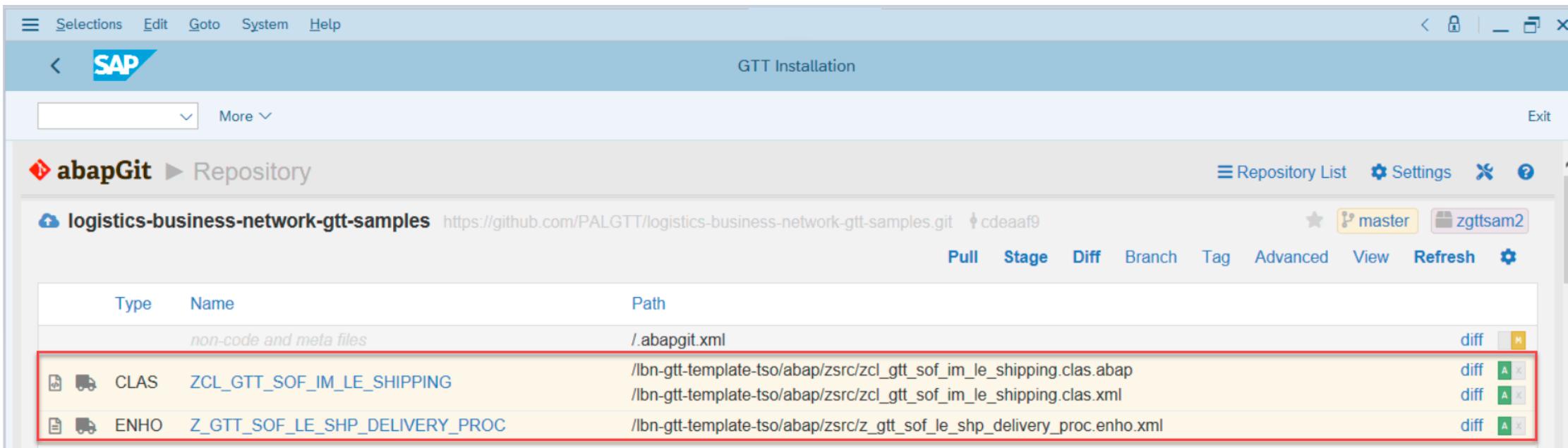
2-8: Click **Pull** to pull down the latest version code.



The screenshot shows the ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and the repository name "abapGit Repository". Underneath is another header with the repository URL "logistics-business-network-gtt-samples" and a commit hash "6dd6c5b". To the right of this header are buttons for "master" and "zgtt_all". A navigation bar below the headers includes "Repository List", "Settings", "Pull" (which is highlighted with a red box), "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The main area is a table with columns "Type", "Name", and "Path". The "Type" column shows entries like "non-code and meta files" and "CLAS". The "Name" column lists file names such as ".abapgit.xml", "ZCL_IM_POF_GTT_LE_SHIPMENT", and various ABAP and XML files under the ZCL_IM_POF_GTT_LE_SHIPMENT class. The "Path" column shows the full file paths. To the right of each row are "diff" buttons and small colored status indicators (yellow, green, blue).

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

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



The screenshot shows the SAP GTT Installation interface. At the top, it says "GTT Installation". Below that, it shows a GitHub repository "logistics-business-network-gtt-samples" with the URL <https://github.com/PALGTT/logistics-business-network-gtt-samples.git>. The commit hash is cdeaaf9. The master branch is selected. A red box highlights the "diff" button for two specific files: "ZCL_GTT_SOFTWARE_SHIPPING" and "Z_GTT_SOFTWARE_SHIPPING_PROC".

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

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

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

Solution:

Option 1:

1-1) Deactivate the BADI implementation.

