



SAP Business Network Global Track and Trace Track PO Fulfillment - SAP ERP Integration

SAP Business Network for Logistics
March 2021

PUBLIC

Objectives



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

- Learn what prerequisites are necessary for SAP Business Network Global Track and Trace
- Learn how to maintain IDOC configurations in ERP for integration
- Learn how to maintain extractors in ERP for integration
- Learn how to download and implement sample ABAP code from Github
- Learn how to customize the logic based on sample code

Agenda

A Prerequisites

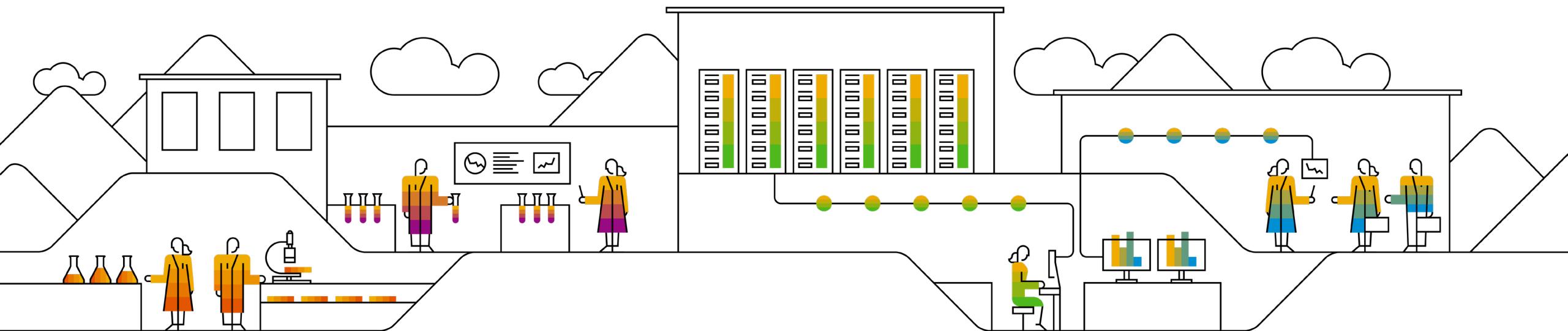
B Configuration and Implementation - Basic

 B1 IDOC Configuration

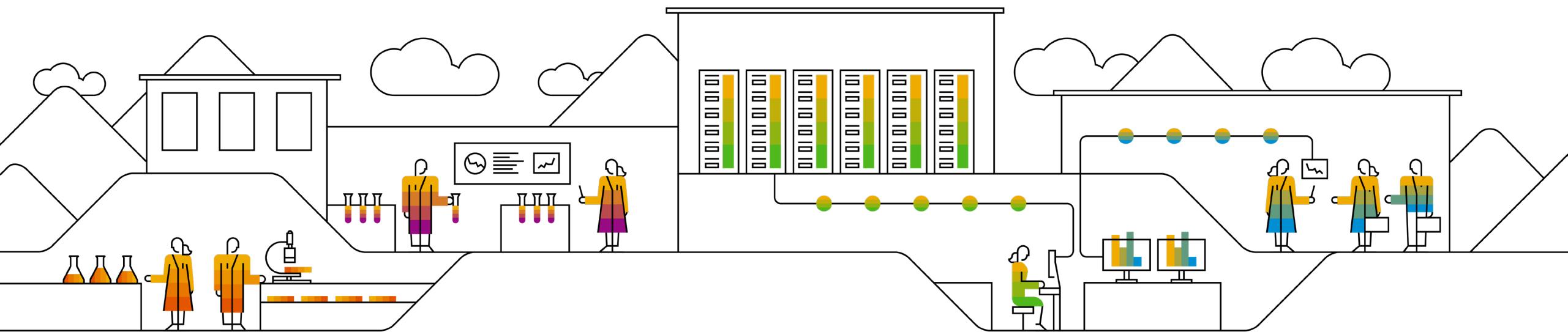
 B2 Extractor Configuration

C Download ABAP Code from GitHub

D Configuration and Coding Guide - Advanced



A) Prerequisites



STEP 1: Check the SAP Product Version

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

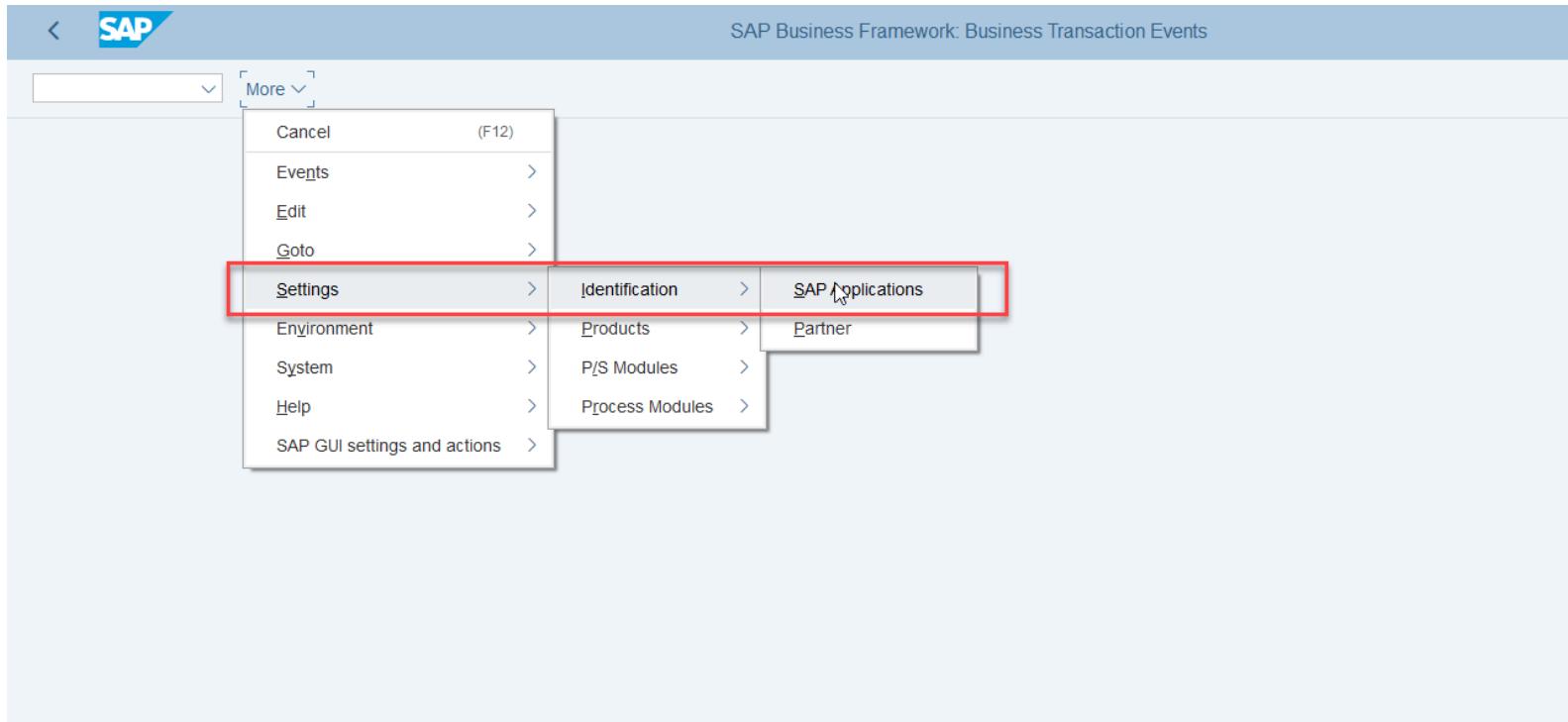
1-2: The ABAP codes on Github to support sample apps for SAP Business Network Global Track and Trace shall be implemented in SAP S/4HANA 1909 SP03 on premise or higher. Please note that the codes are not validated in its lower version or other ECC series of products, so you might need to do further adaptation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

STEP 2: Log on the Development Client to Configure BTE

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

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

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

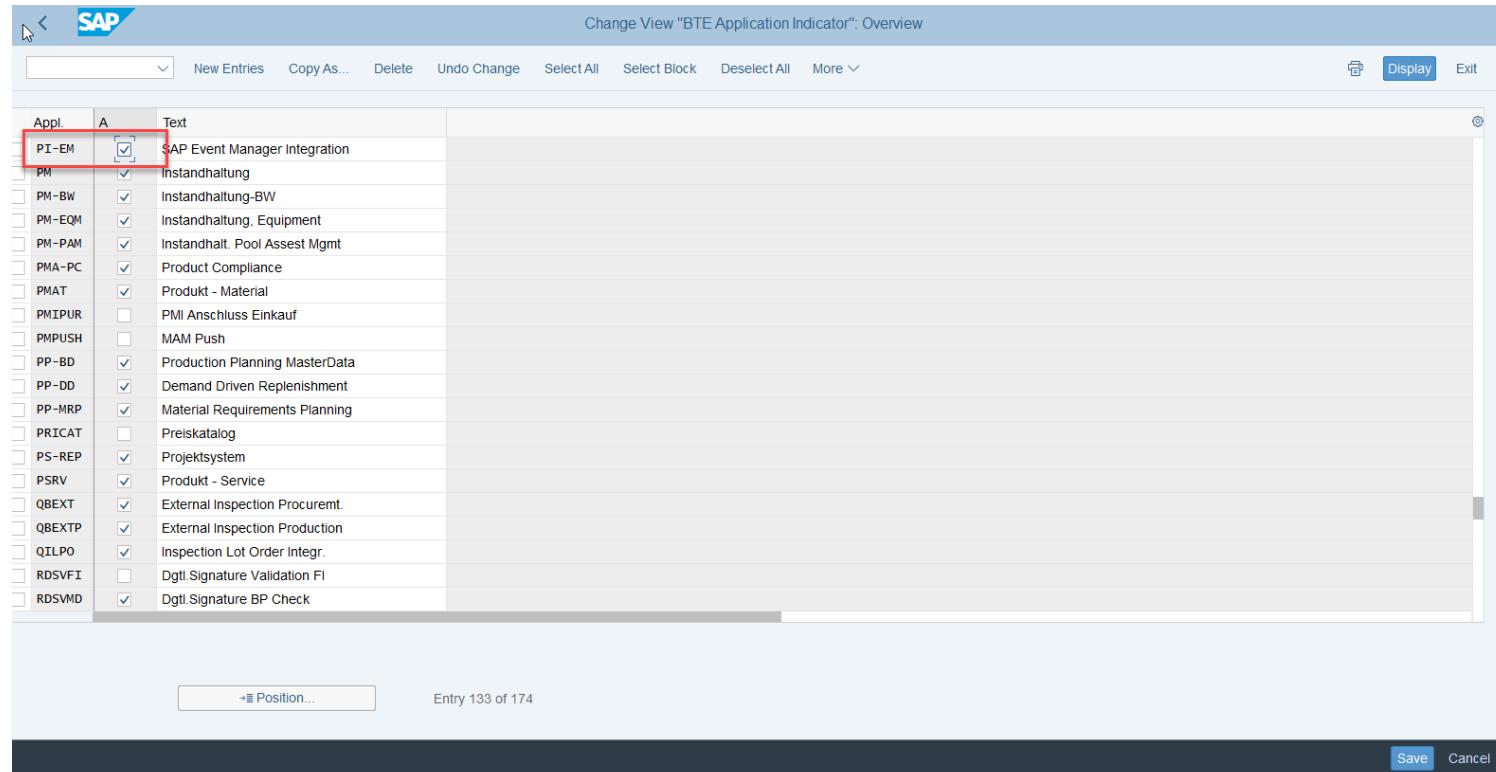


STEP 2: Activate SAP Event Manager Integration

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

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

2-6: Click **Save**



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
MPUSH	<input type="checkbox"/>	MAM Push
PP-BD	<input checked="" type="checkbox"/>	Production Planning MasterData
PP-DD	<input checked="" type="checkbox"/>	Demand Driven Replenishment
PP-MRP	<input checked="" type="checkbox"/>	Material Requirements Planning
PRICAT	<input type="checkbox"/>	Preiskatalog
PS-REP	<input checked="" type="checkbox"/>	Projektsystem
PSRV	<input checked="" type="checkbox"/>	Produkt - Service
QBEXT	<input checked="" type="checkbox"/>	External Inspection Procurement
QBEXTP	<input checked="" type="checkbox"/>	External Inspection Production
QILPO	<input checked="" type="checkbox"/>	Inspection Lot Order Integr.
RDSVFI	<input type="checkbox"/>	Dgtl.Signature Validation FI
RDSVMD	<input checked="" type="checkbox"/>	Dgtl.Signature BP Check

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



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

1-1: Log on to the business client

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

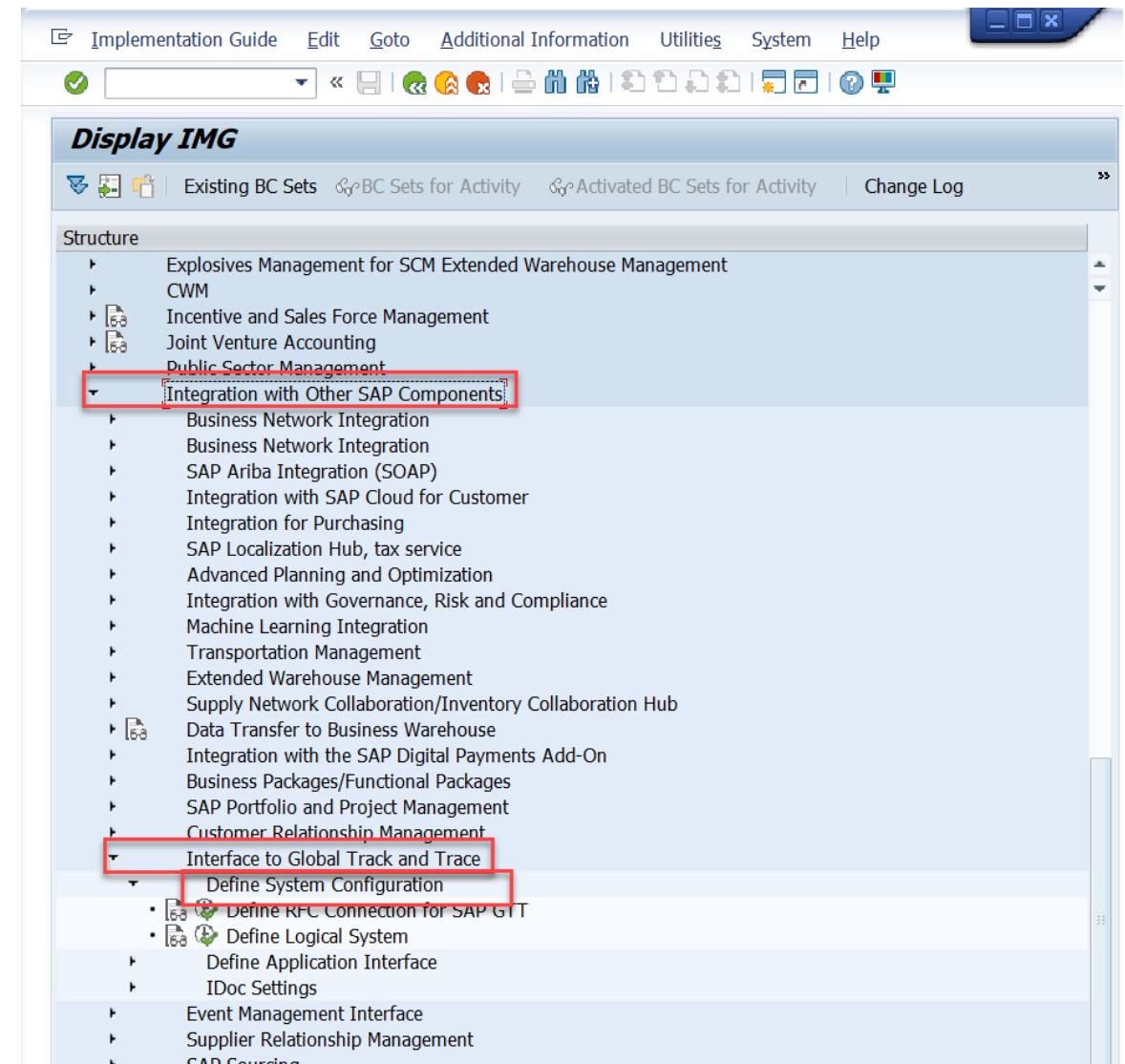
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

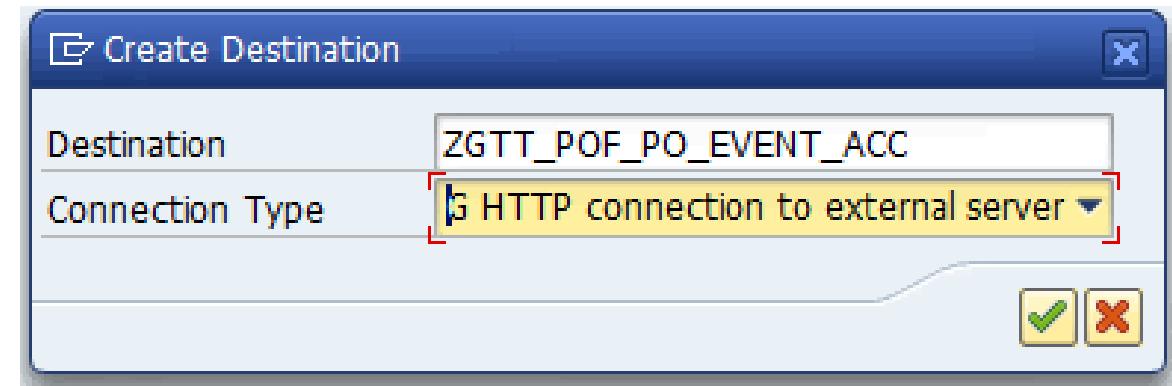
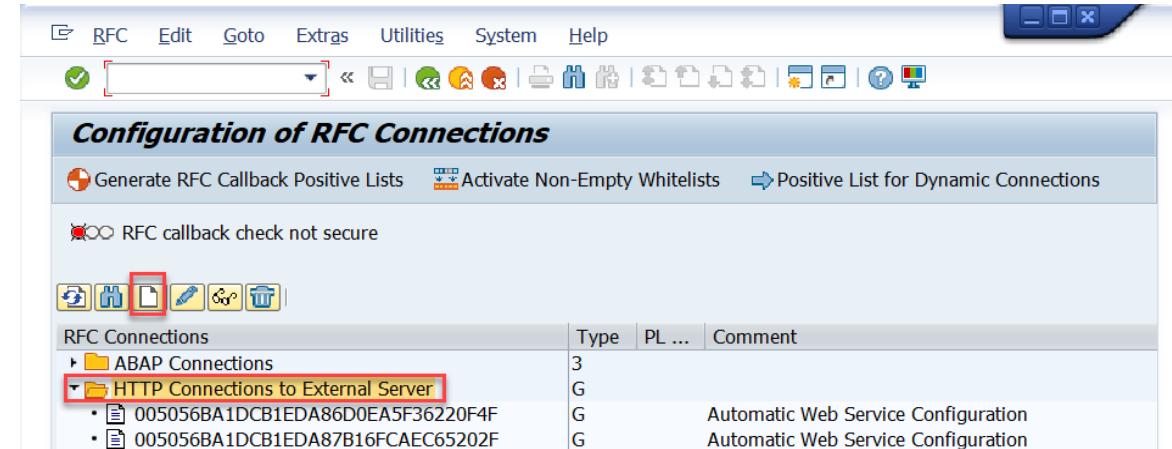
Define RFC Connection for SAP GTT



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

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

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



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

1-7: Enter a description

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

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

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

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

Port: 443

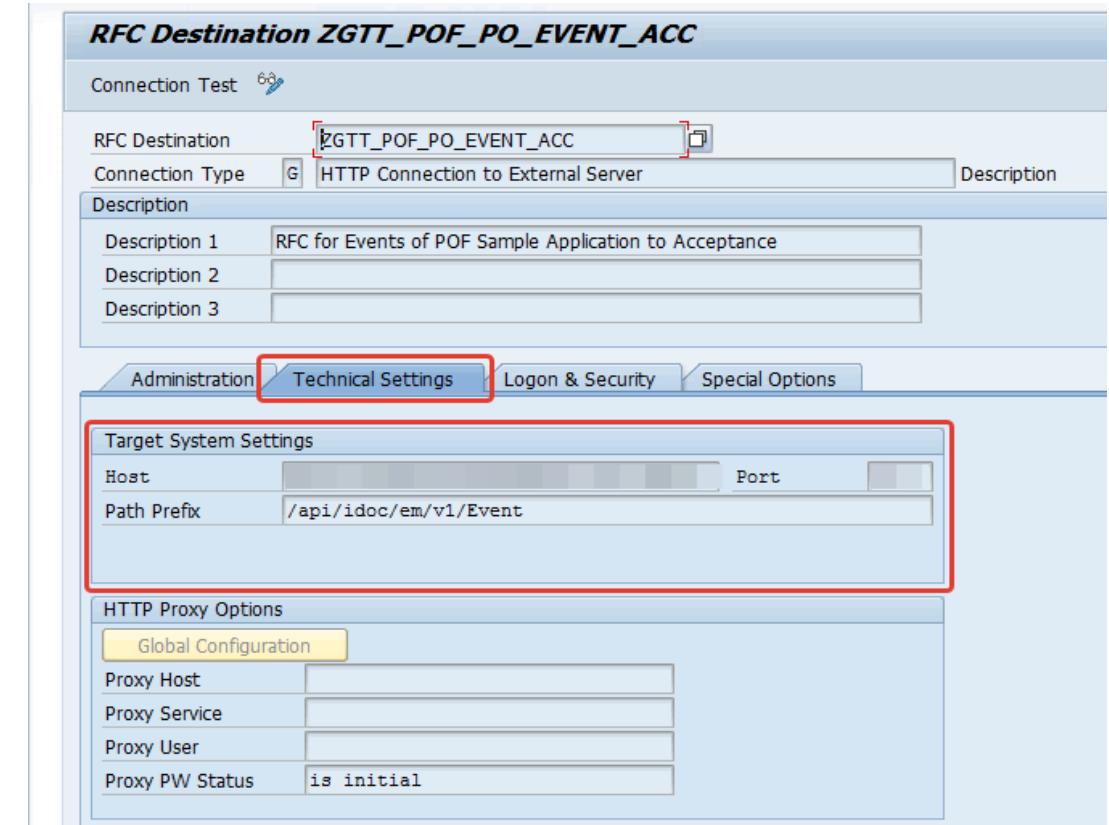
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

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
ZGTT_POF_PO_EVENT_ACC	RFC for Events of POF Sample Application to Acceptance	xxxxxx.xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/Event	443
ZGTT_POF_PO_TP_ACC	RFC for Tracked Process of POF Sample Application to Acceptance	xxxxxx.xxxxxx.gtt-flp-lbnplatform.cfapps.eu10.hana.ondemand.com	/api/idoc/em/v1/TrackedProcess	443