Option 2:

2-1) Delete the enhancement implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC

2-2) Delete the BADI implementation class ZCL_GTT_SOF_IM_LE_SHIPPING

Notes:

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

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

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

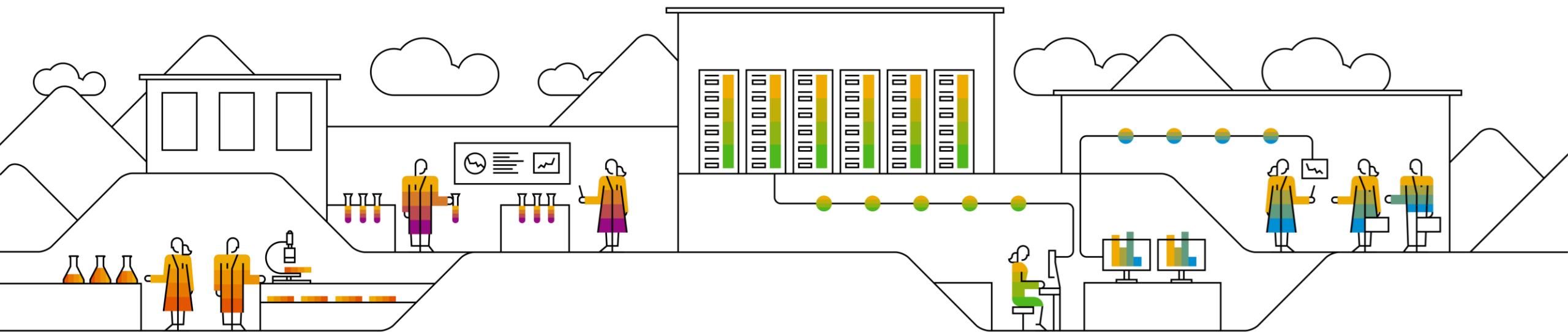
For option 1: Use transaction code SE19 and deactivate the BADI implementation.

The screenshot shows two SAP application windows. The top window is titled "BAdI Builder: Initial Screen for Implementations". It has a search bar and several menu items: Check, Delete implementation, Copy implementation, Rename implementation, Application help, and More. Below the menu is a section titled "Edit Implementation" with a radio button for "New BAdI" selected. The "Enhancement Implementation" field contains the value "Z_GTT_SOF_LE_SHP_DELIVERY_PROC", which is highlighted with a red box. The bottom window is titled "Enhancement Implementation Z_GTT_SOF_LE_SHP_DELIVERY_PROC Display". It also has a search bar and menu items: Previous Object, Next Object, Display >> Change, Other Object..., Check, Activate, Where-Used List, Display Object List, and Fullscreen On/Off. Below the menu is a table with tabs: Properties, History, Technical Details, and Implementation Elements. The "Implementation Elements" tab is active. It shows a list of "BAdI Implementations" with one entry: "Z_GTT_SOF_IM_LE_SHIPPING" (Implementation Class). To the right of the table are several configuration fields:

- BAdI Implementation: `Z_GTT_SOF_IM_LE_SHIPPING`
- Description: Implementation: GTT - Enhancement to update the imputed sales orders' delivery list
- Default Implementation (checkbox)
- Example Implementation (checkbox)
- "Active" not switchable in customizing (IMG) (checkbox, checked)
- Runtime Behavior
 - Implementation is active (checkbox, checked)
 - Runtime Behavior: The implementation will not be called (text)

D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

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

The image shows two screenshots of the SAP GTT (Global Trace and Trace) interface. The left screenshot displays the 'Define Application Object Types' screen under 'Define Used Business Process'. It shows a business process type 'ESC_SORDER' and an application object type 'ZGTT_SO_INT_HD' (highlighted with a red box). The right screenshot shows the 'IDOC Integration' tab of the 'sof' namespace configuration. It lists a tracked process 'SalesOrder' and its mapping to an ERP object type 'Others'. Both the application object type 'ZGTT_SO_INT_HD' and the tracked process name 'SalesOrder' are highlighted with red boxes to indicate they must match.

2: Maintain Tracking ID Type

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

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

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

SAP AOT Screenshot: Shows the "Display View 'Define Application Object Types': Details" screen. Under "Bus. Proc. Type:" is "ESC_SORDER". Under "Appl. Obj. Type:" is "ZGTT_SO_INT_HD" with the description "Extract sales order header information to Global Track and Trace Integration". Under "Text:" is "Sales Order Header". In the "Parameter Setup" section, under "Tracking ID Setup", the "TrkID Method:" is set to "B Determine from Field", "Tr.ID Tab. Type:" is "1 Main Object Table", and "Trk.ID Code Set:" is "SALES_ORDER". The "Tracking ID Fld:" field contains "VBELN".