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

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

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

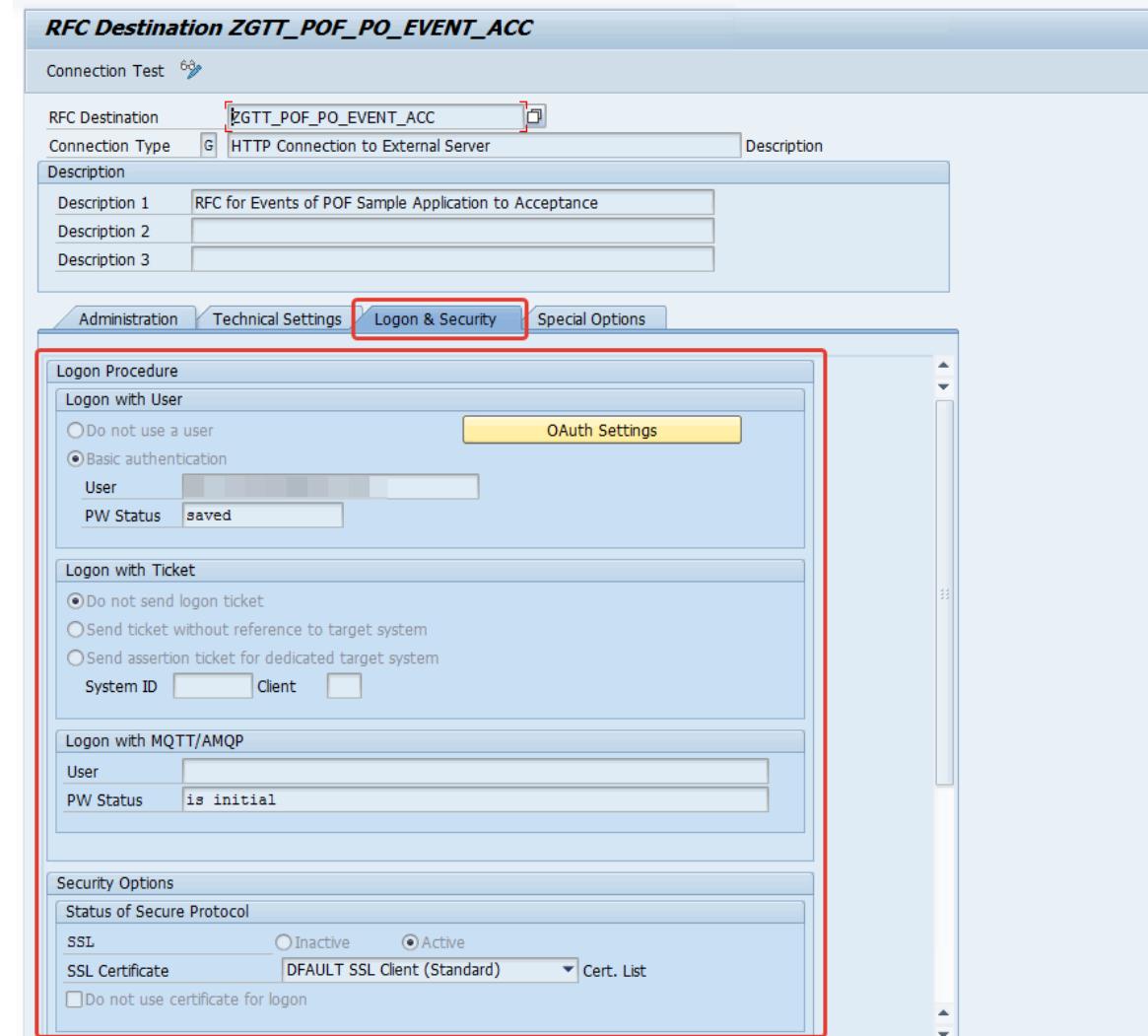
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

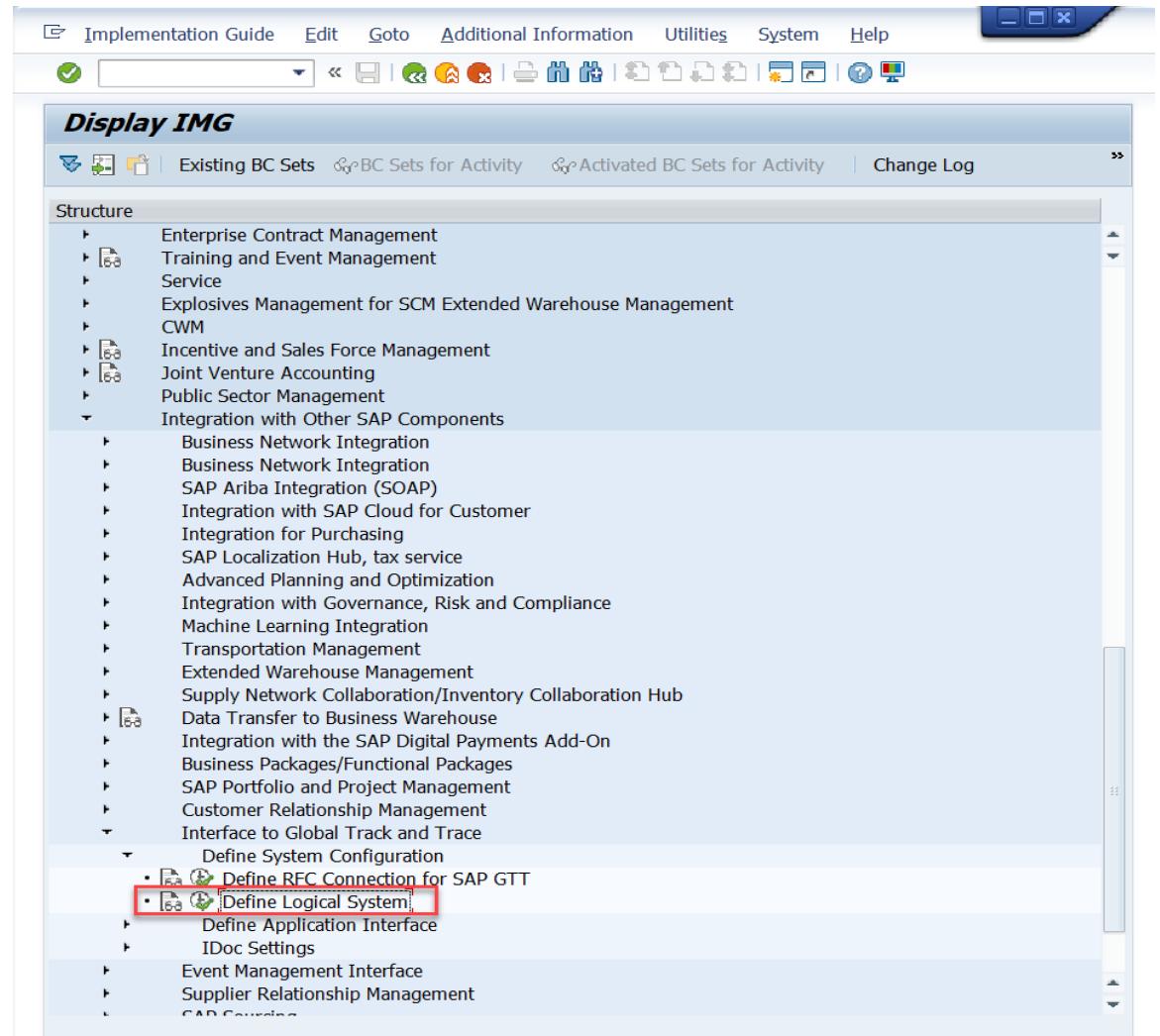
- one for event and
- the other for tracked Process.



STEP 2: Define Logical System

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

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

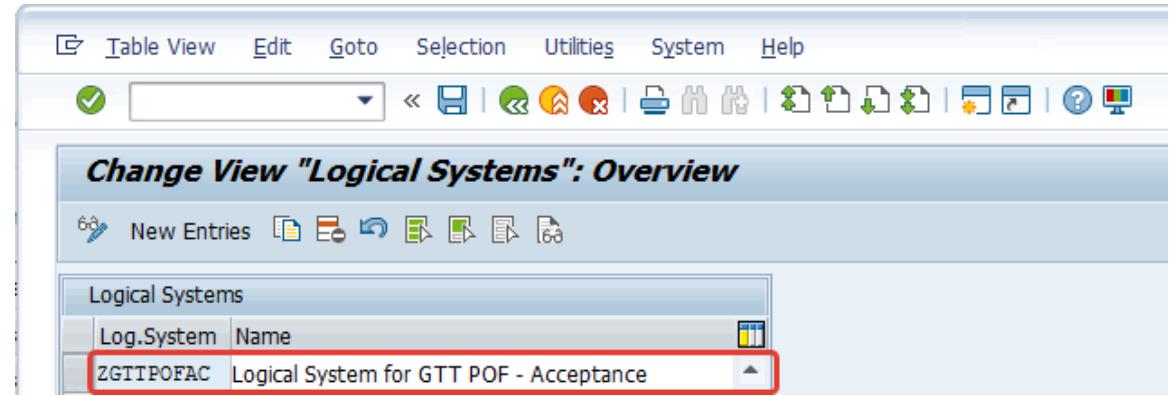


STEP 2: Define Logical System

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

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

2-4: Save the configuration



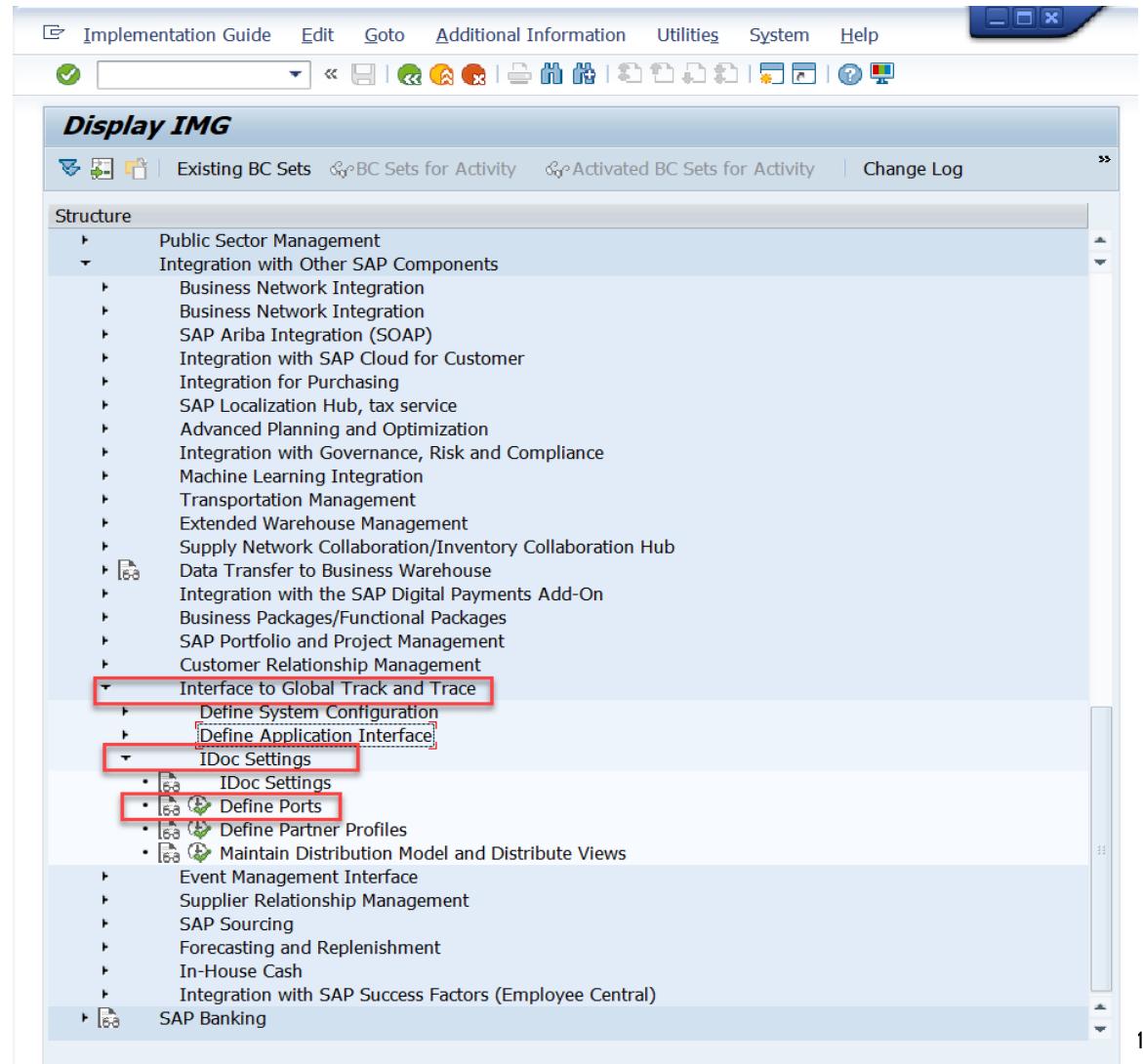
The screenshot shows the SAP GUI interface for defining logical systems. The title bar reads "Change View 'Logical Systems': Overview". Below the toolbar, there is a menu bar with "Table View", "Edit", "Goto", "Selection", "Utilities", "System", and "Help". The main area displays a table titled "Logical Systems" with two columns: "Log.System" and "Name". A new entry has been added, highlighted with a red border: "ZGTTPOFAC" in the Log.System column and "Logical System for GTT POF - Acceptance" in the Name column.

Log.System	Name
ZGTTPOFAC	Logical System for GTT POF - Acceptance

STEP 3: Define Ports

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

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



STEP 3: Define Ports

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

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

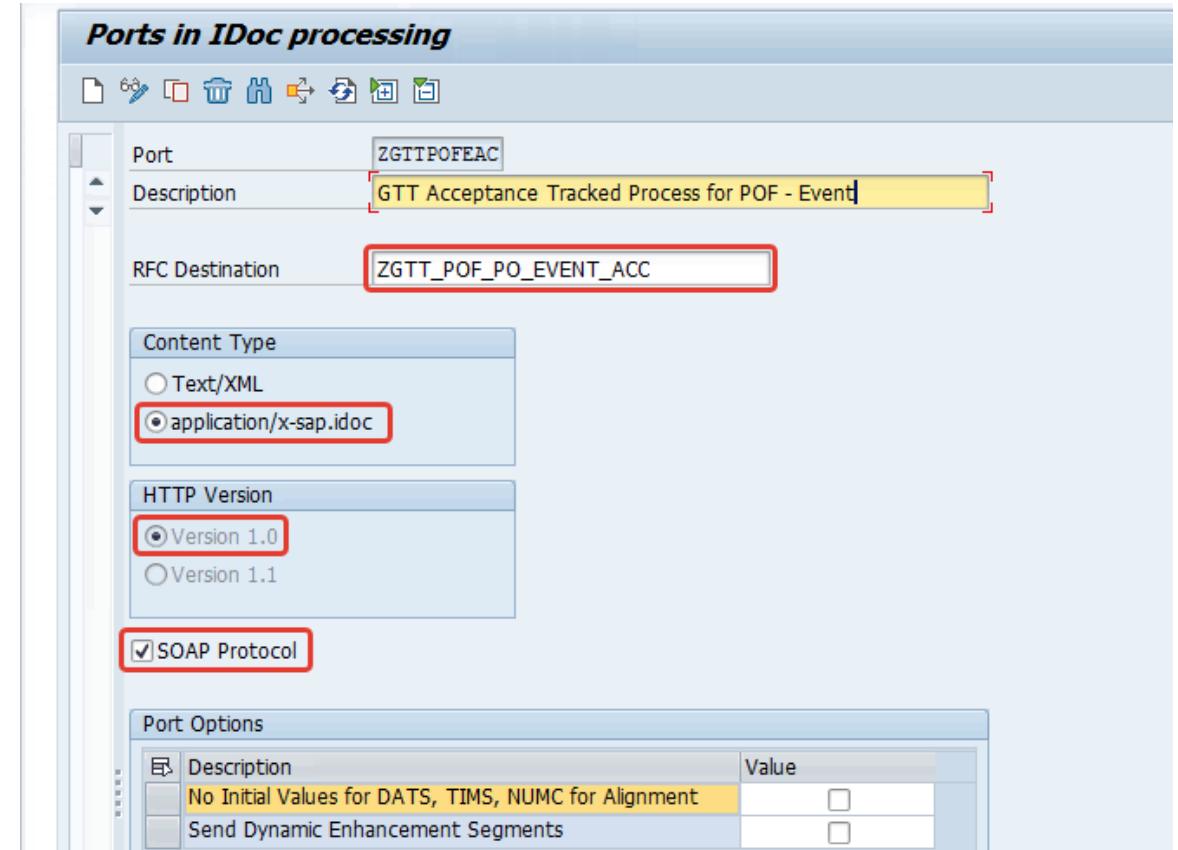
3-5: Choose **Content Type** as *application/x-sap.idoc*

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

3-7: Mark it as SOAP Protocol

3-8: Save the configuration

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

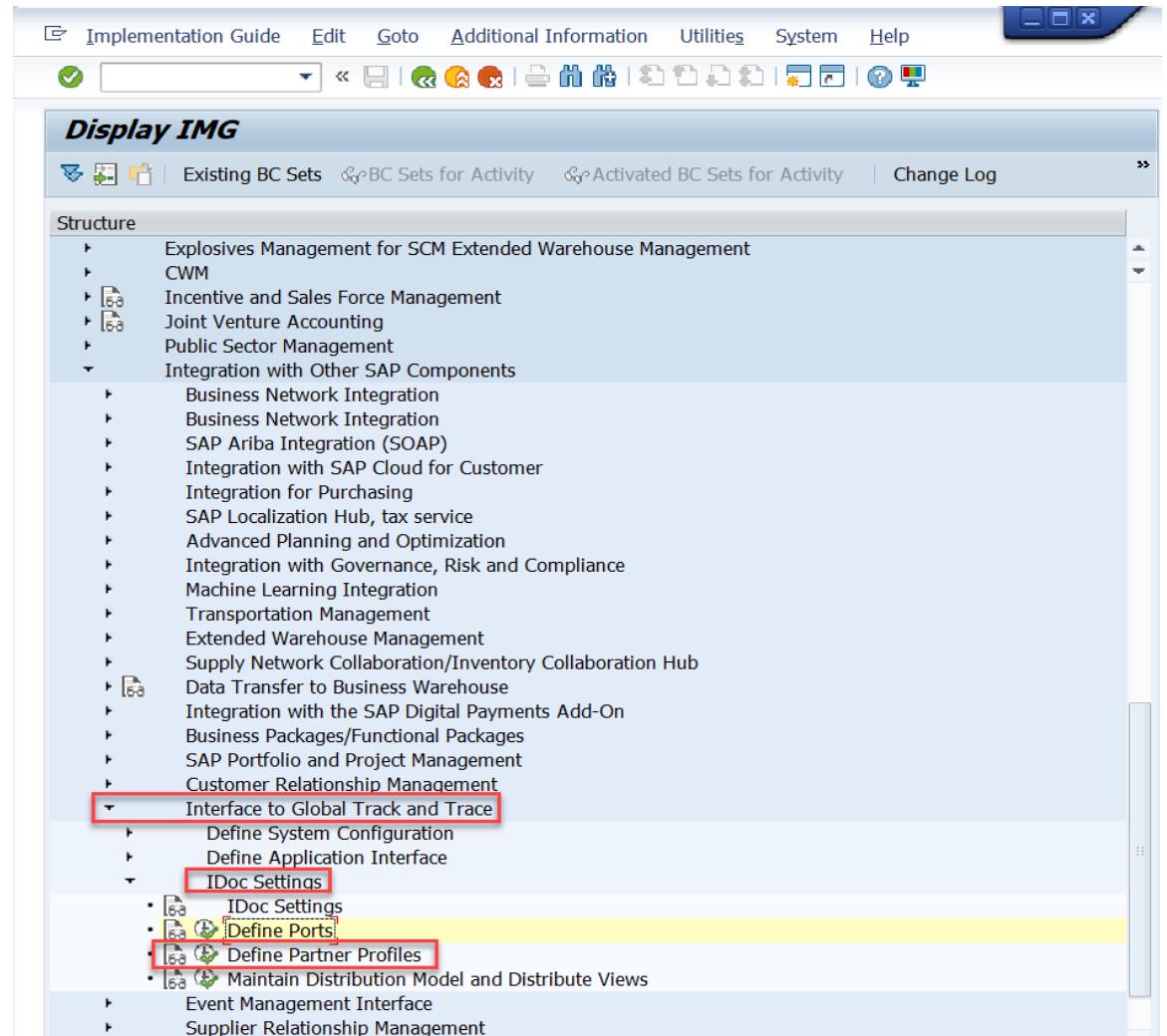


Port	Description	RFC Destination	Content Type	HTTP Version	SOAP Protocol
ZGTTPOFEAC	GTT Acceptance Tracked Process for POF - Event	ZGTT_POF_PO_EVENT_ACC	application/x-sap.idoc	Version 1.0	Checked
ZGTTPOFTAC	GTT Acceptance Tracked Process for POF - TP	ZGTT_POF_PO_TP_ACC	application/x-sap.idoc	Version 1.0	Checked

STEP 4: Define Partner Profiles

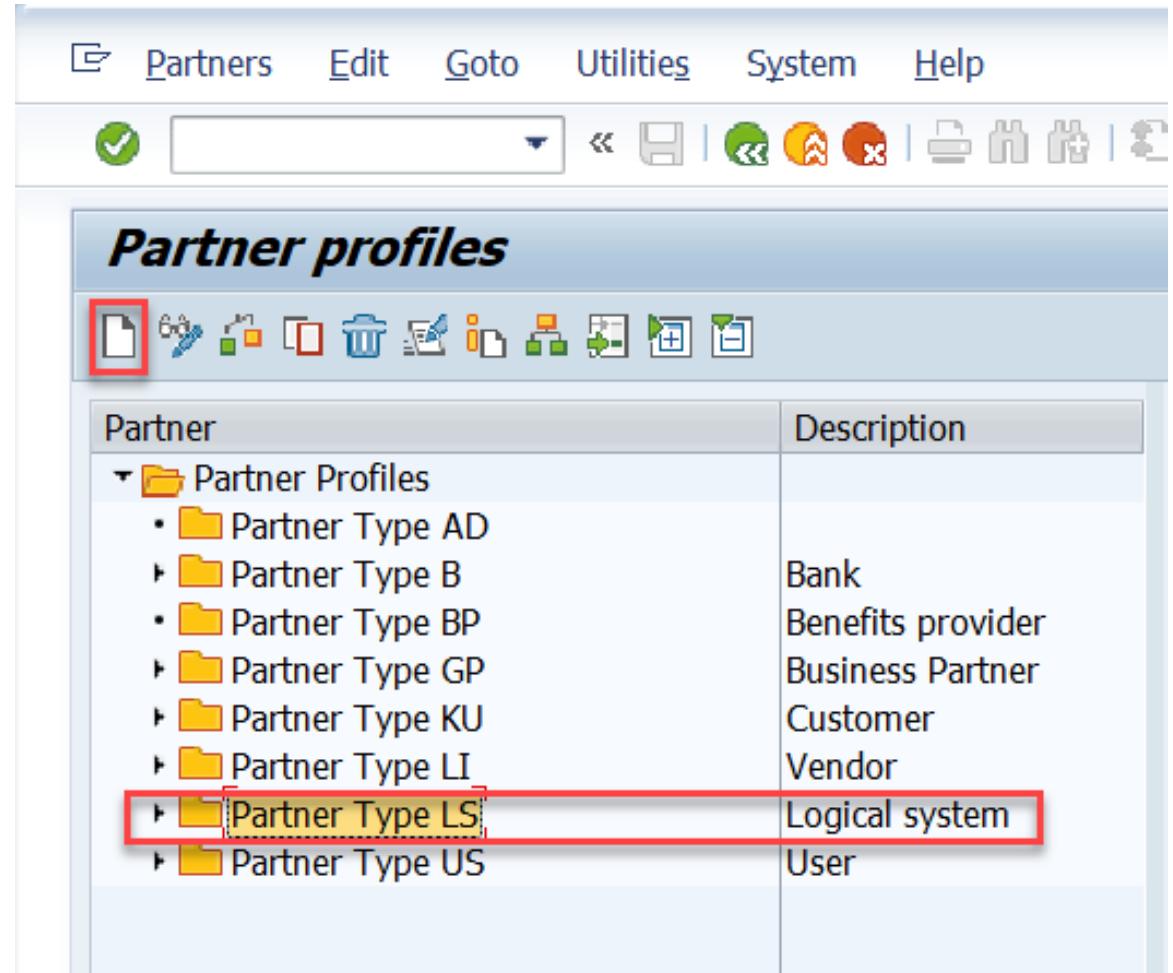
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

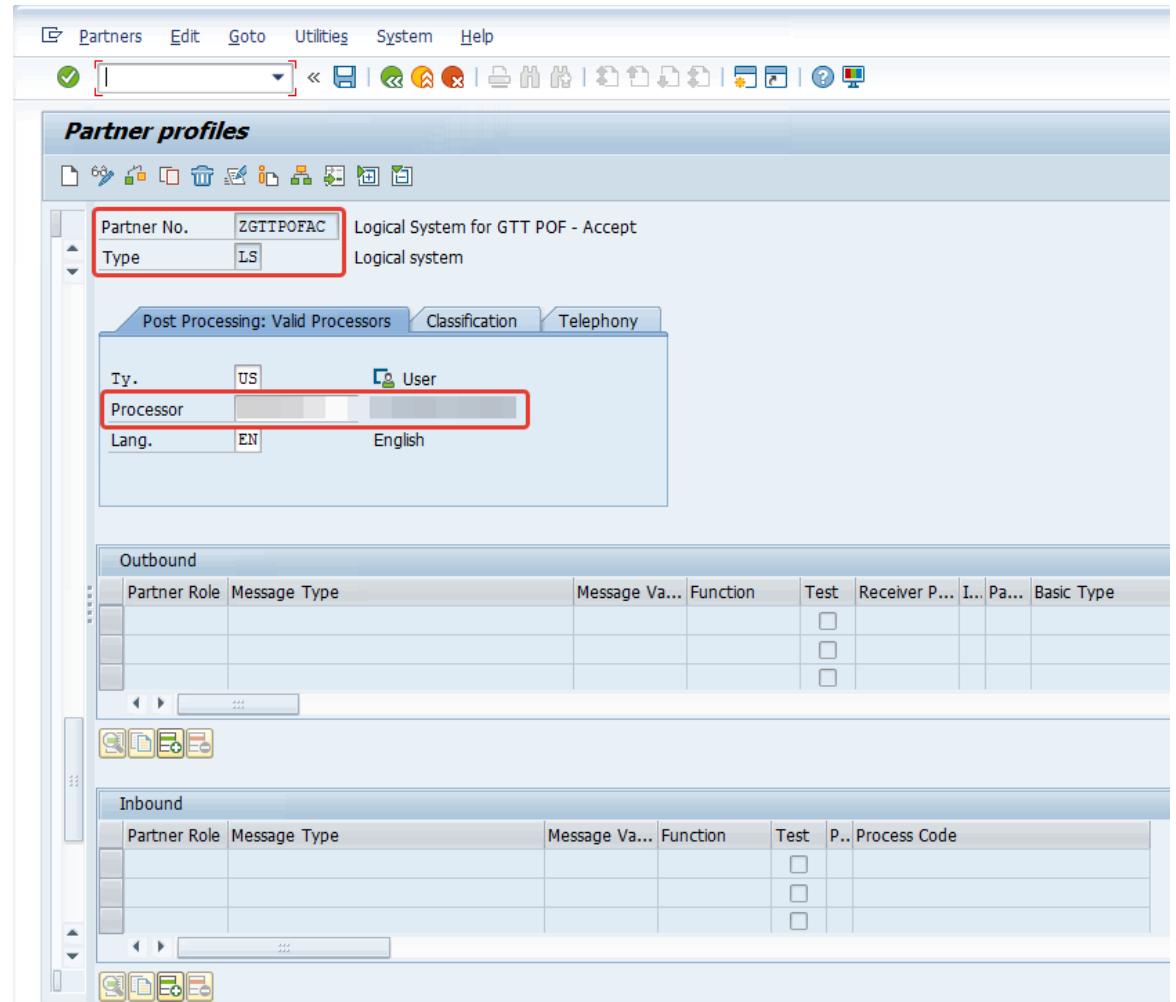
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

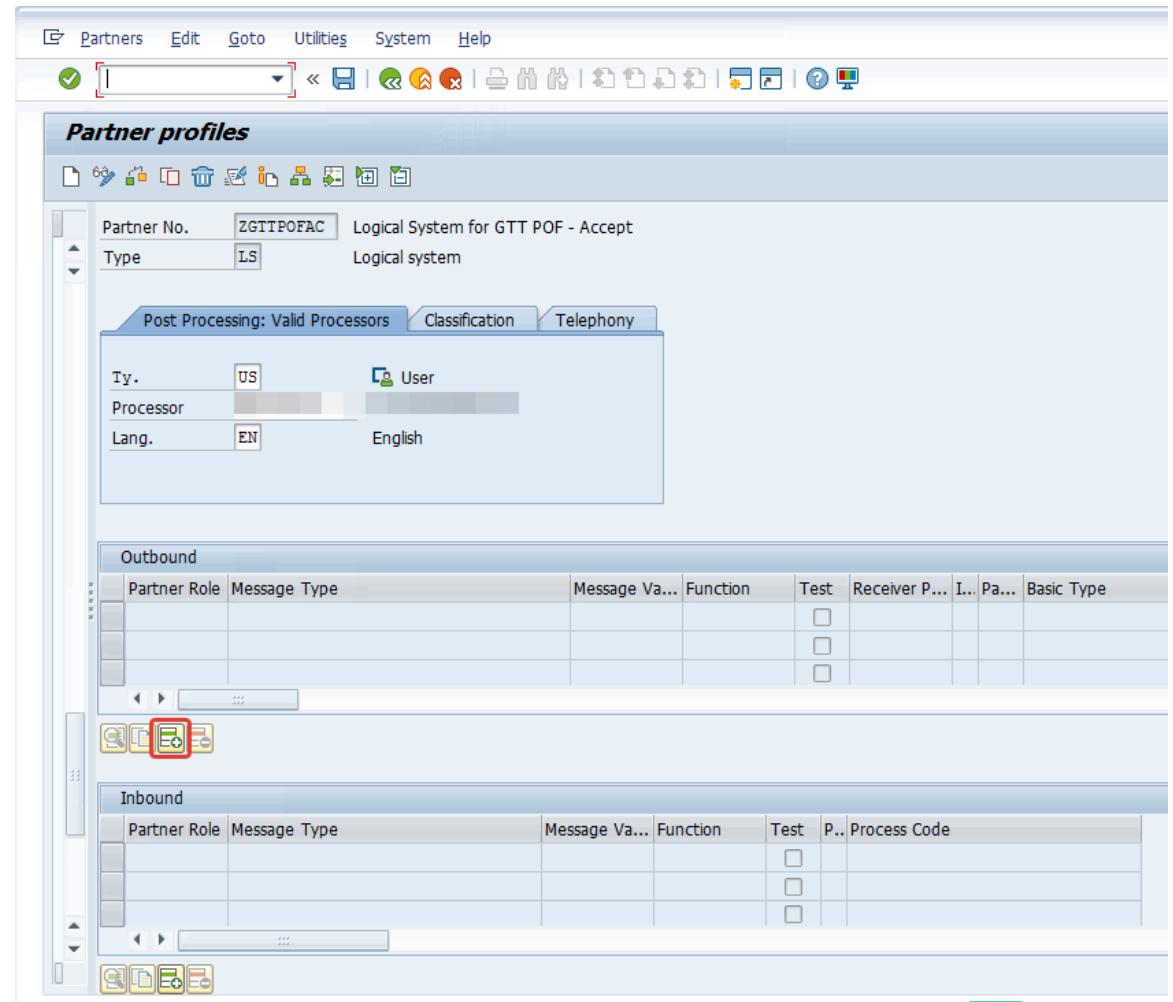
4-4: Fill in the **Partner No.** that you created in STEP 2

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



STEP 4: Define Partner Profiles

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



STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

For the tracked Process:

Message Type: AOPOST

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

4-9: Save the configuration

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

Partner profiles: Outbound parameters

Partner No.	ZGTTPOFAC	Logical System for GTT POF - Accept
Type	LS	Logical system
Partner Role		
Message Type	EVMSTA	
Message Code		
Message Function		<input type="checkbox"/> Test
Outbound Options		
Receiver Port	ZGTTPOFEAC	GTT Acceptance Tracked Proc...
Pack. Size		
<input type="checkbox"/> Queue Processing		
Output Mode		
<input checked="" type="radio"/> Pass IDoc Immediately		Output Mode 2
<input type="radio"/> Collect IDocs		
IDoc Type		
Basic Type	EVMSTA02	SCEM: Event Message Input
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		<input type="checkbox"/> Application Release

STEP 4: Define Partner Profiles

4-10: Fill in the Message Type.

For the tracked Process:

Message Type: AOPOST

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

4-12: Save the configuration

Partner No.	Type	Outbound	Message Type	Receiver Port	IDoc Type
ZGTTPOFAC	LS	Yes	AOPOST	ZGTTPOFTAC	EHPOST01
ZGTTPOFAC	LS	Yes	EVMSTA	ZGTTPOFEAC	EVMSTA02

Partner profiles: Outbound parameters

Partner No.	ZGTTPOFAC	Logical System for GTT POF - Accept
Type	LS	Logical system
Partner Role		
Message Type	AOPOST	
Message Code		
Message Function		
<input type="checkbox"/> Test		

Outbound Options

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

IDoc Type

Basic Type	EHPOST01	SCEM: Event Handler Posting
Extension		
View		
<input checked="" type="checkbox"/> Cancel Processing After Syntax Error		
Seg. release in IDoc type		
Application Release		

B) Configuration and Implementation

- Basic

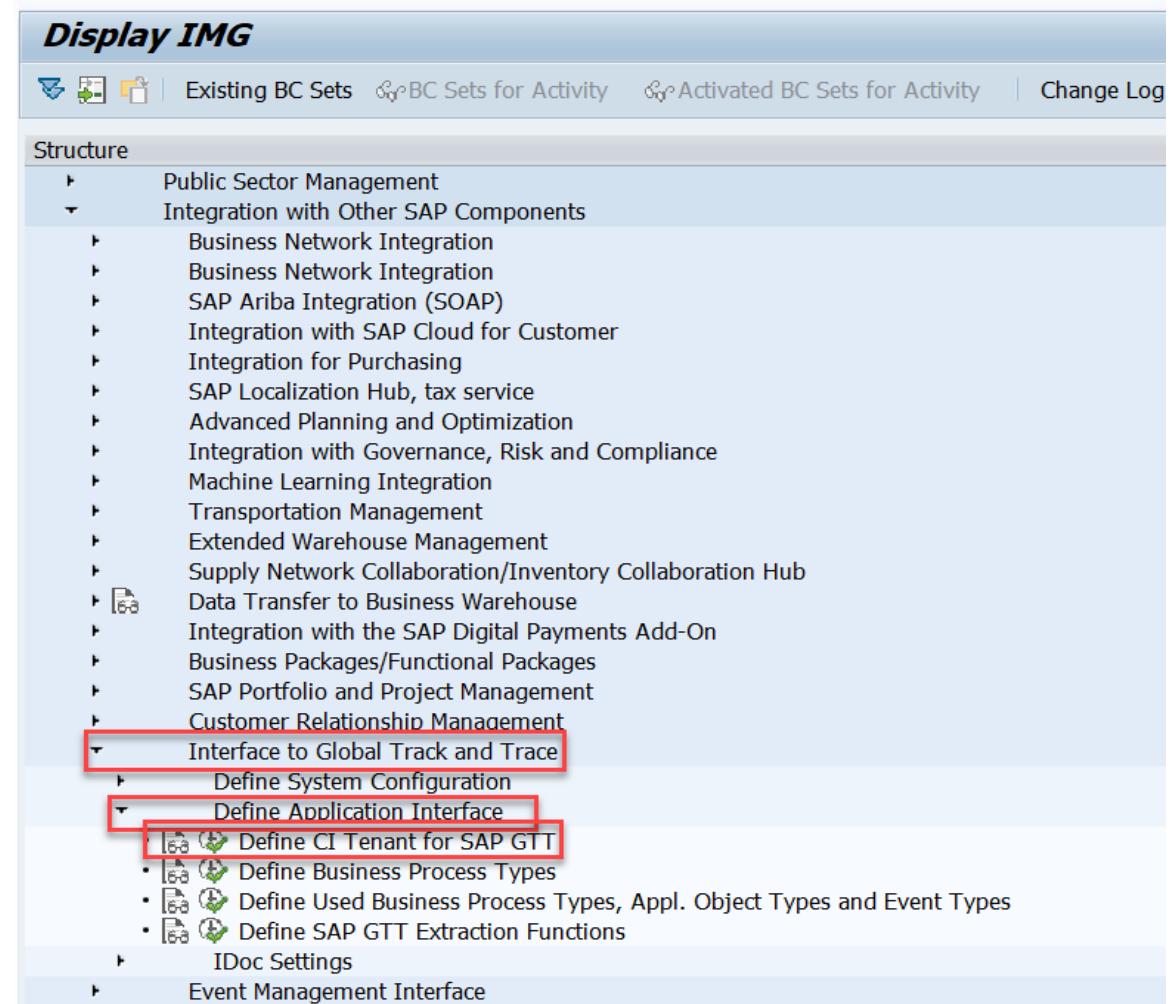
B2. Extractor Configuration



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

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

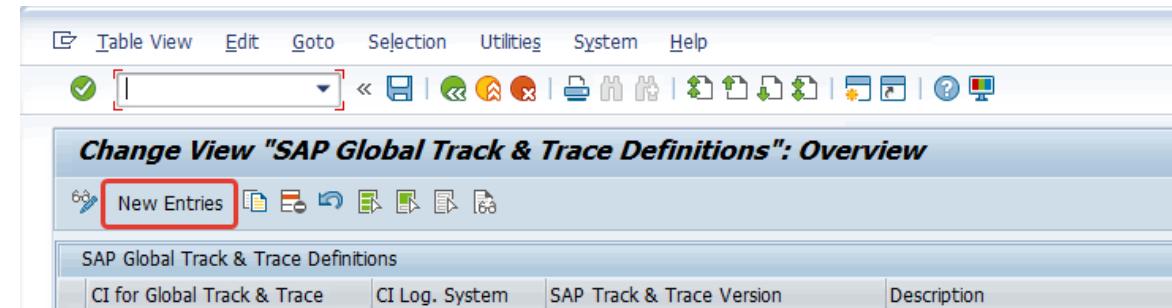
5-2: Choose activity
Define CI Tenant for SAP GTT



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

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

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



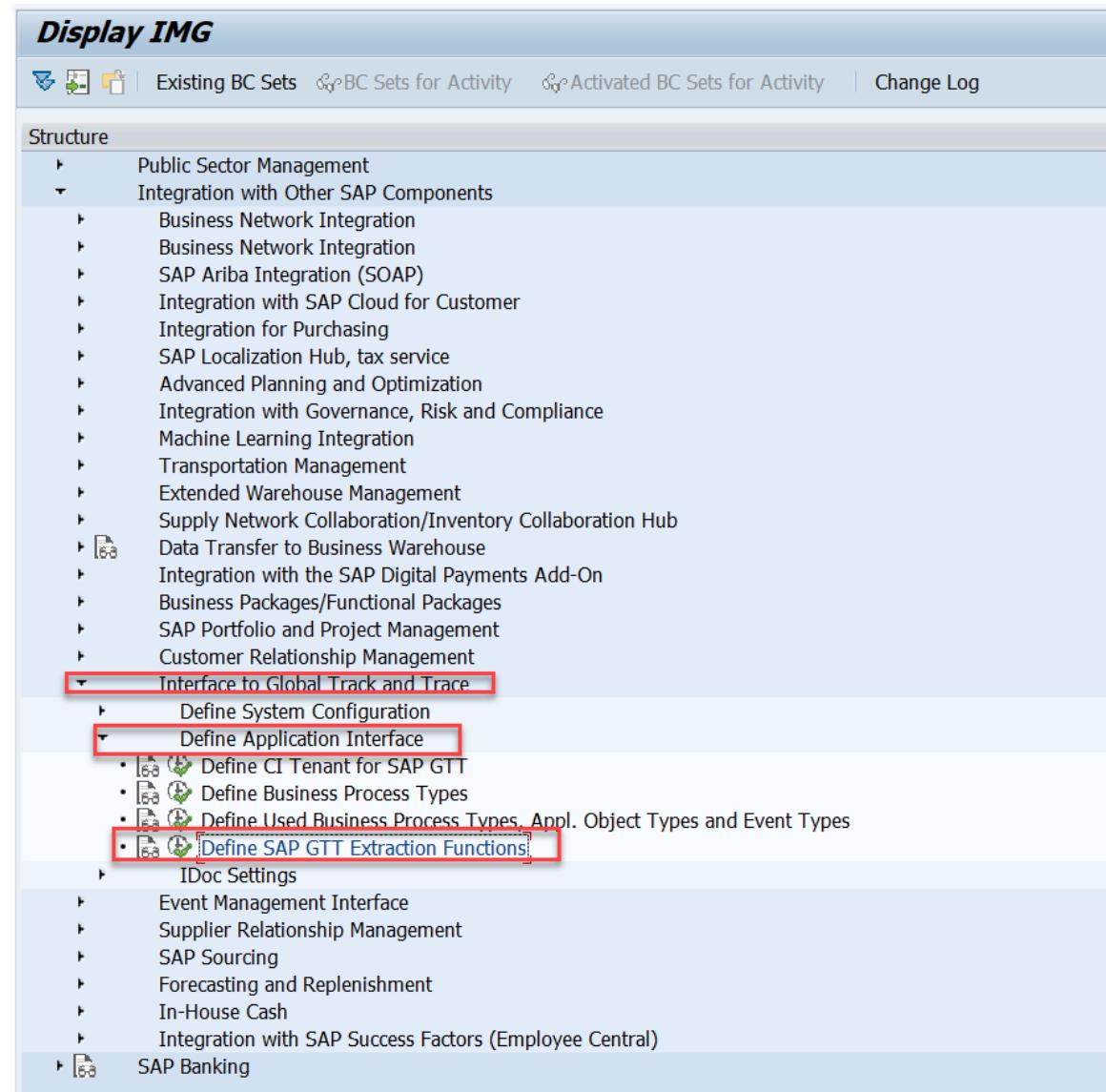
The screenshot shows the SAP Global Track & Trace Definitions overview screen after a new entry has been created. The top navigation bar and toolbar are identical to the previous screenshot. The main title is 'Change View "SAP Global Track & Trace Definitions": Overview'. The 'New Entries' button is visible in the toolbar. The table below now contains one row, which is highlighted with a red box. The row details a new CI entry: 'ZGTTPOFAC' in the CI for Global Track & Trace column, 'ZGTTPOFAC' in the CI Log. System column, 'GTT1.0 Global Track & Trace' in the SAP Track & Trace Version column, and 'CI For GTT Purchasing Order Sample APP - Acceptance' in the Description column.

CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTPOFAC	ZGTTPOFAC	GTT1.0 Global Track & Trace	CI For GTT Purchasing Order Sample APP - Acceptance

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

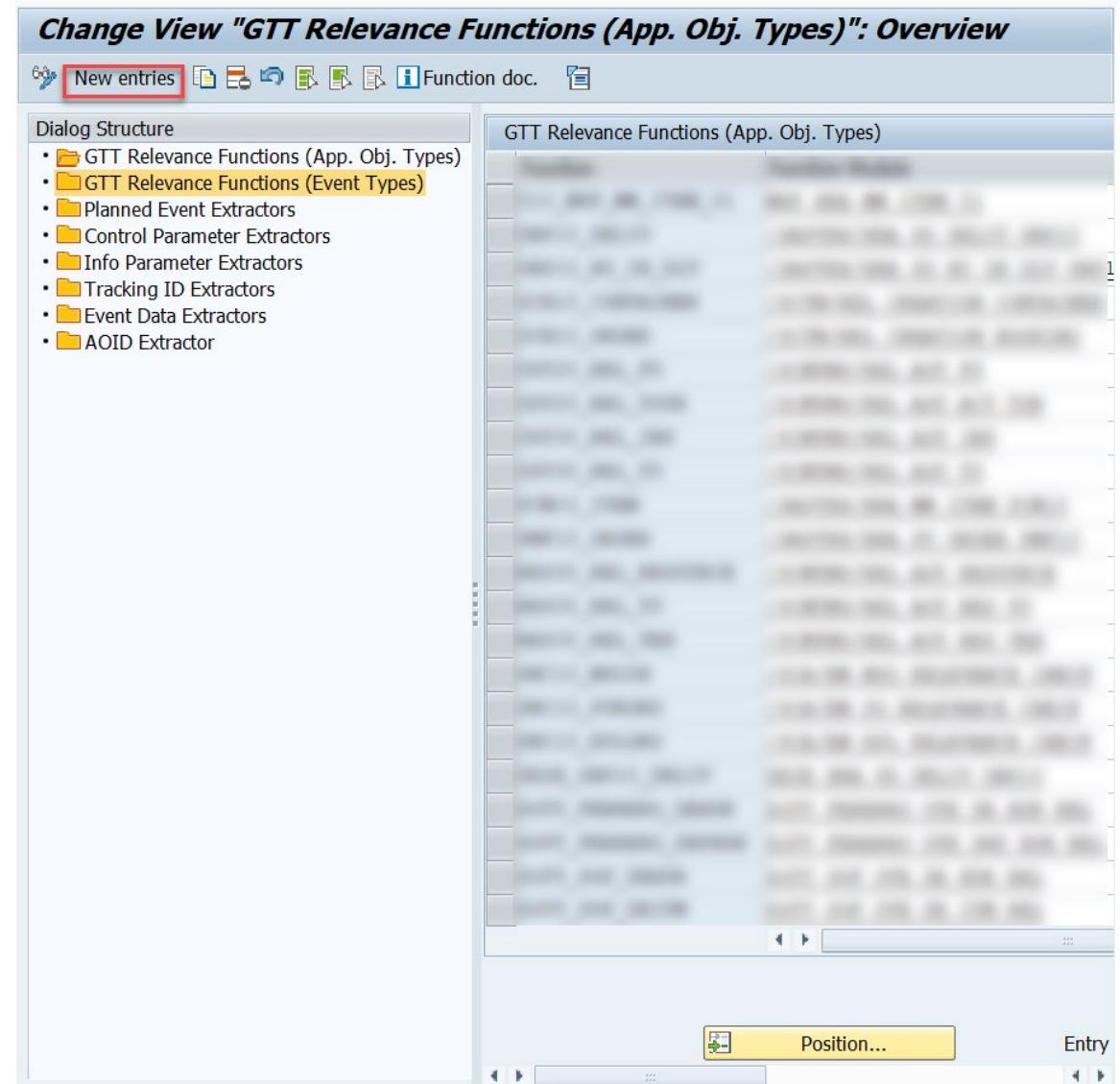


The screenshot shows the SAP Display IMG interface. The top navigation bar includes icons for search, existing BC sets, BC sets for activity, activated BC sets for activity, and change log. Below the navigation is a tree structure under the heading "Structure". The path taken is: Public Sector Management > Integration with Other SAP Components > Interface to Global Track and Trace > Define Application Interface > Define SAP GTT Extraction Functions. The "Define SAP GTT Extraction Functions" node is highlighted with a red box.

- Public Sector Management
- Integration with Other SAP Components
 - Business Network Integration
 - 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
 - Define Used Business Process Types, Appl. Object Types and Event Types
 - Define SAP GTT Extraction Functions
- IDoc Settings
- Event Management Interface
- Supplier Relationship Management
- SAP Sourcing
- Forecasting and Replenishment
- In-House Cash
- Integration with SAP Success Factors (Employee Central)
- SAP Banking

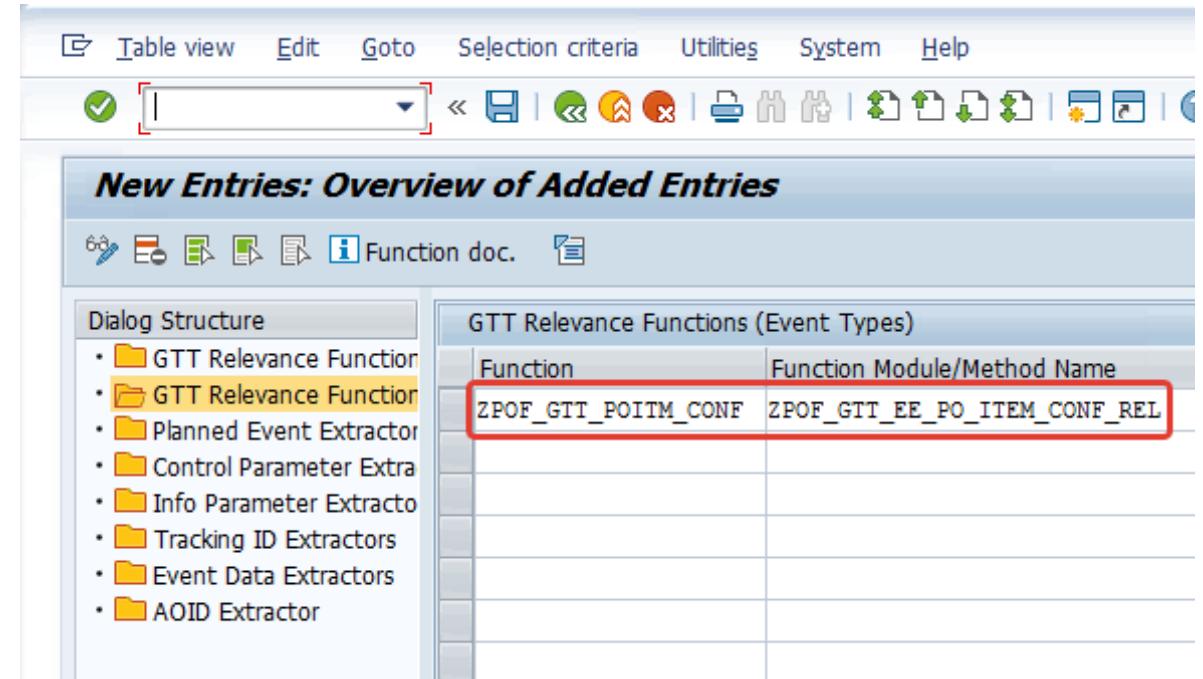
STEP 6: Define GTT Extraction Functions

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



STEP 6: Define GTT Extraction Functions

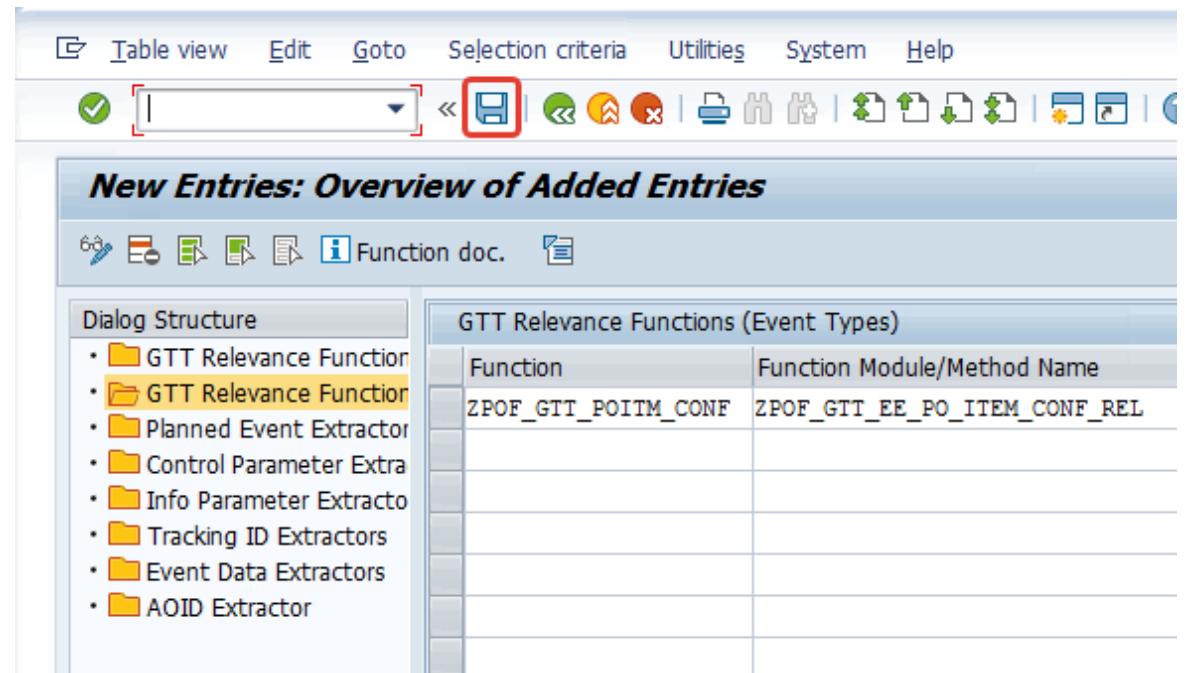
6-4: Input the **Function name** and **Function Module** for the newly created extraction function



GTT Relevance Functions (Event Types)	
Function	Function Module/Method Name
ZPOF_GTT_POITM_CONF	ZPOF_GTT_EE_PO_ITEM_CONF_REL

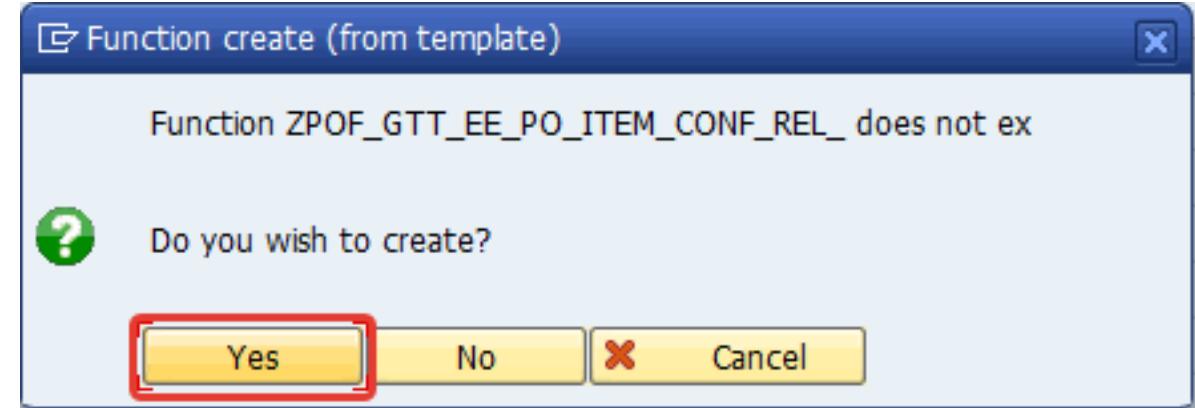
STEP 6: Define GTT Extraction Functions

6-5: Click **Save**



STEP 6: Define GTT Extraction Functions

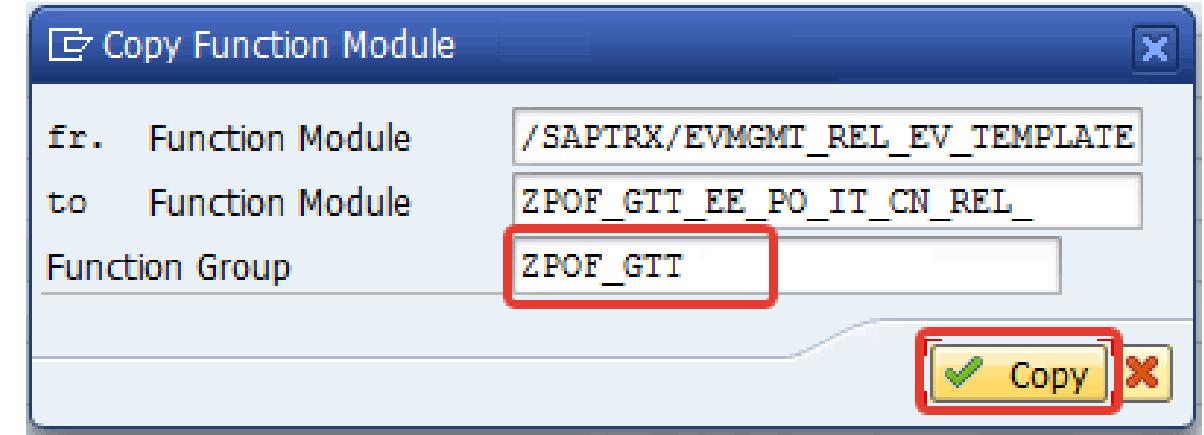
6-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



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.

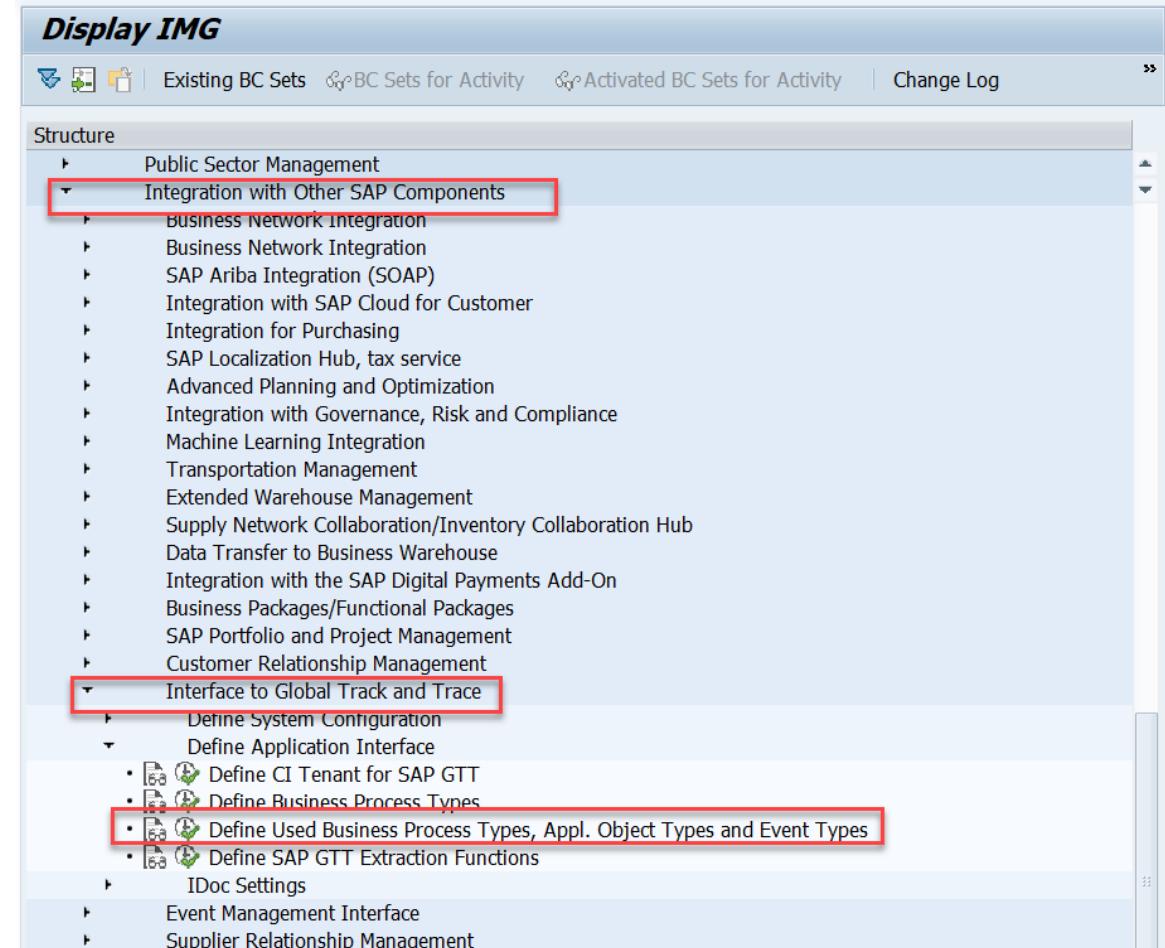
The screenshot shows the SAP SE80 Function Builder interface. The title bar reads "Function Builder: Display ZPOF_GTT_EE_PO_IT_CN_REL_". The left pane is a "Repository Browser" with a dropdown menu set to "Function Group" and the value "ZPOF_GTT" highlighted with a red box. Below it is an "Object Name" tree view under "Function Modules" containing various function names like ZPOF_GTT_CTP_DL_TO_PO, ZPOF_GTT_EE_DL_HDR, etc. A specific function, ZPOF_GTT_EE_PO_IT_CN_REL_, is highlighted with a yellow box. The right pane displays the source code for the selected function:

```
1 FUNCTION ZPOF_GTT_EE_PO_IT_CN_REL_.  
2  
3 *--> Local Interface:  
4 IMPORTING  
5   REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM  
6   REFERENCE(I_EVENT_TYPES) TYPE /SAPTRX/EVTYPES  
7   REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER  
8   REFERENCE(I_EVENTTYPE_TAB) TYPE TRXAS_EVENTTYPE_TABS_WA  
9   REFERENCE(I_EVENT) TYPE TRXAS_EVT_CTAB_WA  
10 EXPORTING  
11   VALUE(E_RESULT) LIKE SY-BINPT  
12   TABLES  
13     C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL  
14 EXCEPTIONS  
15   PARAMETER_ERROR  
16   RELEVANCE_DETERM_ERROR  
17   STOP_PROCESSING  
18  
19 *-->  
20 * Top Include  
21 * TYPE-POOLS:trxas.  
22  
23  
24  
25  
ENDFUNCTION.
```

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

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

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



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

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

In the following:

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

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

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

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

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

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

The screenshot shows the SAP GUI interface for defining event types. The title bar reads "Change View 'Define Event Types': Overview". On the left, there's a toolbar with various icons and a "New Entries" button highlighted with a red box. Below the toolbar is a "Dialog Structure" tree view with nodes like "Define Used Business Pro", "Define Application Ot", and "Define Event Types". The main area is a table titled "Define Event Types" with columns: Business Process Type, Event Type, and Description. The table contains three rows:

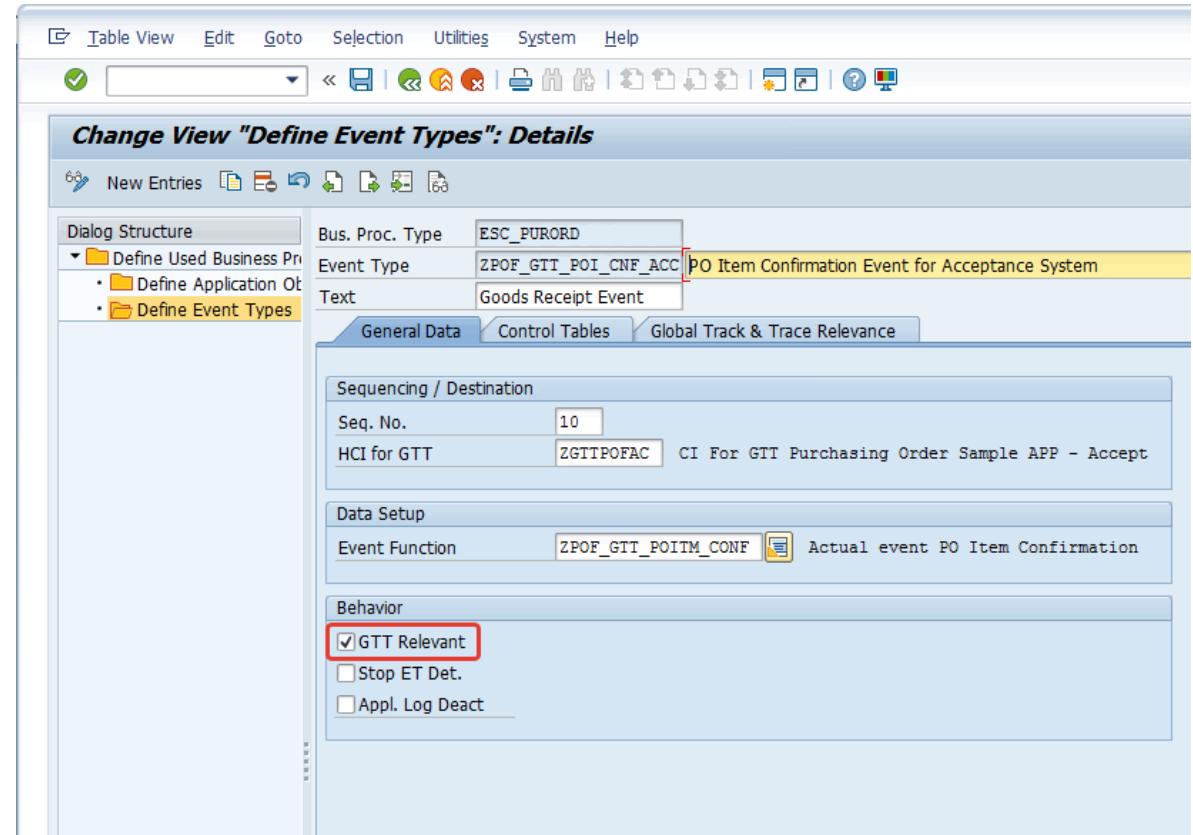
Business Process Type	Event Type	Description
ESC_PURORD	ZPOF_GTT_POI_DEL_ACC	PO Item Confirmation Event for Acceptance System
ESC_PURORD	ZPOF_GTT_POI_DEL_ACZ	PO Item Confirmation Event for Azure System
ESC_PURORD	ZPOF_GTT_POI_DEL_INT	PO Item Confirmation Event for Integration System

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**



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

Bus. Proc. Type	ESC_SHIPMT	
Event Type	ZPOF_GTT_SHH_ARR_ACC	Shipment Header Arrival Event for Acceptance System
Text	Arrival Event	
General Data Control Tables Global Track & Trace Relevance		
Data Source for Events		
Main Obj. Table	SHIPMENT_HEADER_NEW	Event on Header Level
Master Table		
Old Main Obj. Table	SHIPMENT_HEADER_OLD	
Old Master Table		

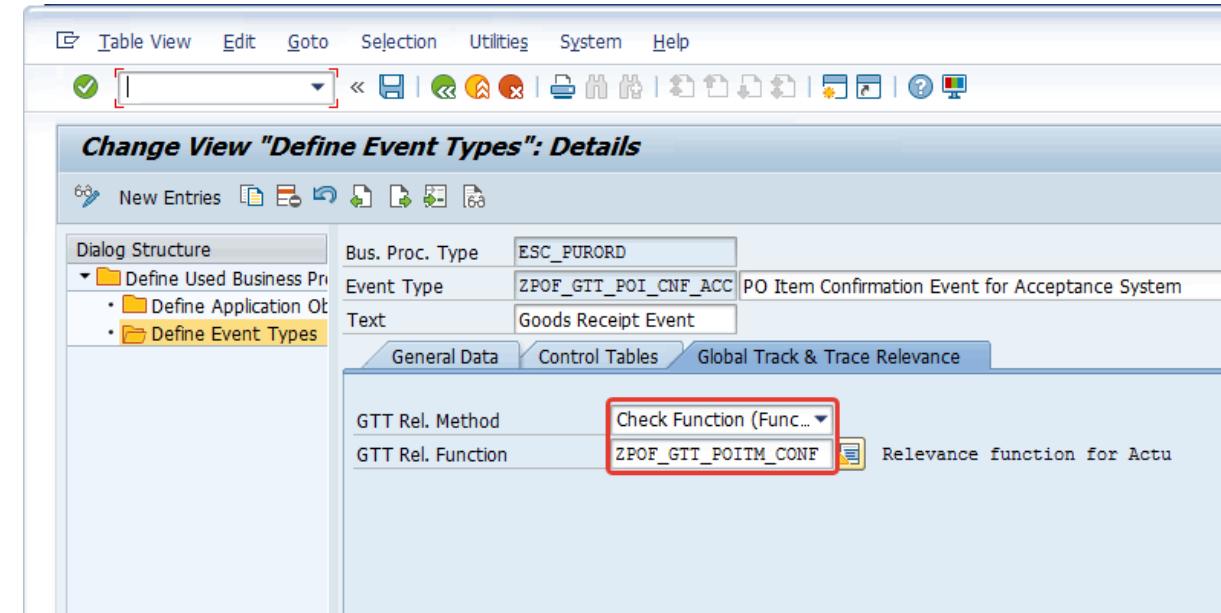
Bus. Proc. Type	ESC_PURORD	
Event Type	ZPOF_GTT_POT_CNF_ACC	PO Item Confirmation Event for Acceptance System
Text	Goods Receipt Event	
General Data Control Tables Global Track & Trace Relevance		
Data Source for Events		
Main Obj. Table	PURCHASE_ITEM_NEW	Event on Item Level
Master Table	PURCHASE_ORDER_HEADER_NEW	
Old Main Obj. Table	PURCHASE_ITEM_OLD	
Old Master Table	PURCHASE_ORDER_HEADER_OLD	
Reference Between Main and Master Table		
First Field Reference from Main to Master Table		
Uplink Field	EBELN	Uplink Mode <input checked="" type="checkbox"/>
Uplink Target Fld	EBELN	Uplink Const