SAP Model Details Screenshot: Shows the "Model Details" screen for "SOF" (Active). Under "Tracked Process", it lists "Items (6)": SalesOrder, SalesOrderItem, Delivery, and DeliveryItem. The "SalesOrder" item has a red box around its "Tracking Id Type: SALES_ORDER" field. In the "User Model Fields (16)" table, the "SalesOrder" row also has a red box around its "Tracking Id Type: SALES_ORDER" field. A modal dialog titled "Edit Tracked Process" is open, showing the "Name:" field as "SalesOrder" and the "Tracking Id Type:" field as "SALES_ORDER", both of which are also highlighted with red boxes. The "OK" button is visible at the bottom right of the modal.

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

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

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

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

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

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

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

4: Sample Codes for Sales Order Fulfillment Application

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

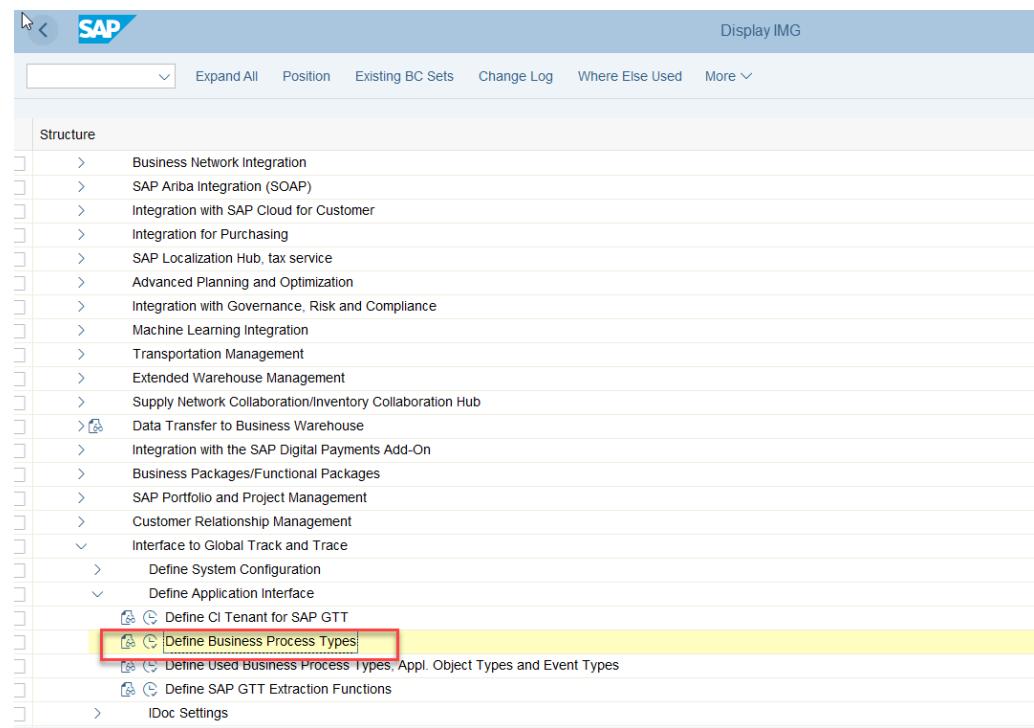
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_HD	Function for setup of control parameters of delivery header
Control Parameter Extractors	ESC_DELIV	ZGTT_SOF_OTE_DE_ITM	Function for setup of control parameters of delivery item
Control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

5: Available Contexts for the extractors' modules

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

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

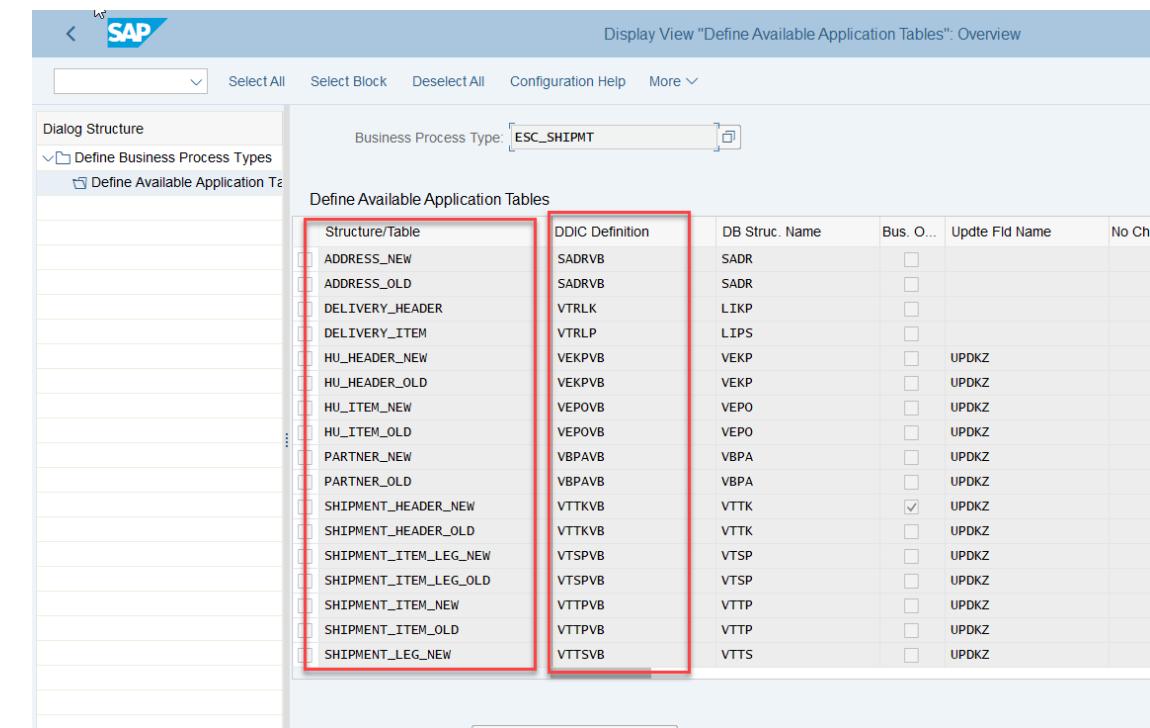
5-3: Please select the Business Process Types to find all the context tables and their structure info.



Display IMG

Structure

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



Display View "Define Available Application Tables": Overview

Dialog Structure

Business Process Type: **ESC_SHIPMT**

Define Available Application Tables

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

6: Coding Tips in the GTT relevance function modules

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

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

See sample code of function: `ZGTT_SOF_OTE_SHP_HDR_REL`

Function Builder: Display ZGTT_SOF_OTE_SHP_HDR_REL

Attributes Import Export Changing Tables Exceptions Source Code

```
28      <ls_xvttk>      TYPE vttkvb.  
29  
30      * <i> Check if Main table is Shipment or not.  
31      IF i_app_object-maintabdef <> gc_bpt_shipment_header_new.  
32          PERFORM create_logitable_ao_rel  
33          TABLES c_logitable  
34          USING i_app_object-maintabdef  
35              space  
36              i_app_obj_types-trrfunc  
37              i_app_object-appobjtype  
38              i_appsy.  
39          RAISE parameter_error.  
40      ELSE.  
41          * Read Main Object Table (Shipment - VTTK)  
42          ASSIGN i_app_object-maintabref*> TO <ls_xvttk>.  
43      ENDIF.  
44  
45  
46      *3 Check Relevance of AOT: IN_OTE  
47      PERFORM check_act_relevance_shp  
48      USING <ls_xvttk>  
49      CHANGING lv_act_relevance.  
50      CHECK lv_act_relevance IS NOT INITIAL.  
51  
52      When shipment is newly created, check relevance of GTT: only when delivery has been assigned.  
53      IF <ls_xvttk>-updtx EQ gc_insert.  
54          PERFORM check_delivery_assignment  
55          USING i_all_appl_tables
```

7: Coding Tips in the Tracking ID function modules

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

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

See sample code of function:

ZGTT_ADD_TRACKID_OTE_SHPHDR

The screenshot shows the SAP Function Builder interface with the title bar "Function Builder: Display ZGTT_ADD_TRACKID_OTE_SHPHDR". Below the title bar, there are tabs: Attributes, Import, Export, Changing, Tables, Exceptions, and Source Code. The "Source Code" tab is selected. The code area displays the ABAP source code for the function module. A red box highlights the first section of code, and another red box highlights the second section of code. The code uses IF statements to handle different update types (I, O1, O4, U) and reads data from a table to fill the tracking ID data structure.