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

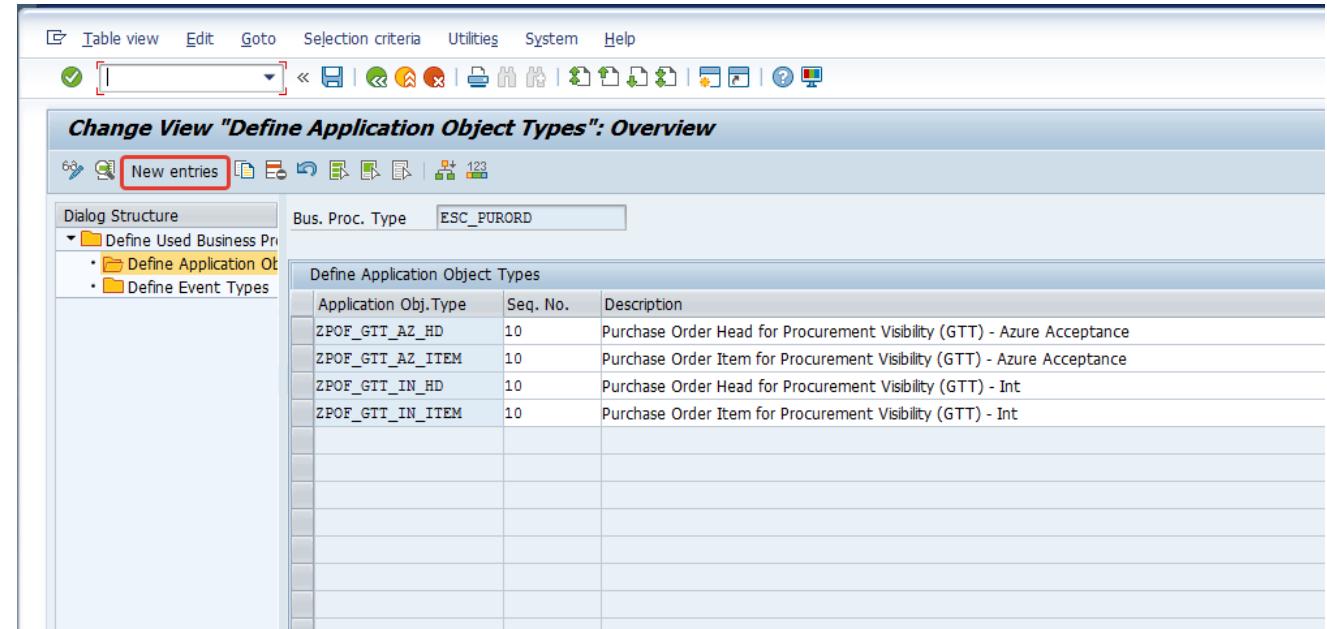
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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

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

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



The screenshot shows the SAP GUI interface for defining application object types. The title bar reads "Change View 'Define Application Object Types': Overview". The toolbar has various icons, and the "New entries" button is highlighted with a red box. The left sidebar shows a tree structure with "Define Used Business Pro" expanded, revealing "Define Application Obj" which is also highlighted with a red box. The main area displays a table titled "Define Application Object Types" with columns "Application Obj.Type", "Seq. No.", and "Description". The table contains four entries:

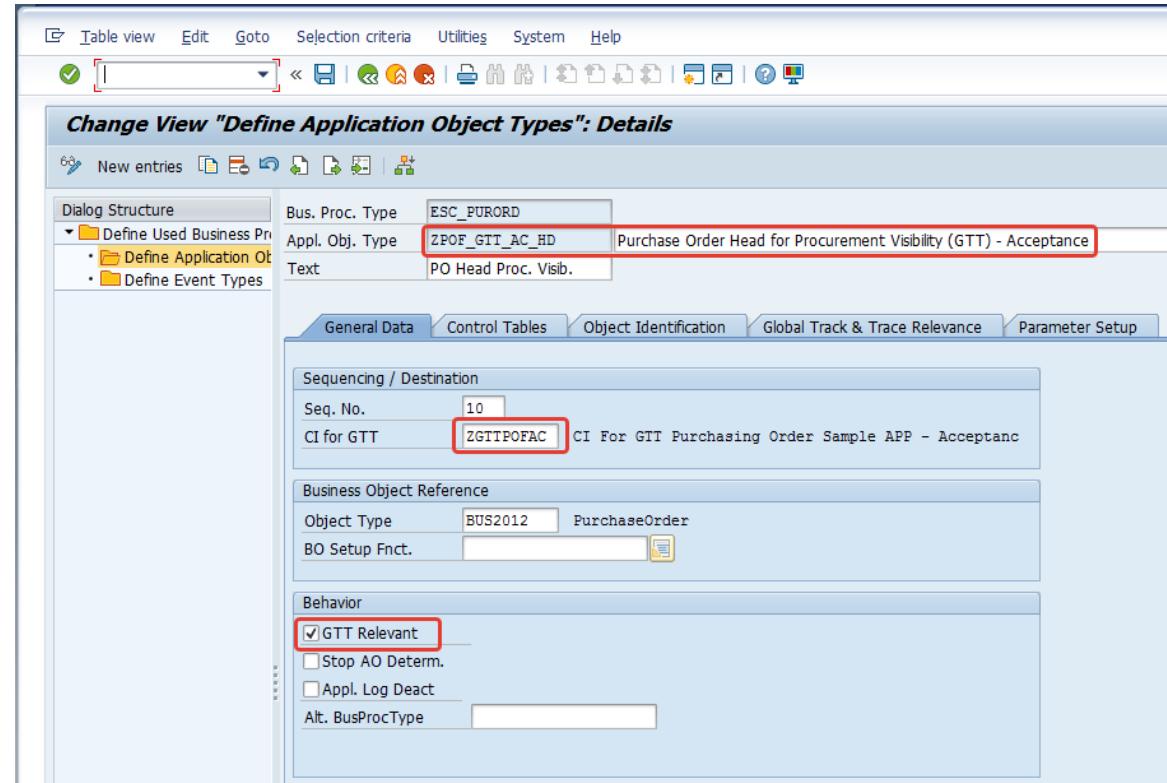
Application Obj.Type	Seq. No.	Description
ZPOF_GTT_AZ_HD	10	Purchase Order Head for Procurement Visibility (GTT) - Azure Acceptance
ZPOF_GTT_AZ_ITEM	10	Purchase Order Item for Procurement Visibility (GTT) - Azure Acceptance
ZPOF_GTT_IN_HD	10	Purchase Order Head for Procurement Visibility (GTT) - Int
ZPOF_GTT_IN_ITEM	10	Purchase Order Item for Procurement Visibility (GTT) - Int

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**

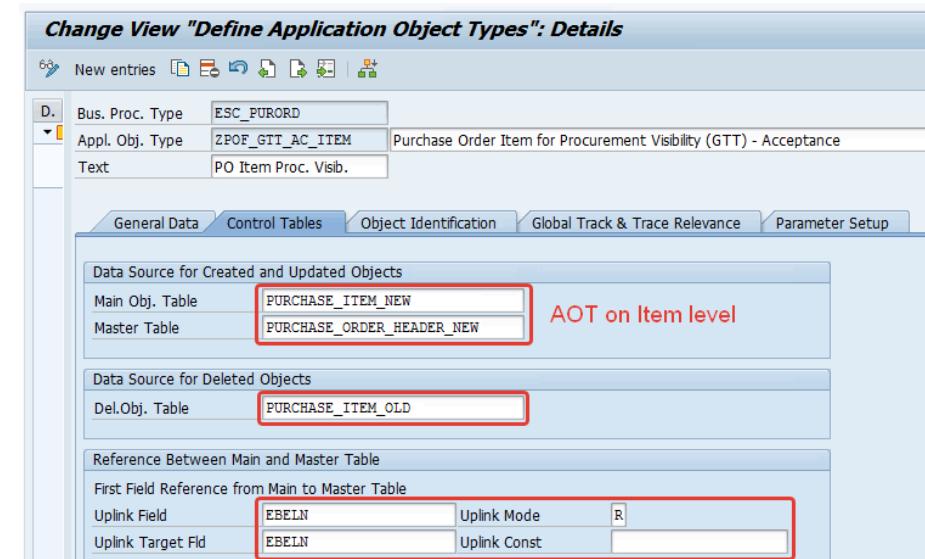
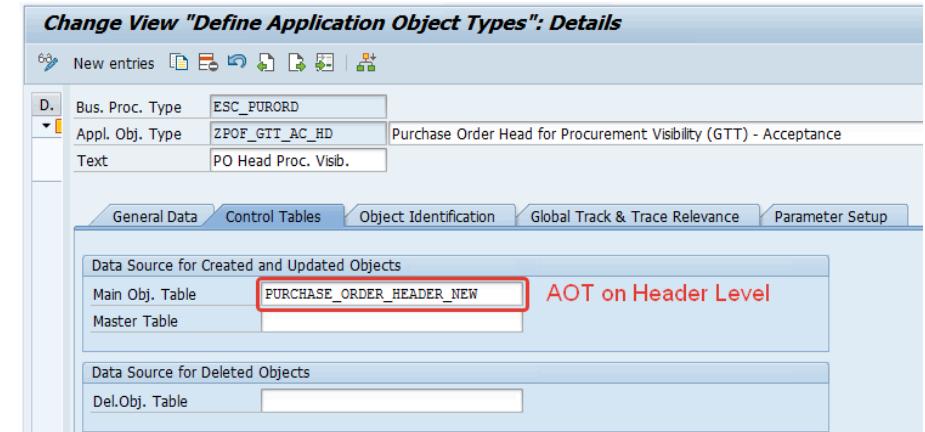


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

Caution:

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



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* and use the key field to fill in the AO ID fields.

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

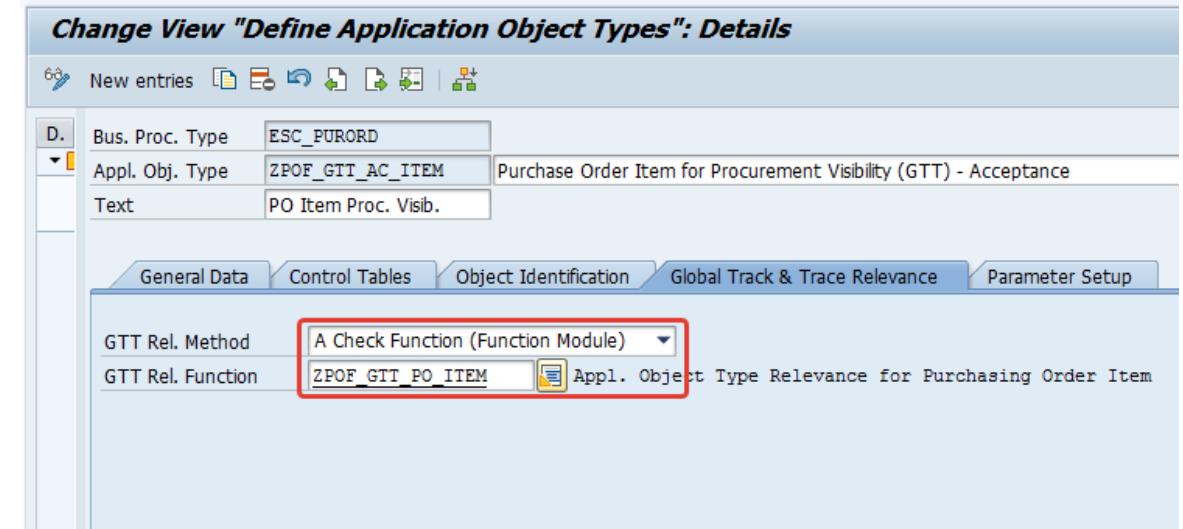
Change View "Define Application Object Types": Details

New entries	Save	Cancel	Print	Help	+
D.	Bus. Proc. Type	ESC_PURORD	Purchase Order Item for Procurement Visibility (GTT) - Acceptance		
	Appl. Obj. Type	ZPOF_GTT_AC_ITEM			
	Text	PO Item Proc. Visib.			
General Data Control Tables Object Identification Global Track & Trace Relevance Parameter Setup					
Method for determination of AOID					
AOID Method	<input type="button" value="Determine from Field"/>				
Application Object ID Source					
First Field to Build Appl. Obj. ID	Cntrl Tab. Type	1 Main Object Table			
	AO ID Field	EBELN			
Second Field to Build Appl. Obj. ID	Cntrl Tab. Type	1 Main Object Table			
	AO ID Field	EBELP			
Determine AOID By Function					
AOID Function	<input type="button" value=""/>				

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**, you need to define a relevance function according to STEP 6, and fill in the relevance function name here.



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

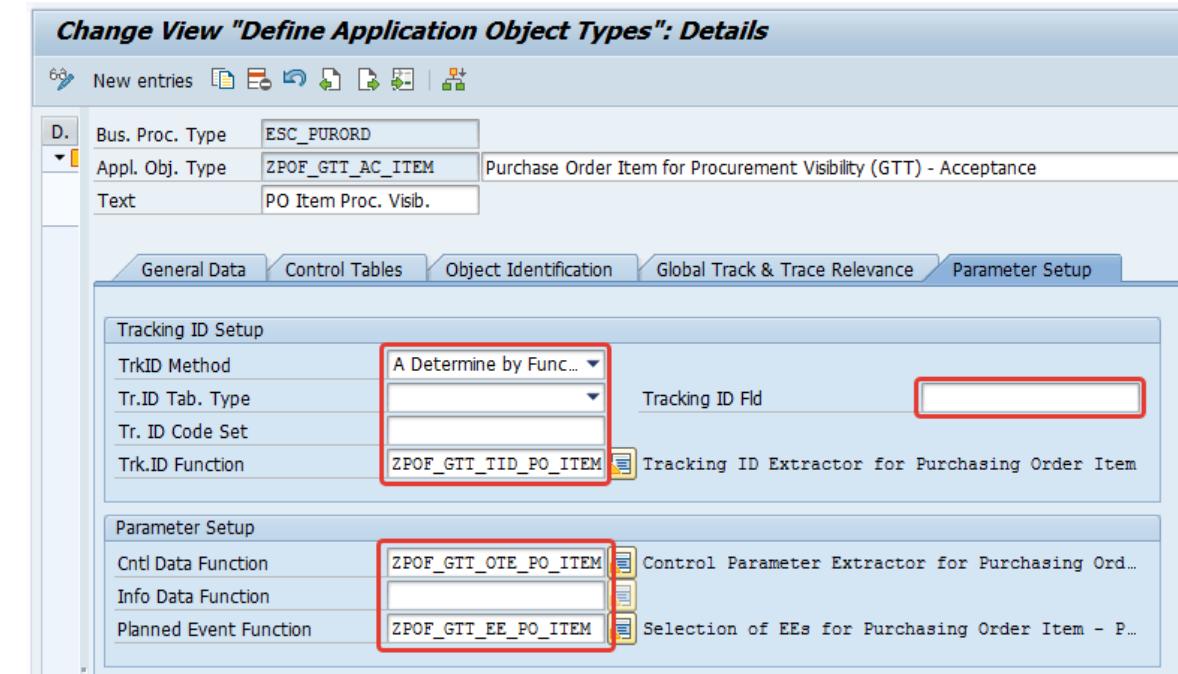
7-21: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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 fill in the key field and name the Code Set for the AOT.

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

Click **Save**.



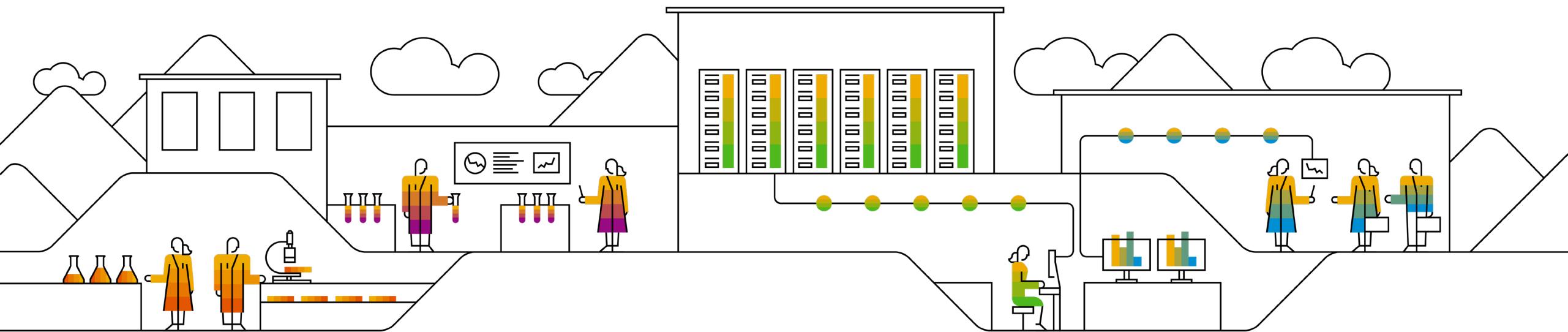
C) Download ABAP Code from GitHub

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

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

C3. Download Another ABAP Code from GitHub (TSOF)

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 TPOF)



STEP 1: Install ABAPGit

You need to install ABAPGit before downloading codes from GitHub.

To install ABAPGit, follow the instructions at <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**.

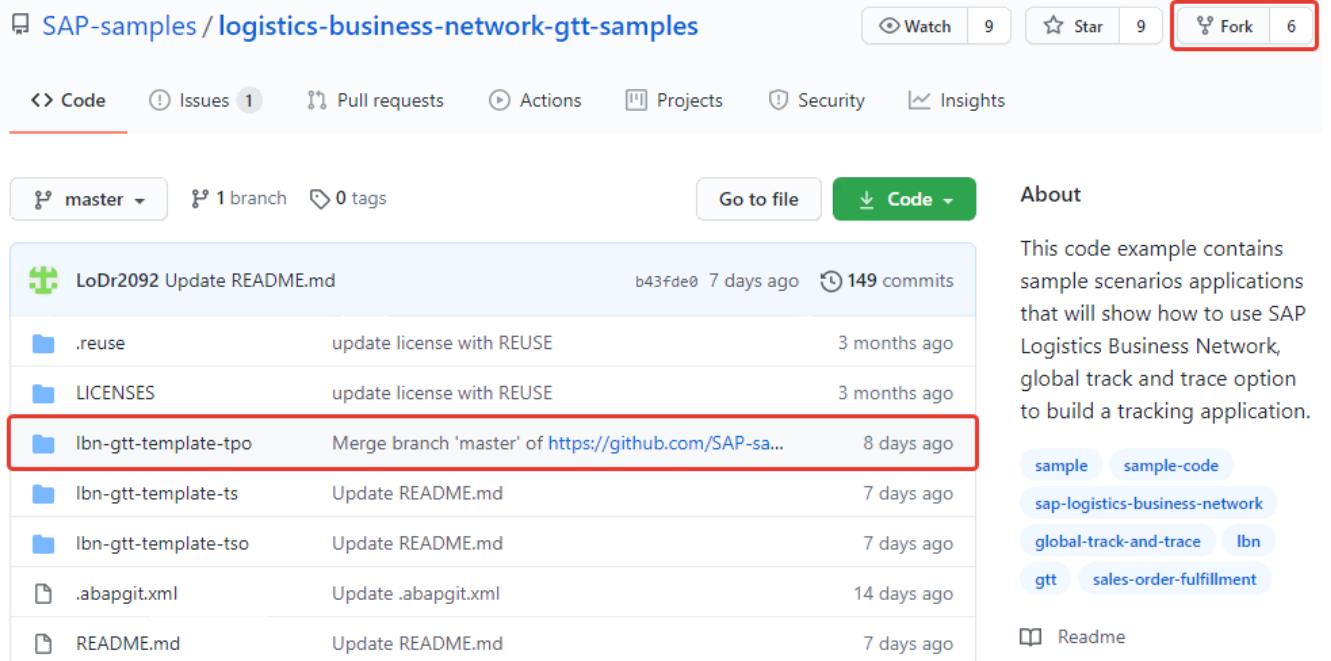
Note: ABAPGit version 1.105.0 is used to create this guide. In case you use different version of ABAPGit, you may face distinctions in interface of the app.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar has sections for "Getting Started" (Installation, Upgrading, Uninstalling, UI features), "Setup" (SSL setup, Proxy configuration, Development version), "Online Projects" (Installing online repo, Keeping code up to date, Uninstall repository, First project, Moving package into git, Contributing to a project), "Offline Projects" (Import zip, Export zip), and "Reference" (Repo Settings (abapgit.xml), Supported object types, Icon Legend, User Exits, Authorizations, Namespaces). The main content area starts with a "Summary" section stating that abapGit exists in two flavours: standalone or developer version. It then describes the standalone version as targeted for users and the developer version as targeted for developers. Below this is a "Prerequisites" section mentioning SAP BASIS version 702 or higher. The "Install standalone version" section is highlighted with a red border and contains steps: 1. Download the ABAP code (right click -> save-as) to a file. 2. Via SE38 or SE80, create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version. 3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload. 4. Activate. A note below says typically abapGit will only be used in the development system so it can be installed in a local \$ package (e.g. \$ZABAPGIT). A final note says now you can use abapGit by executing the report in transaction SE38.

STEP 2: 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 code into the user’s account and meanwhile it will navigate to user’s own repository.