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

8: Coding Tips in the Control Parameter function modules

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

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

See sample code of function: *ZGTT_SOF_OTE_SHP_HD*

8: Coding Tips in the Control Parameter Function Modules

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

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

The main navigation bar includes tabs for 'Tracked Process', 'Field Type Pool', 'Event Type Pool', 'Code List', 'IDOC Integration' (which is underlined in blue), 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. A central search bar has 'Tracked Process: Shipment' and an 'Integration Switch' button set to 'ON'.

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

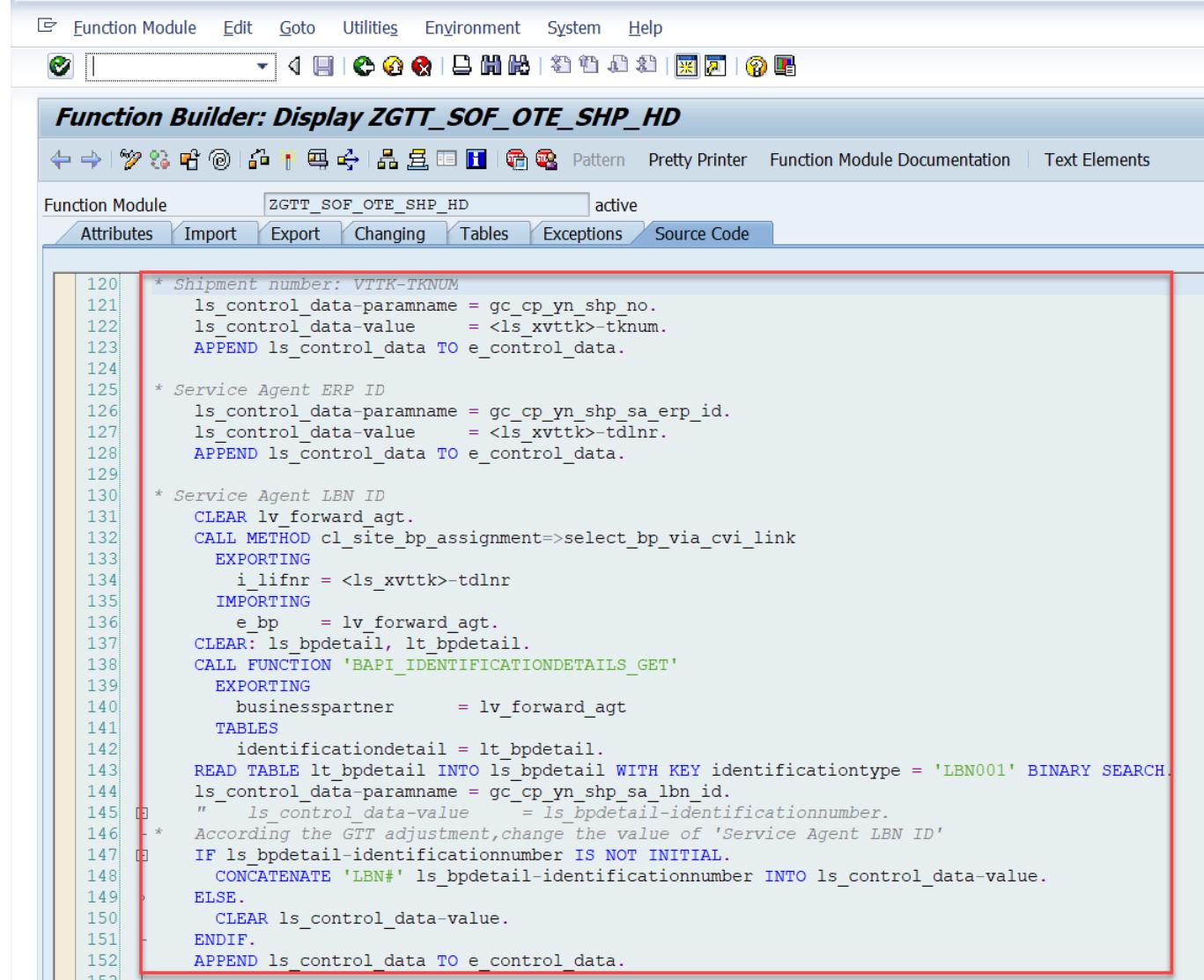
The 'Tracked Process / Events (26)' table lists various tracked processes and their corresponding IDOC segments and event codes. One row for 'ShipmentEvent' is highlighted.

The 'User Model Fields' table, which is highlighted with a red border, maps application object fields to IDOC segments and fields. It includes columns for 'Field', 'IDOC Segment', and 'IDOC Field'.

User Model Fields		
Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPBP	YN_SHP_NO
serviceAgentLbnId	E1EHPBP	YN_SHP_SA_LBN_ID
transportationMode	E1EHPBP	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPBP	YN_SHP_CONTAIN_DGOODS
forwardingAgentTrackingId	E1EHPBP	YN_SHP_FA_TRACKING_ID
> stops		
shippingType	E1EHPBP	YN_SHP_SHIPPING_TYPE
> resourceTPs		

8: Coding Tips in the Control Parameter Function Modules

Main logic of shipment is implemented in function module ZGTT_SOF_OTE_SHP_HD



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

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

9: Coding Tips in the Planned Event function modules

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

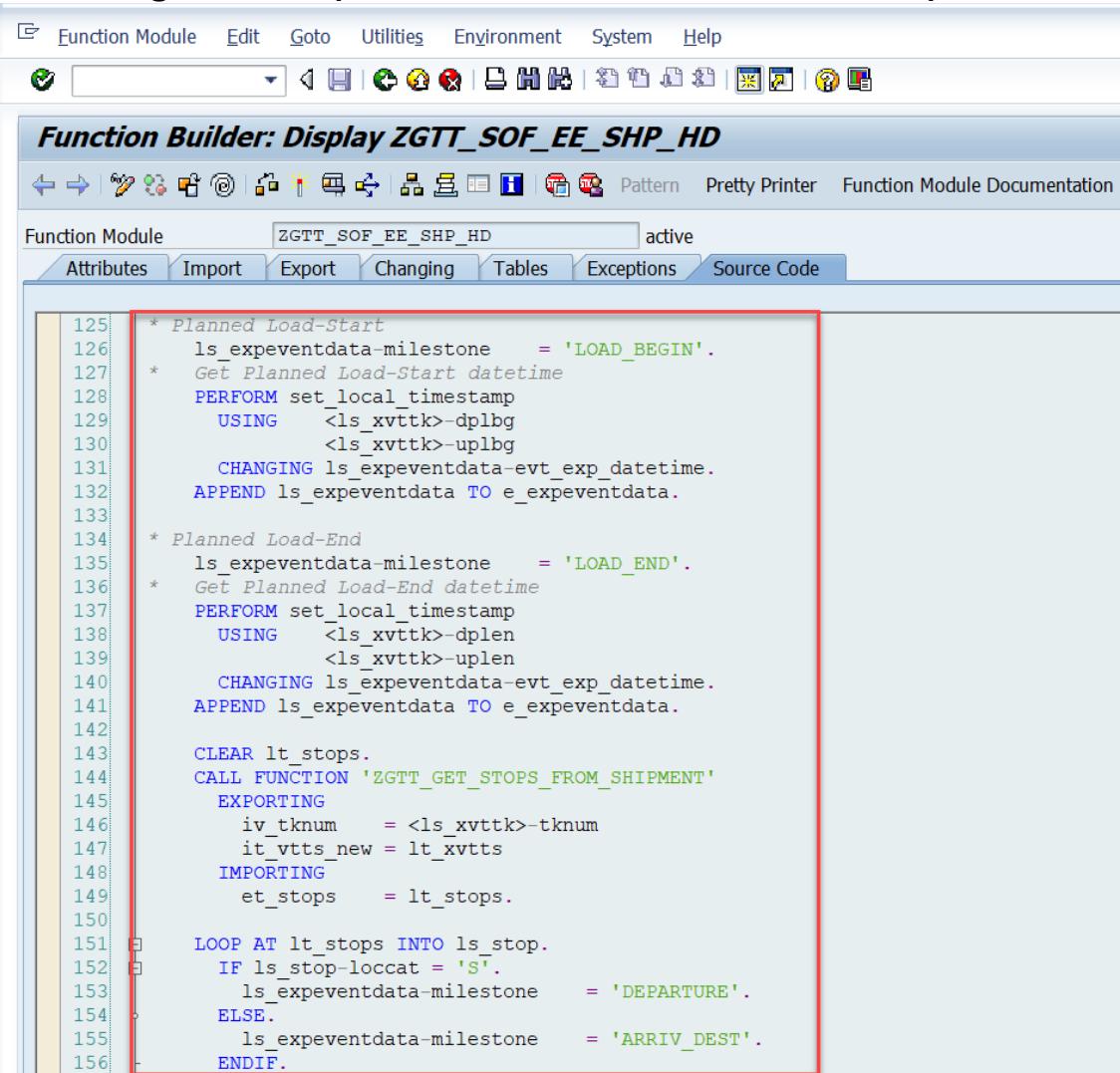
1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. As default except no change made on the model configuration, GTT version 2 asks for full transport for all the planned events, which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

See sample code of function: *ZGTT_SOF_EE_SHP_HD*

Tracked Process	Field Type Pool	Event Type Pool	Code List	IDOC Integration	Visibility Provi
Tracked Process: Shipment					
Tracked Process Mapping					
ERP Object Type: Others					
Name	IDOC	Event Code			
Tracked Process					
ShipmentEvent	E1EHPAO				
Event Types					
LoadingStart	E1EVMHDR02	LOAD_BEGIN			
POD	E1EVMHDR02	POD			
Departure	E1EVMHDR02	DEPARTURE			
Arrival	E1EVMHDR02	ARRIV_DEST			
LoadingEnd	E1EVMHDR02	LOAD_END			
Checkin	E1EVMHDR02	CHECK_IN			

9: Coding Tips in the Planned Event Function Modules

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



The screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZGTT_SOF_EE_SHP_HD". The function module "ZGTT_SOF_EE_SHP_HD" is active. The "Source Code" tab is selected. The code is written in ABAP and handles planned load events. A red box highlights the logic for "Planned Load-Start" and "Planned Load-End".

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

10: Coding Tips in the Event Data function modules

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

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

See sample code of function: *ZGTT_SOF_EE_DE_PICKING*

10: Coding Tips in the Event Data Function Modules

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

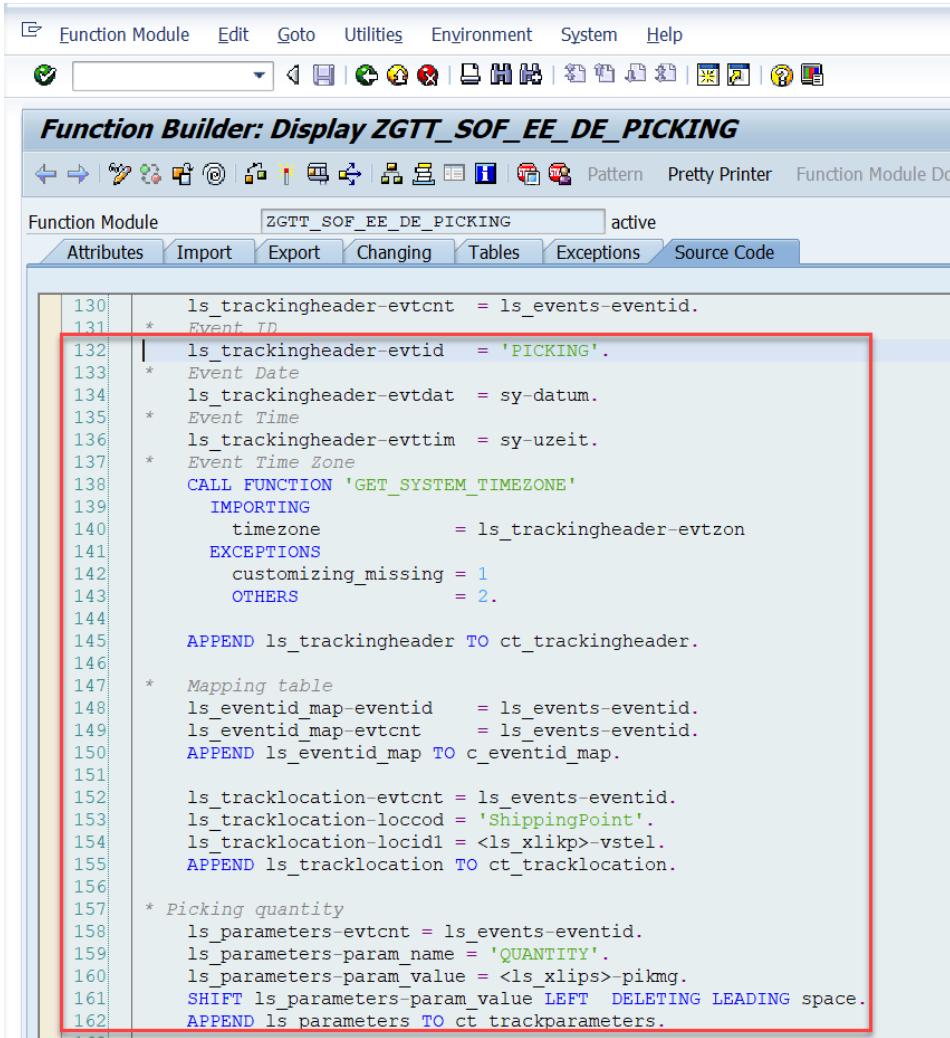
The screenshot shows the SAP Model Details interface for the 'Internal - Test' model. The 'sof' model is active. The 'IDOC Integration' tab is selected. The 'Tracked Process' dropdown is set to 'DeliveryItem'. The 'Integration Switch' is turned 'ON'. The 'Tracked Process Mapping' section shows 'ERP Object Type: Others' and 'Application Object Type: ZGTT_DE_INT_ITEM'. The 'Tracked Process / Events (4)' table lists four entries: 'Tracked Process' (row), 'DeliveryItemEvent' (row), 'Event Types' (row), and three specific event types ('Picking', 'Packing', 'DeliveryItemPOD'). The 'User Model Fields' table maps fields like 'quantity' to IDOC segments and fields. The row for 'quantity' is highlighted with a red border.

Name	IDOC	Event Code
Tracked Process		
DeliveryItemEvent	E1EHPAO	
Event Types		
Picking	E1EVMDR02	PICKING
Packing	E1EVMDR02	PACKING
DeliveryItemPOD	E1EVMDR02	DLV POD

Field	IDOC Segment	IDOC Field
quantity	E1EVMPAR	QUANTITY

10: Coding Tips in the Event Data Function Modules

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



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

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

11: Enhancement codes for cross-processes tracking

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

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

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

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

SAP Business Add-In Builder: Display Implementation Z_GTT_SOF_LE_SHIPMNT

Implementation Name: Z_GTT_SOF_LE_SHIPMNT (Active)

Implementation Short Text: GTT - Enhancement to update the impacted delivery orders

Definition Name: BADI_LE_SHIPMENT

Runtime Behavior: Implementation will be called

Properties Interface

Interface Name: IF_EX_BADI_LE_SHIPMENT

Name of Implementing Class: ZCL_IM_GTT_SOF_LE_SHIPMNT

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

Default Implementation Class:

11: Enhancement codes for cross-processes tracking

The cross processes tracking scenarios cover below:

Shipment -> Delivery and Delivery Item:

1\ Tracking ID (Delta Transport)

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

2\ Shipment Composition (Full Transport)

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

3\ Planned Event in Delivery (Full Transport)

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

4\ Planned Event in Delivery Item (Full Transport)

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

12: Known Issues

1. Planned Event Extension not enabled

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

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

2. IDOC sequencing issue

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

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

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

13: Solution of IDOC Sequencing Issue

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

2. Create CI tenant.

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

3. Create RFC destination

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

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

The image contains two screenshots of SAP GUI screens. The top screenshot shows the 'SAP Global Track & Trace Definitions' table with one entry: ZGTTSOFIN2, ZLSGTTINT, GTT2.0 Logistics Business N..., and CI For GTT V2 Integration system Sales Order Sample APP. The bottom screenshot shows the 'RFC Destination ZGTTV2_SOF_INT2' configuration screen. It includes fields for RFC Destination (ZGTTV2_SOF_INT2), Connection Type (HTTP Connection to External Server), and Description (Description 1: RFC for Tracked Process of Sales Order Sample Application). At the bottom, the 'Path Prefix' field is highlighted with a red box and contains the value `/api/idoc/em/v1/TrackedProcessAndEvent`.

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[®]