The screenshot shows a GitHub repository page for 'SAP-samples / logistics-business-network-gtt-samples'. The top navigation bar includes 'Code', 'Issues 1', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. Below the navigation is a dropdown for 'master', showing '1 branch' and '0 tags'. A green 'Code' button is visible. The main area displays a list of commits:

Commit	Message	Date
	LoDr2092 Update README.md	b43fde0 7 days ago
	.reuse update license with REUSE	3 months ago
	LICENSES update license with REUSE	3 months ago
	Ibn-gtt-template-tpo Merge branch 'master' of https://github.com/SAP-sa...	8 days ago
	Ibn-gtt-template-ts Update README.md	7 days ago
	Ibn-gtt-template-tso Update README.md	7 days ago
	.abapgit.xml Update .abapgit.xml	14 days ago
	README.md Update README.md	7 days ago

To the right of the commit list is a sidebar with 'About' text: '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 sidebar are several blue buttons with white text: 'sample', 'sample-code', 'sap-logistics-business-network', 'global-track-and-trace', 'Ibn', 'gtt', and 'sales-order-fulfillment'. At the bottom right is a 'Readme' button.

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

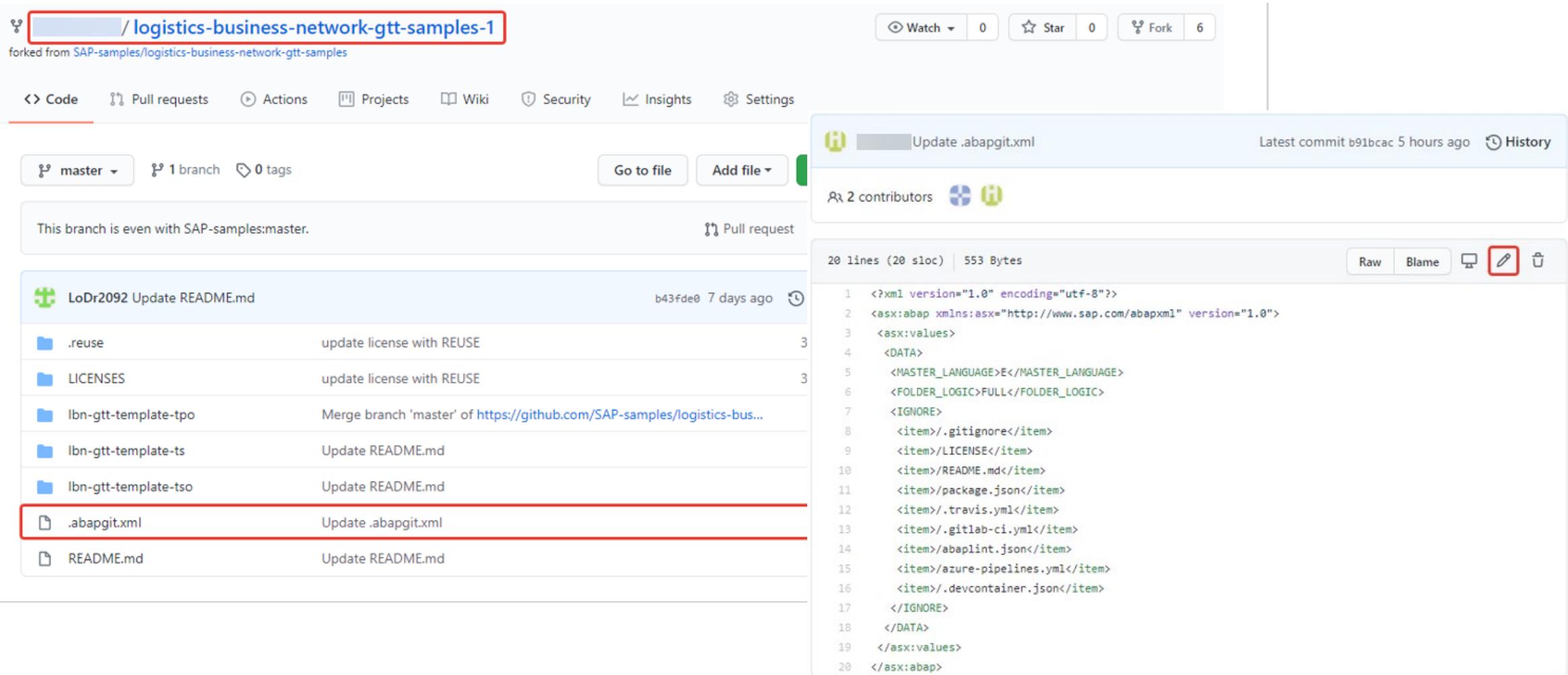
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The repository is forked from SAP-samples/logistics-business-network-gtt-samples. The main navigation bar includes Watch (0), Star (0), Fork (6), and tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a summary showing master branch, 1 branch, 0 tags, and a green 'Code' button. A message states 'This branch is even with SAP-samples:master.' with 'Pull request' and 'Compare' buttons. The repository history lists several commits, with the last commit to '.abapgit.xml' highlighted by a red box. The commit details are: LoDr2092 Update README.md, b43fde0 7 days ago, 149 commits. The commit to '.abapgit.xml' is: Update .abapgit.xml, 14 days ago. Other commits listed include: .reuse, LICENSES, lbn-gtt-template-tpo, lbn-gtt-template-ts, lbn-gtt-template-tso, README.md. To the right of the commit list is an 'About' section describing sample scenarios for SAP Logistics Business Network, global track and trace options, and a 'Readme' link. Below the commit list are sections for 'Releases' (no releases published) and 'Packages' (no packages published).

File	Commit Message	Time Ago
LoDr2092 Update README.md	b43fde0 7 days ago	149 commits
.reuse	update license with REUSE	3 months ago
LICENSES	update license with REUSE	3 months ago
lbn-gtt-template-tpo	Merge branch 'master' of https://github.com/SAP-samples/logistics-bus...	8 days ago
lbn-gtt-template-ts	Update README.md	7 days ago
lbn-gtt-template-tso	Update README.md	7 days ago
.abapgit.xml	Update .abapgit.xml	14 days ago
README.md	Update README.md	7 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 'logistics-business-network-gtt-samples-1'. The URL in the address bar is highlighted with a red box. The repository has 0 stars, 6 forks, and 6 issues. The master branch is selected, showing 1 branch and 0 tags. A pull request button is visible. The commit history lists several changes, including 'Update README.md' by LoDr2092, 'update license with REUSE' for .reuse and LICENSES, a merge from SAP-samples:master, and updates to README.md for Ibn-gtt-template-tso and Ibn-gtt-template-ts. The commit 'Update .abapgit.xml' by LoDr2092 is selected and shown in detail. This commit has 2 contributors and was made 5 hours ago. The code editor shows the XML content of the .abapgit.xml file:

```
<?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>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
    </IGNORE>
    <item>/.gitignore</item>
    <item>LICENSE</item>
    <item>README.md</item>
    <item>package.json</item>
    <item>/.travis.yml</item>
    <item>/.gitlab-ci.yml</item>
    <item>/abaplint.json</item>
    <item>/azure-pipelines.yml</item>
    <item>/.devcontainer.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-tpo/abap/zsrc/</STARTING_FOLDER>" as follows.

3-4: Commit changes.

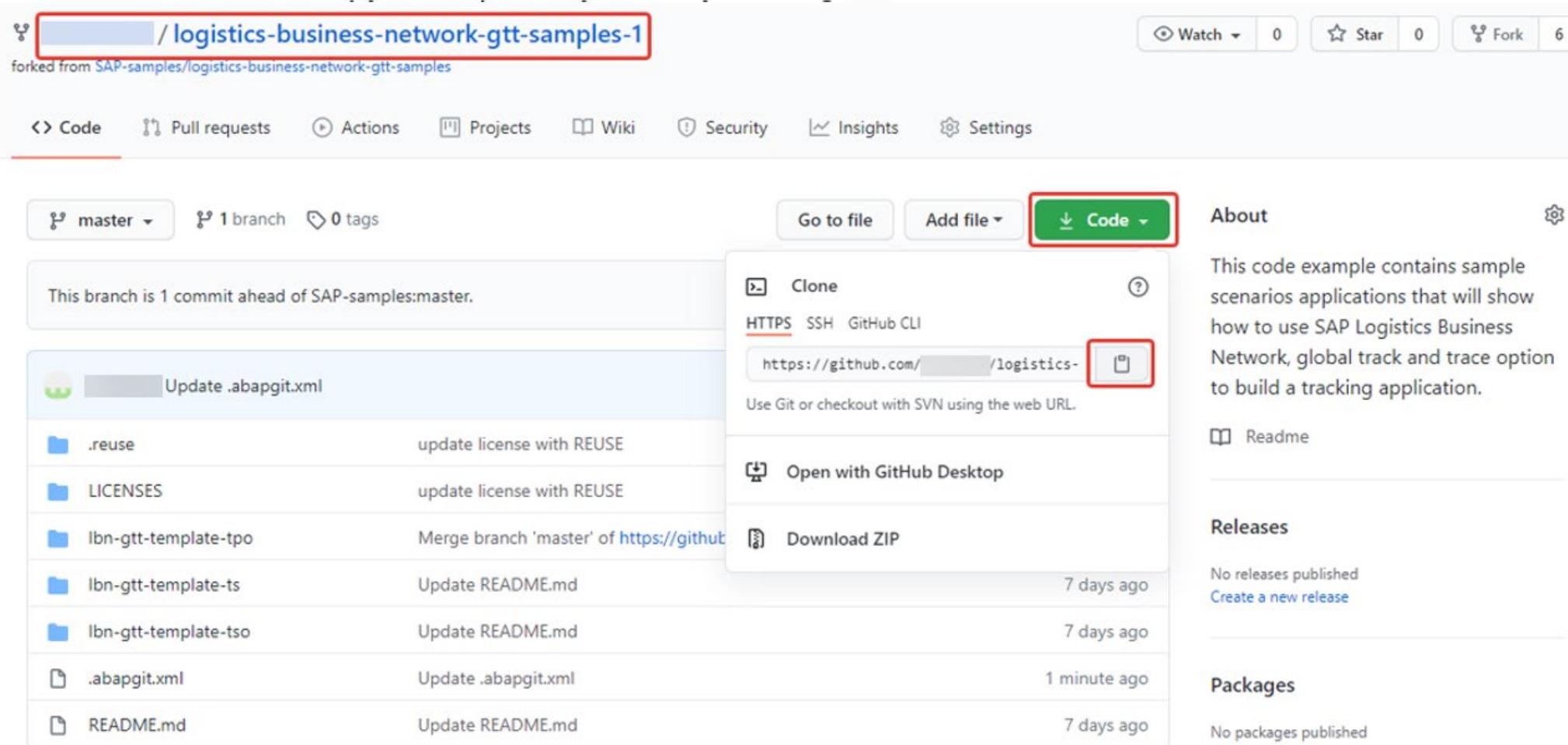
The screenshot shows a GitHub commit dialog for the file `.abapgit.xml` in the `logistics-business-network-gtt-samples-1` repository. The code editor on the left displays the XML configuration file. A specific line has been highlighted with a red box:

```
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-tpo/abap/zsrc/</STARTING_FOLDER>
7       <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8     <IGNORE>
9       <item>/.gitignore</item>
10      <item>/LICENSE</item>
11      <item>/README.md</item>
12      <item>/package.json</item>
13      <item>/.travis.yml</item>
14      <item>/.gitlab-ci.yml</item>
15      <item>/abaplint.json</item>
16      <item>/azure-pipelines.yml</item>
17      <item>/devcontainer.json</item>
18    </IGNORE>
19  </DATA>
20 </asx:values>
21 </asx:abap>
```

The line `<STARTING_FOLDER>lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>` is highlighted with a red box. The commit message on the right is set to "Update .abapgit.xml". The "Commit changes" button is highlighted with a red border.

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 'logistics-business-network-gtt-samples-1'. The repository was forked from SAP-samples/logistics-business-network-gtt-samples. The master branch is 1 commit ahead of SAP-samples:master. A recent update to '.abapgit.xml' is listed. On the right, a context menu is open over the 'Code' button, specifically the 'Clone' section. The URL 'https://github.com/.../logistics...' is highlighted with a red box, and the copy icon next to it is also highlighted with a red box. The menu includes options for 'Clone' (with HTTPS, SSH, and GitHub CLI), 'Open with GitHub Desktop', and 'Download ZIP'. The repository has 0 stars, 0 forks, and 6 issues.

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

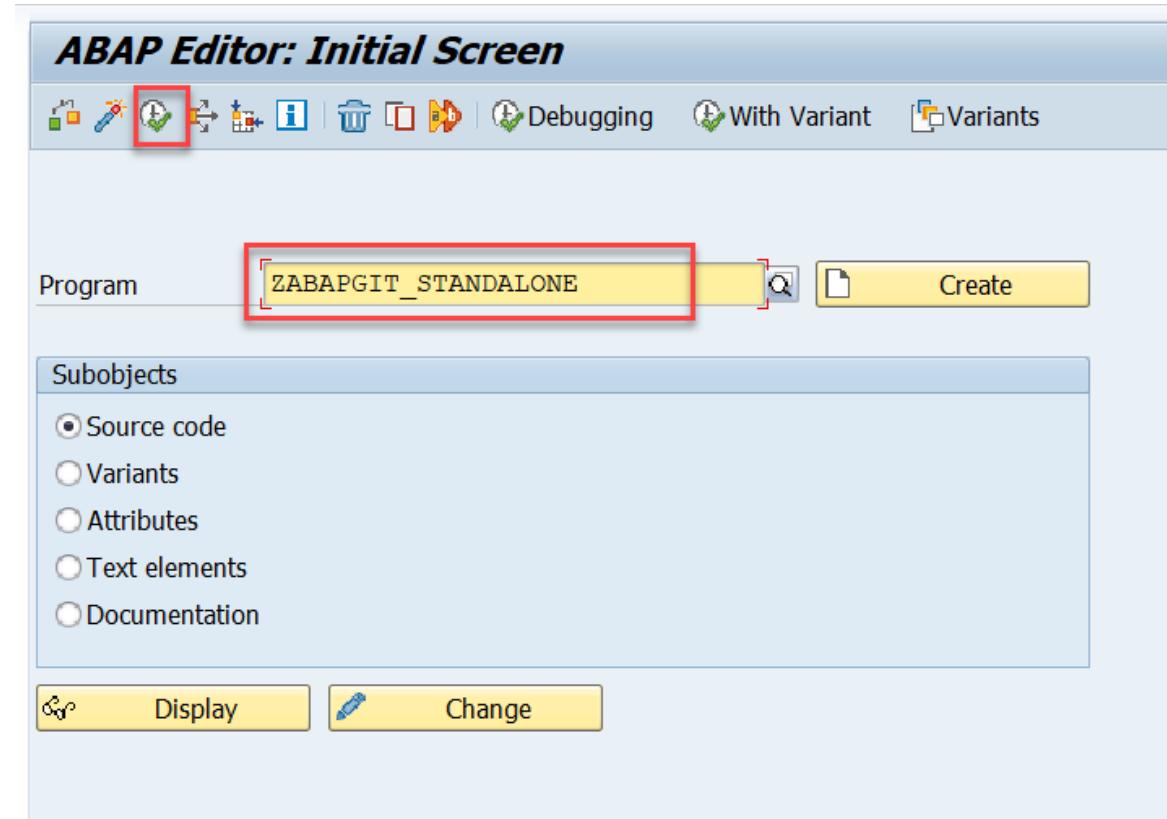
Packages

No packages published

STEP 4: Download ABAP Code from GitHub

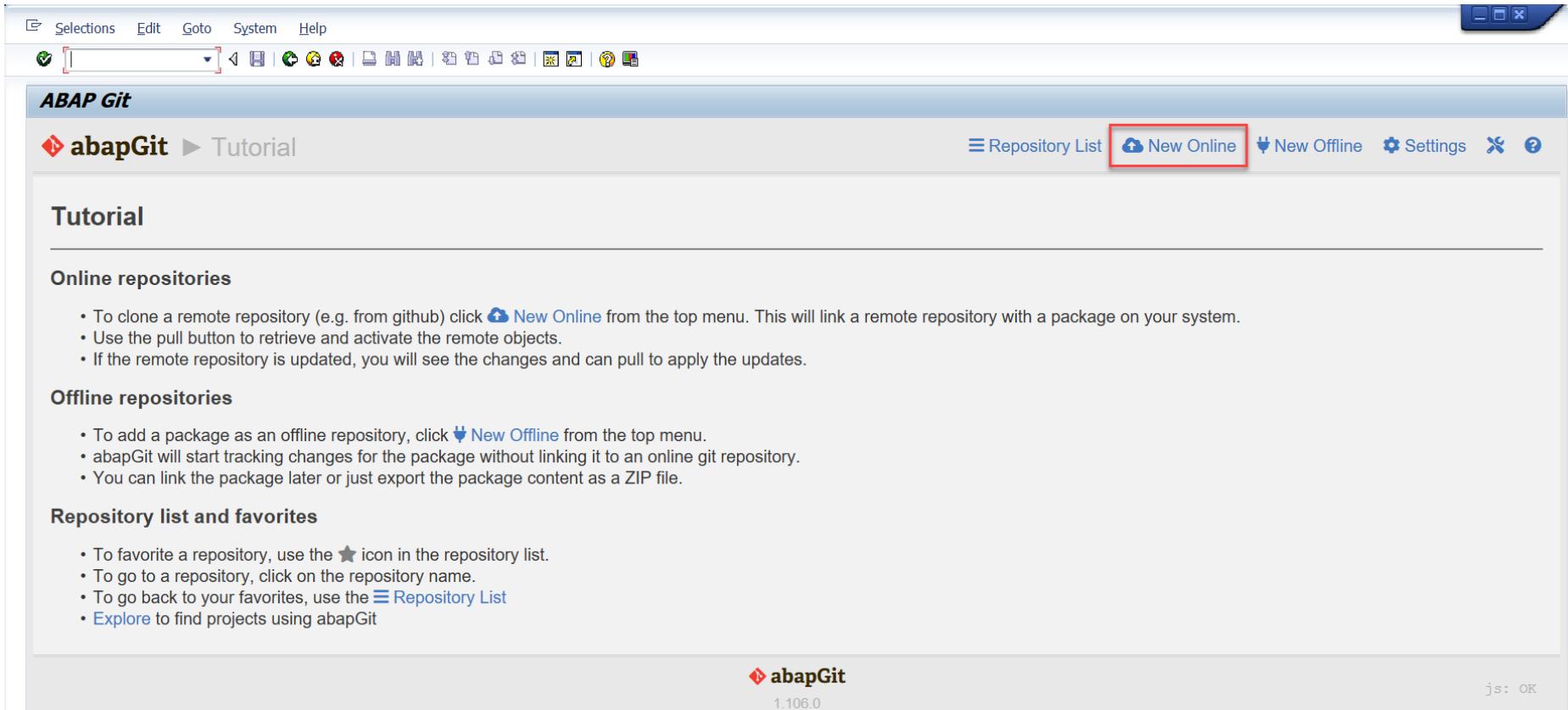
4-1: Enter T-code **SE38** and fill the report name from STEP 1, **ZABAPGIT_STANDALONE**

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**:

`https://github.com/xxxxxx/logistics-business-network-gtt-samples-1.git`

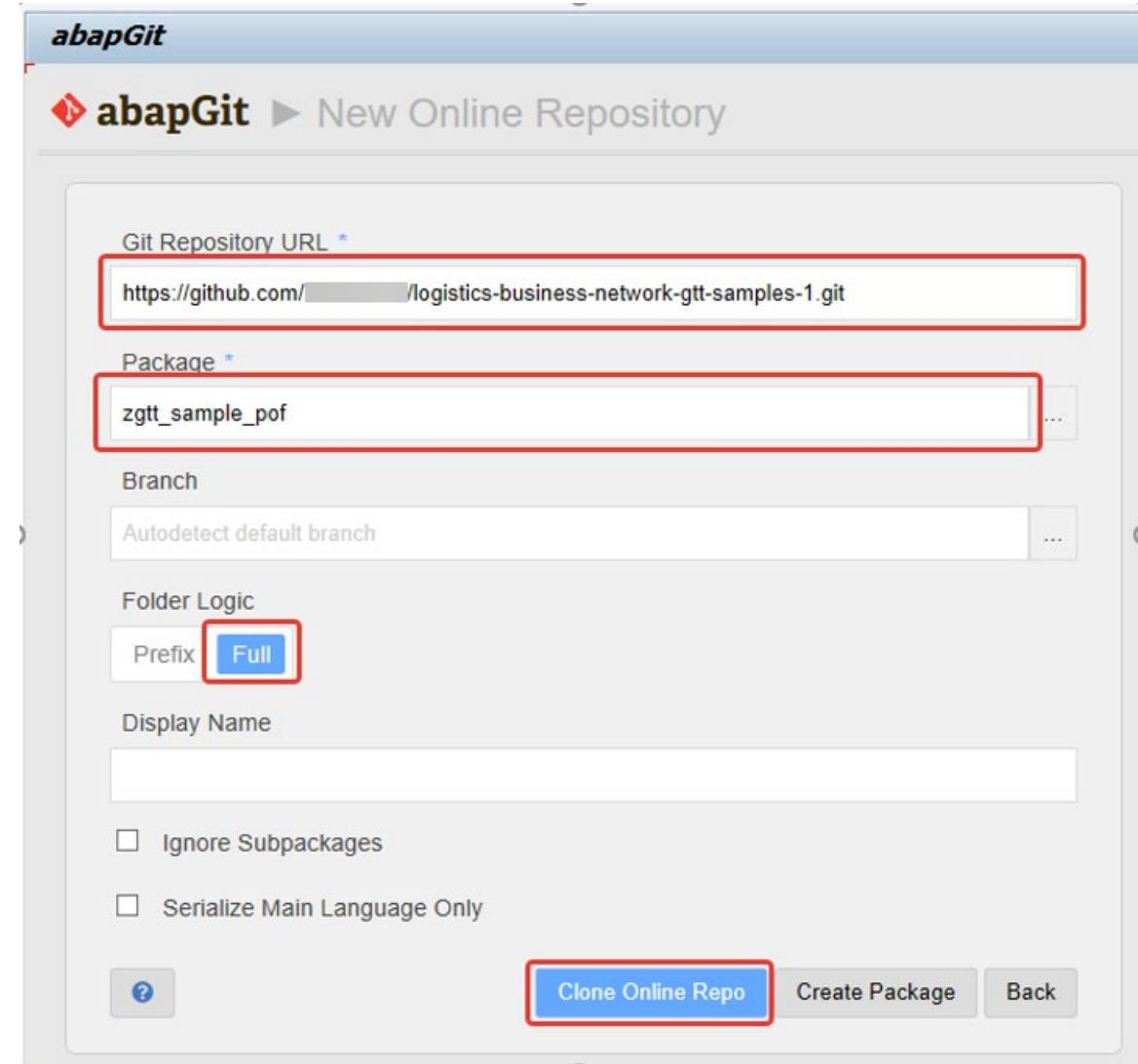
Caution:

This URL is the user's account 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.

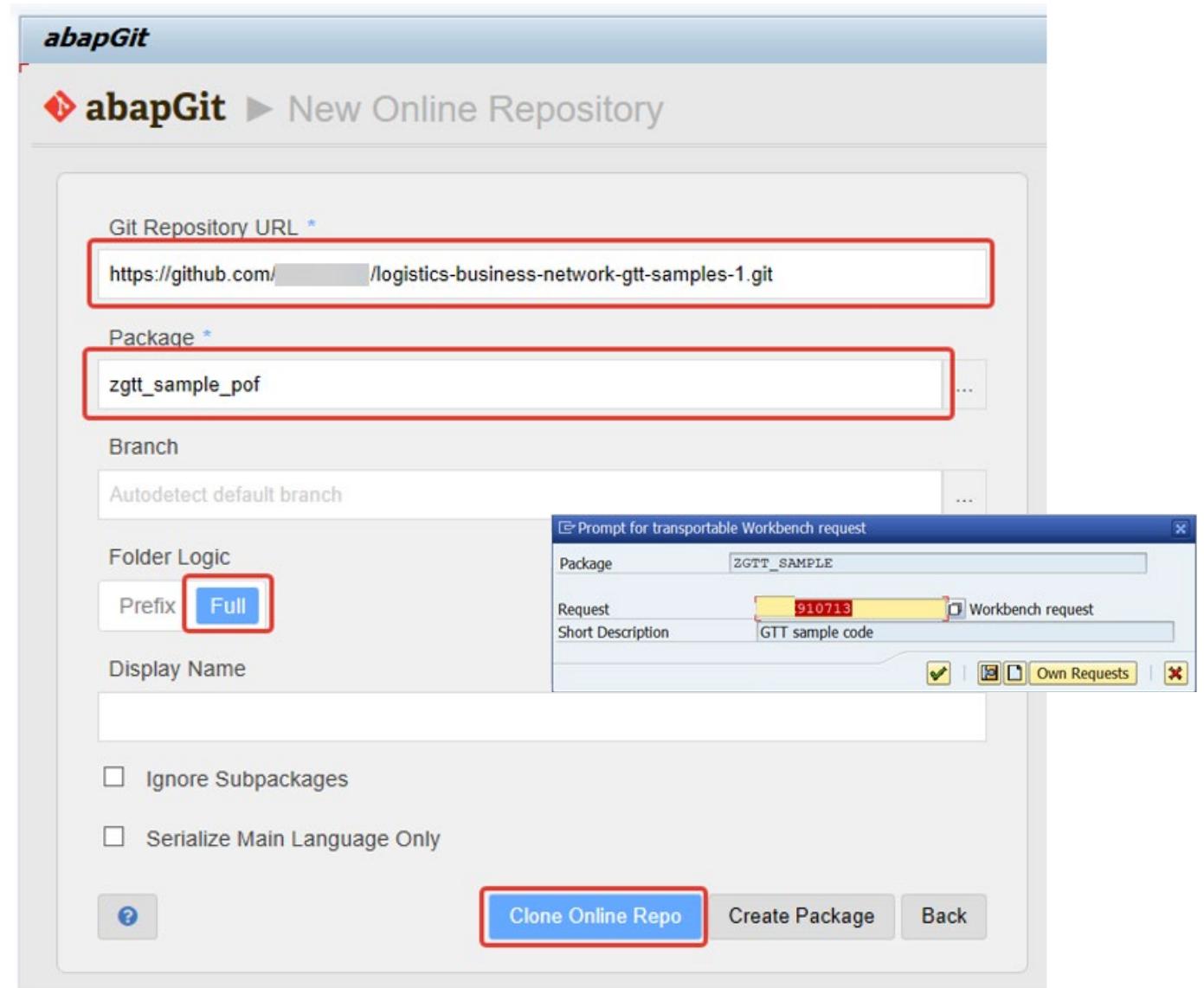
4-6: Set *Full* for **Folder Logic**

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

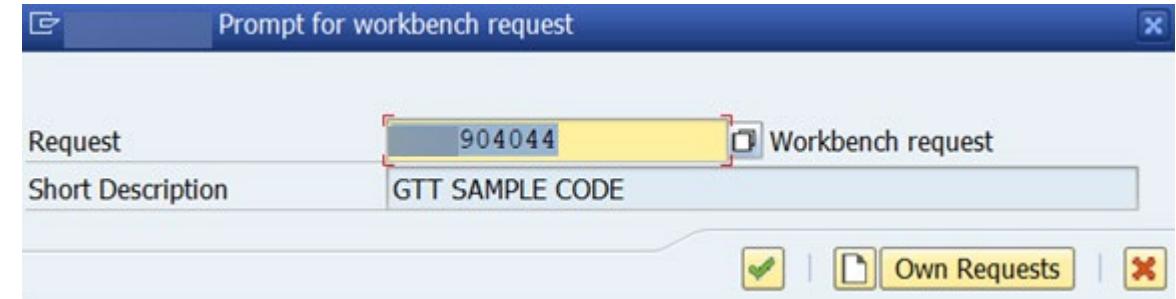
4-9: Click **Pull** to pull down the latest version of sample code.

The screenshot shows the abapGit interface. At the top, there's a navigation bar with 'abapGit' logo, 'Repository' link, 'Repository List' button, 'Settings' button, and a help icon. Below the navigation is a header for the repository 'logistics-business-network-gtt-samples-1' with its URL, commit hash 'baaf604', and branches 'master' and 'zgtt_sample_pof'. A toolbar below the header has buttons for 'Pull' (which is highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a settings gear icon. The main area is a table with columns 'Type', 'Name', and 'Path'. The 'Type' column includes icons for 'non-code and meta files', 'CLAS' (class), and 'DEVC' (development). The 'Name' column lists file names like '.abapgit.xml', 'ZCL_IM_POF_GTT_LE_SHIPMENT', and 'ZGTT_SAMPLE_POF'. The 'Path' column shows the full path for each file. To the right of the table, there are 'diff' buttons with colored status indicators (yellow, green, red) and 'A' or 'M' labels.

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

STEP 4: Download ABAP Code from GitHub

4-10: Assign the change to a change request. If you do not have any available change requests, you need to create a new one.



STEP 4: Download ABAP Code from GitHub

4-11: After you download the code, you can check them with T-code SE80.

Object Navigator

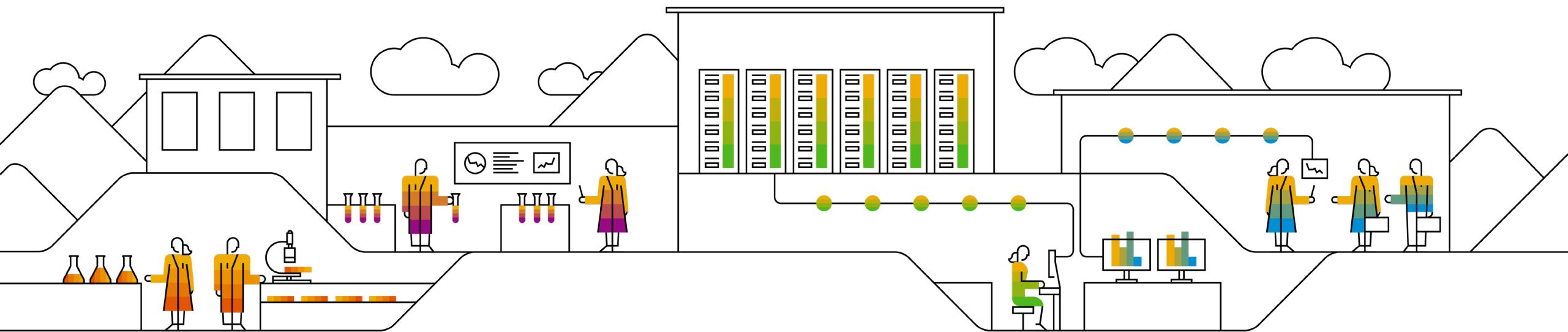
Repository Browser

Package: ZGTT_SAMPLE_POF

Object Name	Description
ZGTT_SAMPLE_POF	ABAP sample package for global track and trace of LBN
Dictionary Objects	
Database Tables	ZPOF_GTT_EE_REL
Data Elements	ZPOF_KOSTA ZPOF_PDSTK ZPOF_PKSTA ZPOF_WBSTA
Class Library	ZCL_IM_POF_GTT_LE_SHIPMENT
Functions	ZPOF_GTT
Message Classes	ZPOF_GTT
Enhancements	ZPOF_GTT_LE_SHIPMENT
Implementation Class for BAdI Implementation ZPOF_GTT_L...	
POF GTT	
Purchasing Order Fulfillment Messages	
Shipment Cross TP Update	

C) Download ABAP Code from GitHub

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

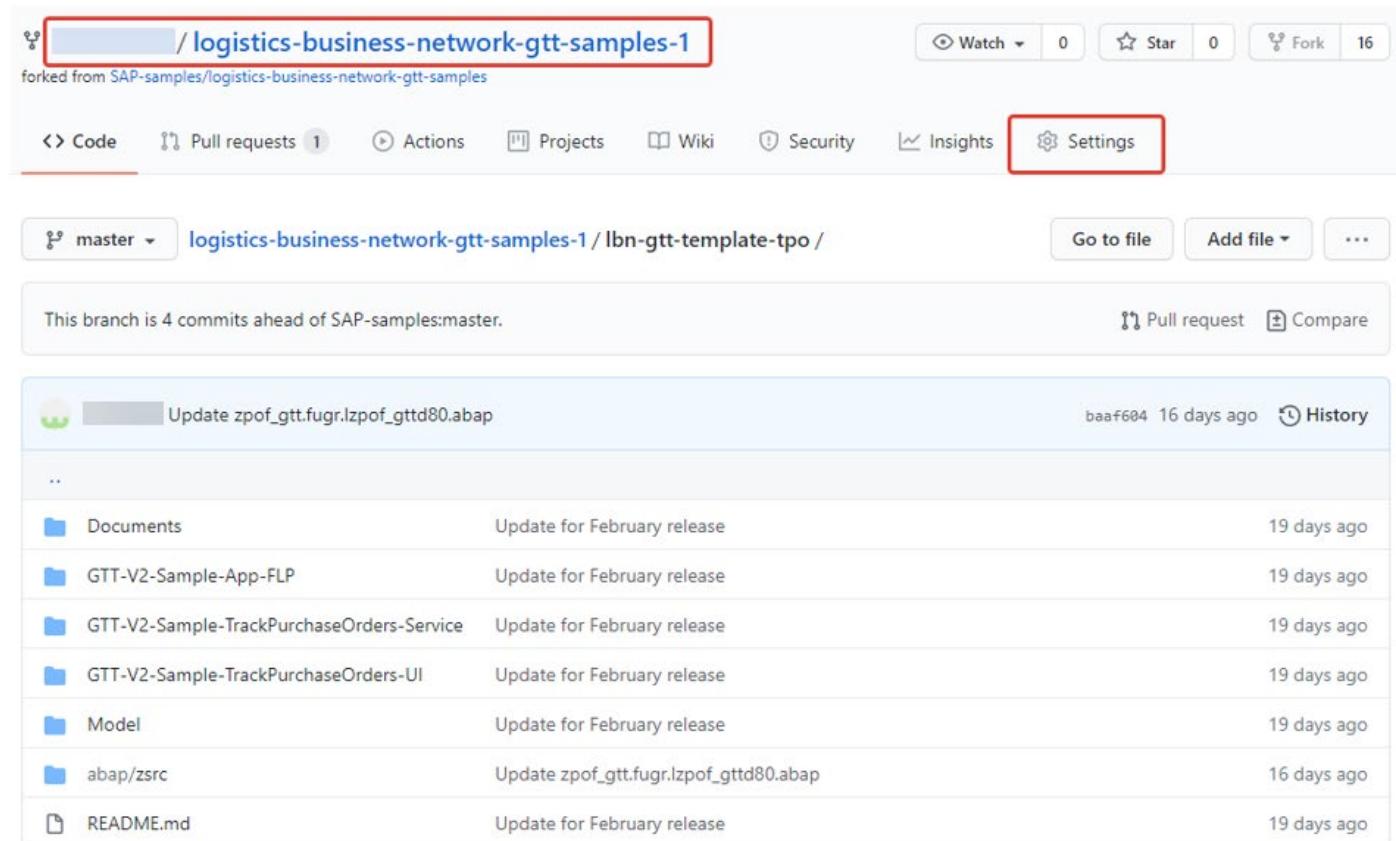


STEP 1: Delete the User's Account Repository

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

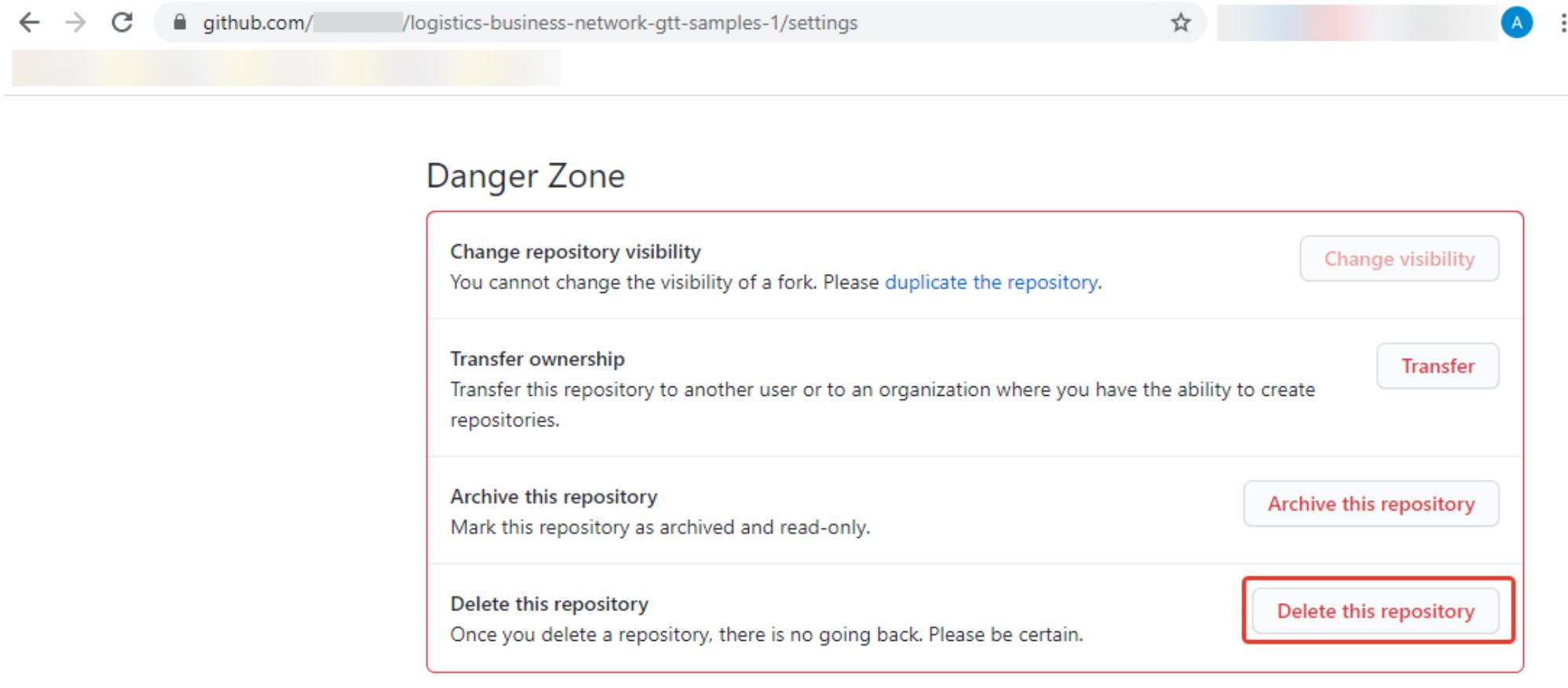
In the latest release, there is some code changes in public sample code, you need to update the local code according to the latest public sample code.

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



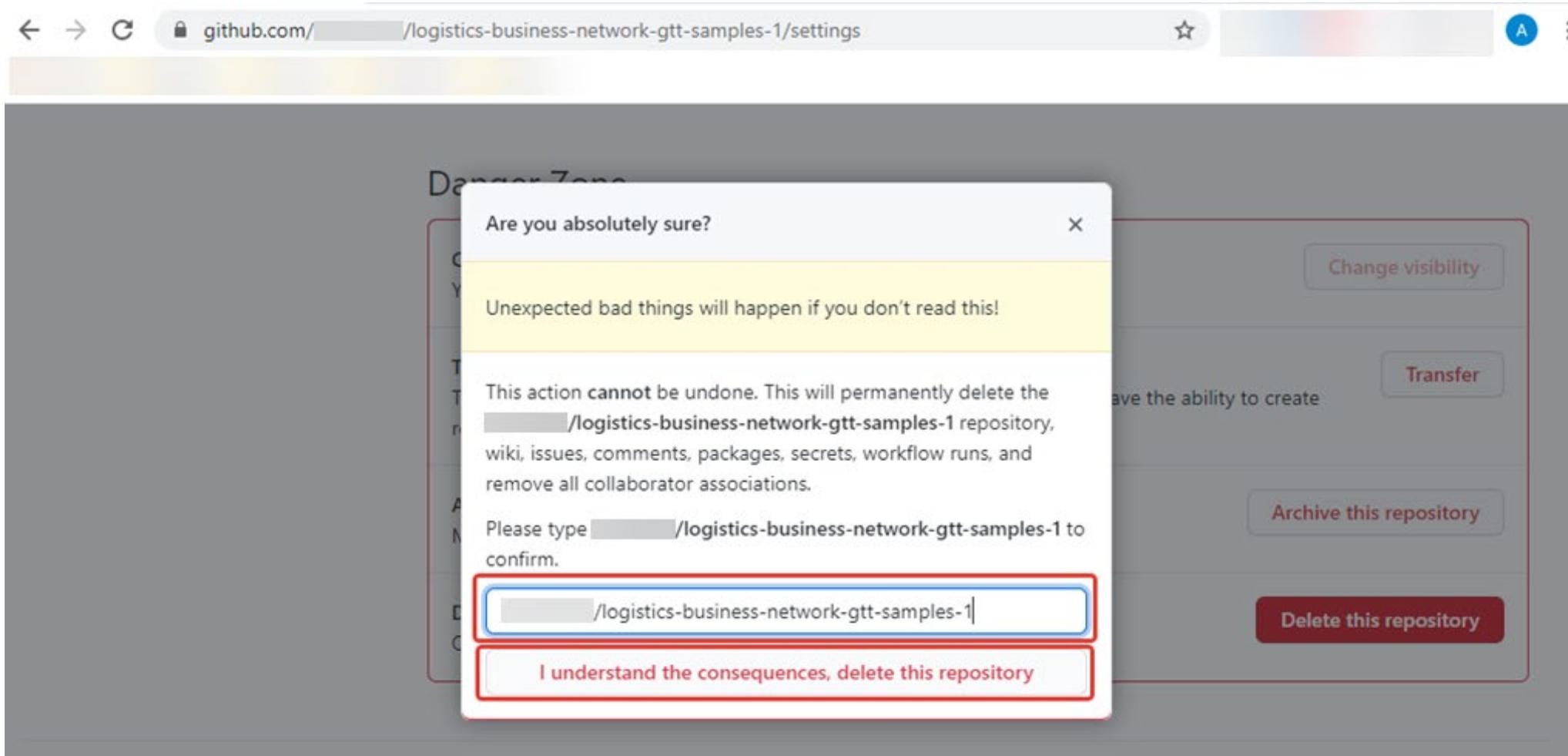
STEP 1: Delete the User's Account Repository

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



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

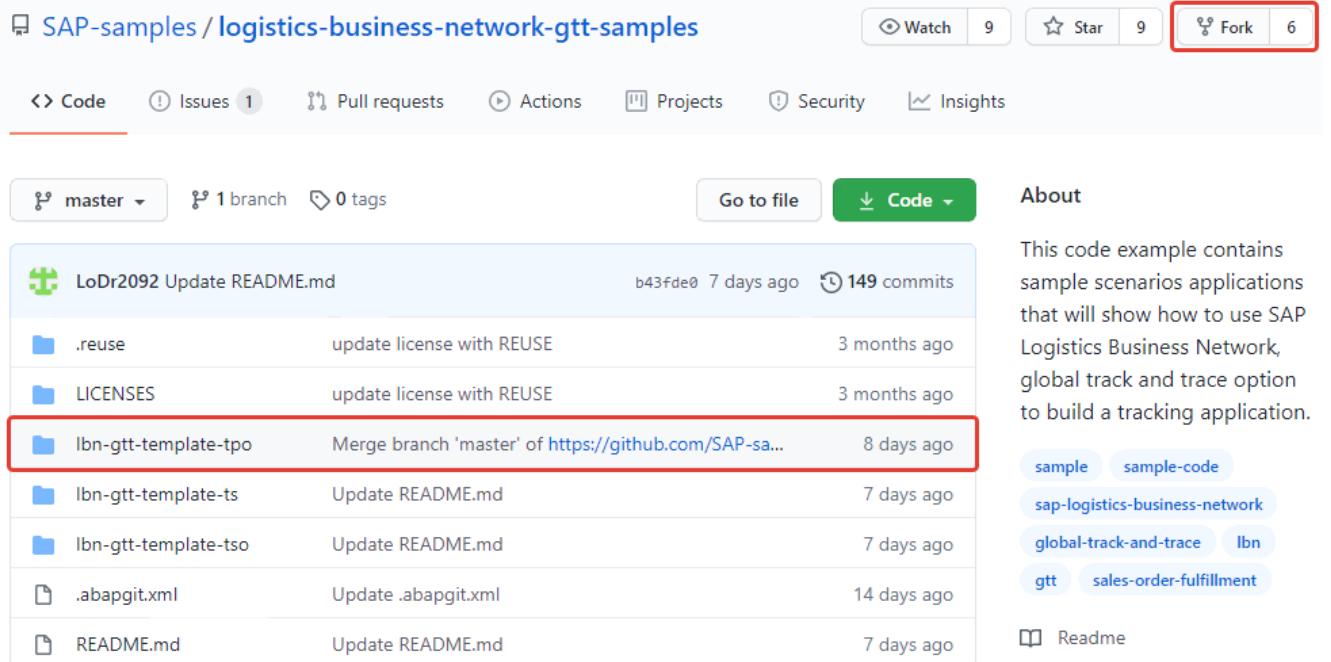
1-5: The user's account repository is deleted.

The screenshot shows a web browser window with the URL `github.com` in the address bar. The GitHub homepage is visible, featuring the navigation bar with `Pull requests`, `Issues`, `Marketplace`, and `Explore`. A prominent message in the center of the page reads: "Your repository \"[REDACTED]/logistics-business-network-gtt-samples-1\" was successfully deleted." Below this message, there is a sidebar with options like "Create your first project", "Create repository", and "Import repository". A large modal window titled "Introduce yourself" is open, providing instructions on how to create a README file. It includes a sample README.md content with five items, each preceded by an emoji icon. At the bottom of the modal are "Dismiss this" and "Continue" buttons.

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 code into the user's account and meanwhile it will navigate to user's own repository.



The screenshot shows a GitHub repository page for 'SAP-samples / logistics-business-network-gtt-samples'. The top navigation bar includes 'Watch' (9), 'Star' (9), 'Fork' (6), and other standard GitHub icons. Below the header, there are tabs for 'Code', 'Issues' (1), 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. A dropdown menu shows 'master' selected, along with '1 branch' and '0 tags'. Buttons for 'Go to file' and 'Code' are present. The main content area displays a list of commits. One commit, 'Ibn-gtt-template-tpo' (Merge branch 'master' of https://github.com/SAP-sa...), is highlighted with a red border. To the right of the commit list is a descriptive text block: '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 this are several blue circular tags with white text: 'sample', 'sample-code', 'sap-logistics-business-network', 'global-track-and-trace', 'Ibn', 'gtt', and 'sales-order-fulfillment'. At the bottom right is a 'Readme' link.

Commit	Message	Date
LoDr2092 Update README.md	b43fde0 7 days ago	149 commits
.reuse	update license with REUSE	3 months ago
LICENSES	update license with REUSE	3 months ago
Ibn-gtt-template-tpo	Merge branch 'master' of https://github.com/SAP-sa...	8 days ago
Ibn-gtt-template-ts	Update README.md	7 days ago
Ibn-gtt-template-tso	Update README.md	7 days ago
.abapgit.xml	Update .abapgit.xml	14 days ago
README.md	Update README.md	7 days ago

STEP 2: Fork Sample Code Repository

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

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The repository was forked from SAP-samples/logistics-business-network-gtt-samples. The main navigation bar includes Watch (0), Star (0), Fork (16), and tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the tabs, it shows master branch (1 branch, 0 tags), Go to file, Add file, and a green Code button. A message states 'This branch is even with SAP-samples:master.' with Pull request and Compare buttons. The commit history lists several updates by LoDr2092, including updates to README.md, .reuse, LICENSES, folder names, and abapgit.xml, along with a FAQ update and a README update. To the right, there are sections for About, Readme, Releases, and Packages, all currently empty.

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

File / Commit	Description	Date
LoDr2092 Update README.md	update license with REUSE	17 days ago
.reuse	update license with REUSE	5 months ago
LICENSES	update license with REUSE	5 months ago
Ibn-gtt-template-tpo	change folder name from ABAP to abap	17 days ago
Ibn-gtt-template-ts	Update README.md	17 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	17 days ago
.abapgit.xml	Update .abapgit.xml	19 days ago
FAQs_for_Template_Code_Im...	Update for February release	19 days ago
README.md	Update README.md	17 days ago

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

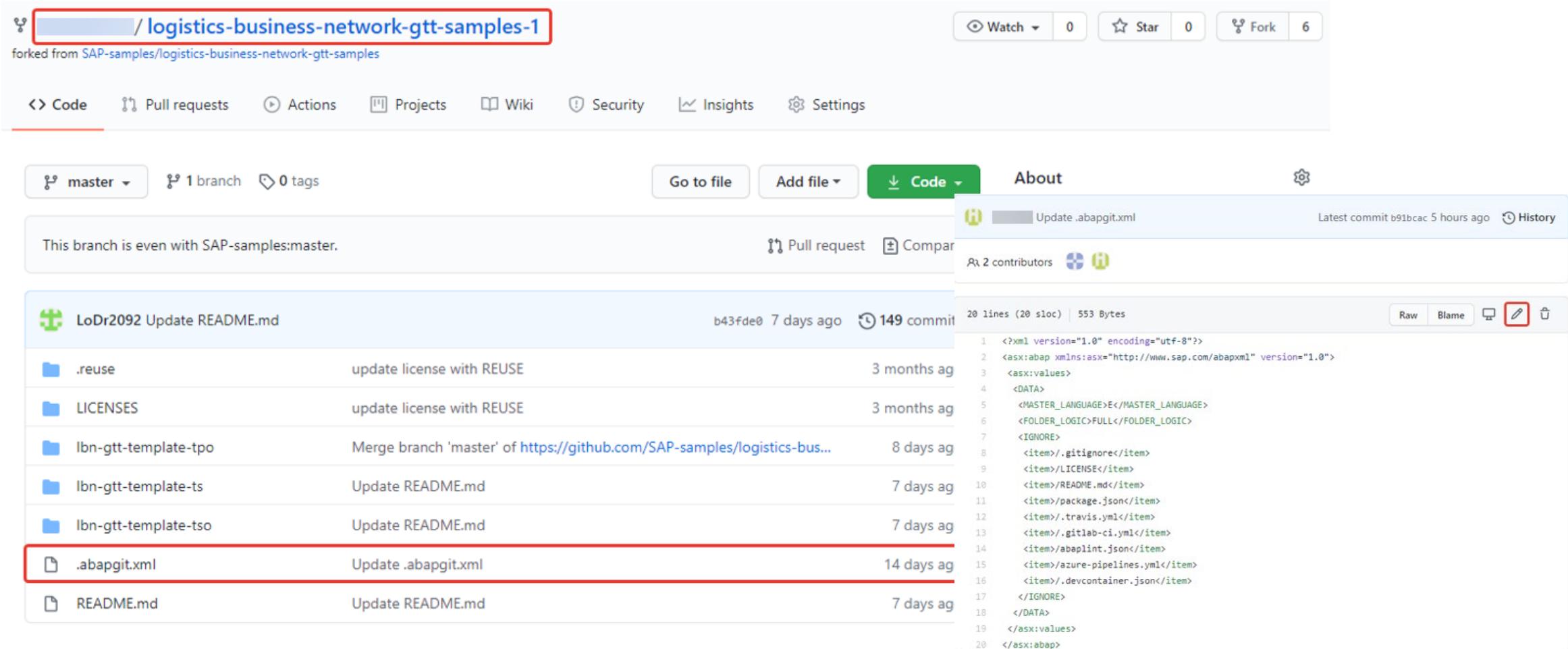
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The repository is a fork from SAP-samples/logistics-business-network-gtt-samples. The 'Code' tab is selected. The 'master' branch is current, with 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including a merge from SAP-samples and updates to README files. The commit 'Update .abapgit.xml' is highlighted with a red box. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages', all of which are currently empty.

Commit	Message	Date
LoDr2092 Update README.md	b43fde0 7 days ago	149 commits
.reuse	update license with REUSE	3 months ago
LICENSES	update license with REUSE	3 months ago
Ibn-gtt-template-tpo	Merge branch 'master' of https://github.com/SAP-samples/logistics-bus...	8 days ago
Ibn-gtt-template-ts	Update README.md	7 days ago
Ibn-gtt-template-tso	Update README.md	7 days ago
.abapgit.xml	Update .abapgit.xml	14 days ago
README.md	Update README.md	7 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 'logistics-business-network-gtt-samples-1'. The repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' tab is selected. The commit history lists several changes, with the last commit, 'Update .abapgit.xml' by 'LoDr2092' (b43fde0), highlighted with a red border. This commit updated the '.abapgit.xml' file 14 days ago. The file content is displayed on the right:

```
<?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>
<FOLDER_LOGIC>FULL</FOLDER_LOGIC>
<IGNORE>
<item>/.gitignore</item>
<item>/LICENSE</item>
<item>/README.md</item>
<item>/package.json</item>
<item>/.travis.yml</item>
<item>/gitlab-ci.yml</item>
<item>/abaplint.json</item>
<item>/azure-pipelines.yml</item>
<item>/devcontainer.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-tpo/abap/zsrc/</STARTING_FOLDER>" as follows.

3-4: Commit changes.

The screenshot shows a GitHub commit dialog for the file '.abapgit.xml' in the repository 'logistics-business-network-gtt-samples-1'. The code editor on the left displays the XML configuration file. Line 6 contains the path '<STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>'. This line is highlighted with a red rectangle. The commit message field on the right contains the text 'Update .abapgit.xml'. Below the message field are two radio button options: one selected for committing directly to the 'master' branch, and another for creating a new branch and starting a pull request. A large green 'Commit changes' button is at the bottom of the dialog, which is also highlighted with a red rectangle.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <DATA>
      <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
      <STARTING_FOLDER>/lbn-gtt-template-tpo/abap/zsrc/</STARTING_FOLDER>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
    <IGNORE>
      <item>/.gitignore</item>
      <item>/LICENSE</item>
      <item>/README.md</item>
      <item>/package.json</item>
      <item>/.travis.yml</item>
      <item>/.gitlab-ci.yml</item>
      <item>/abaplint.json</item>
      <item>/azure-pipelines.yml</item>
      <item>/devcontainer.json</item>
    </IGNORE>
  </DATA>
</asx:values>
</asx:abap>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

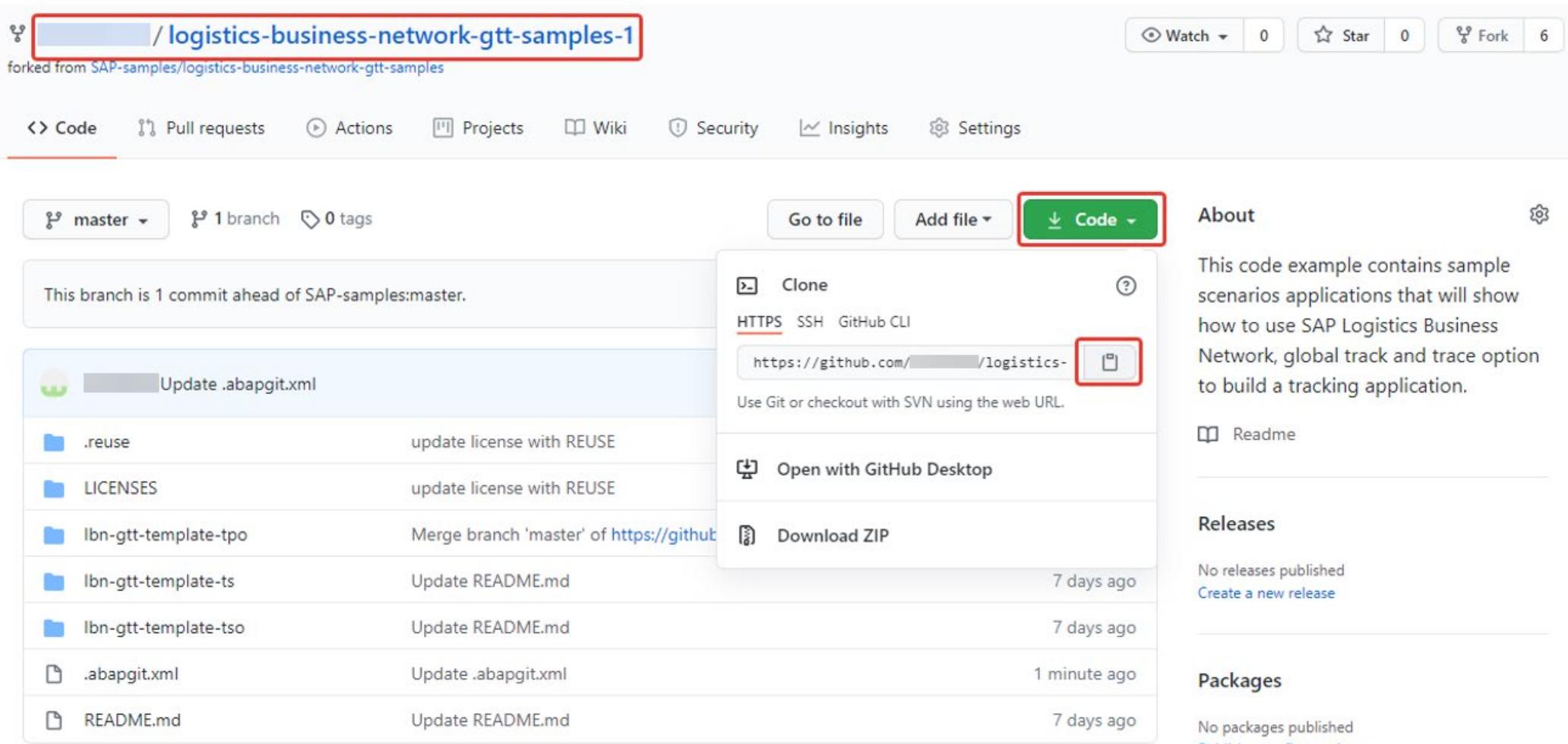
-o- Commit directly to the master branch.

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

Commit changes Cancel

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

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

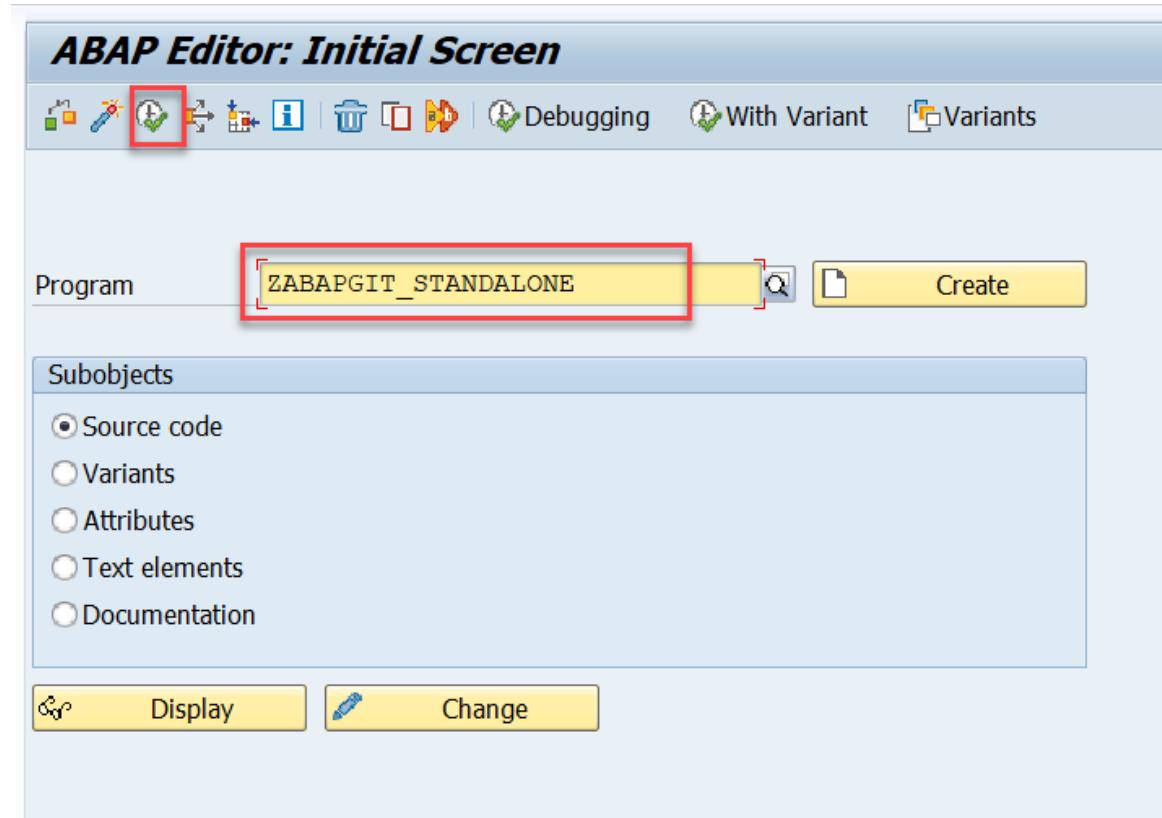


The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The URL in the address bar is highlighted with a red box. The repository has 0 stars and 6 forks. The main navigation bar includes Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, it shows 1 branch and 0 tags. A message indicates the 'master' branch is 1 commit ahead of SAP-samples:master. On the left, a list of files shows an update to '.abapgit.xml'. On the right, a 'Code' dropdown menu is open, showing options for Clone (with a red box around the copy icon), SSH, GitHub CLI, and a link to the repository's web URL ([https://github.com/\[REDACTED\]/logistics-](https://github.com/[REDACTED]/logistics-)). The 'Clone' option is highlighted with a red box. Other menu items include Open with GitHub Desktop and Download ZIP. To the right of the menu, there is an 'About' section describing the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and a Readme link. There are also sections for Releases, Packages, and a note about publishing packages.

STEP 4: Update ABAP Code from GitHub

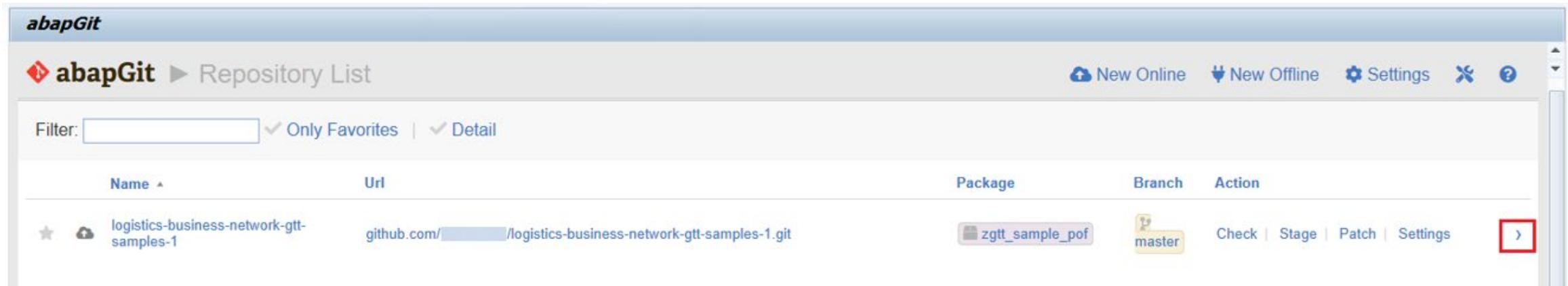
4-1: Enter T-code *SE38* and fill in the report name *ZABAPGIT_STANDALONE*.

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



STEP 4: Update ABAP Code from GitHub

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

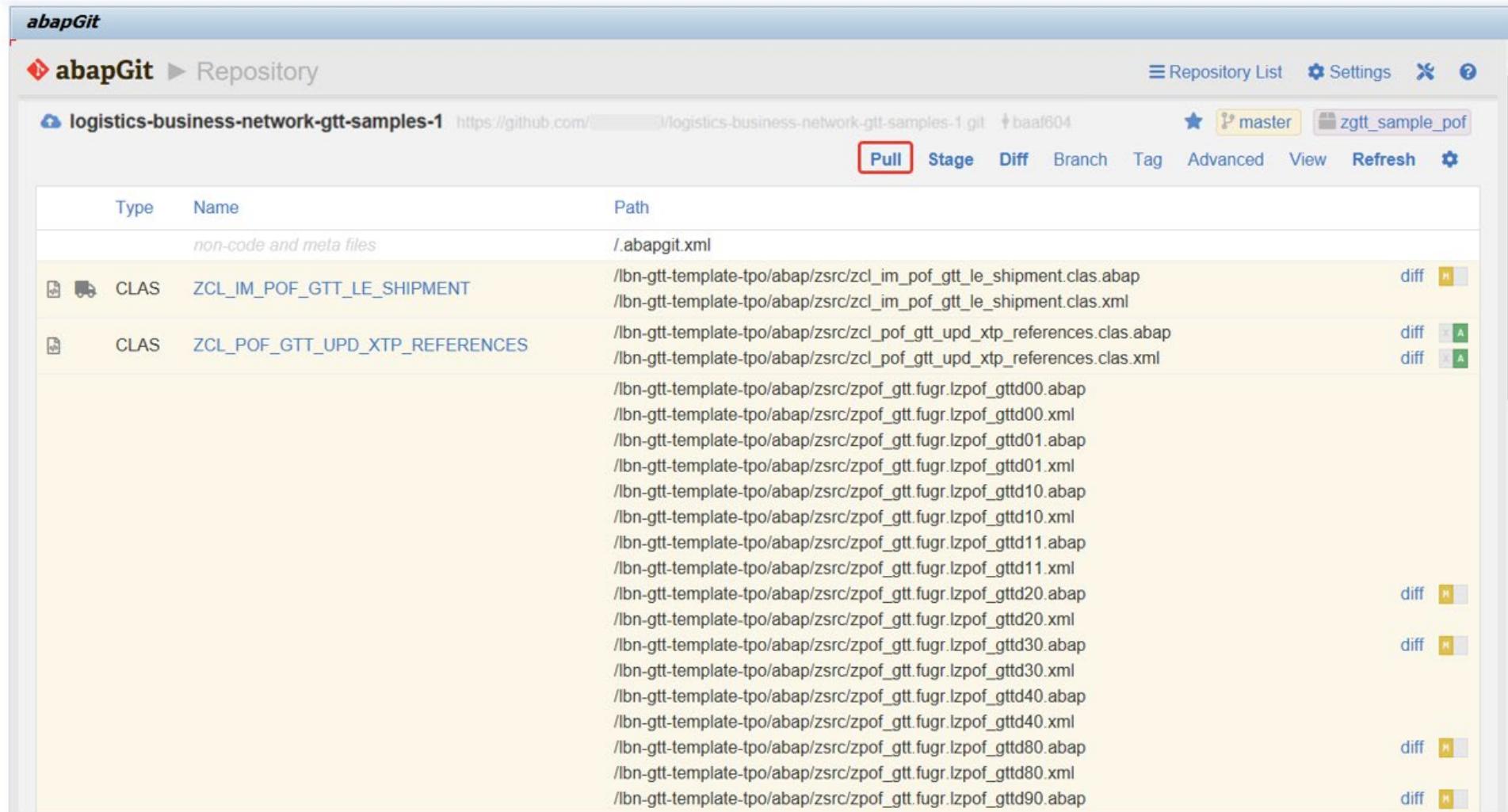


The screenshot shows the abapGit application interface. The title bar says "abapGit". The main area is titled "Repository List". There is a "New Online" button, a "New Offline" button, a "Settings" button, and a "Help" button. Below these are "Only Favorites" and "Detail" filter options. A "Filter:" input field is present. The main table has columns: Name, Url, Package, Branch, and Action. One row is shown: "logistics-business-network-gtt-samples-1" with Url "github.com/[REDACTED]/logistics-business-network-gtt-samples-1.git", Package "zgtt_sample_pof", Branch "master", and Action buttons "Check", "Stage", "Patch", and "Settings". The "Check" button is highlighted with a red box.

Name	Url	Package	Branch	Action
logistics-business-network-gtt-samples-1	github.com/[REDACTED]/logistics-business-network-gtt-samples-1.git	zgtt_sample_pof	master	Check Stage Patch Settings

STEP 4: Update ABAP Code from GitHub

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

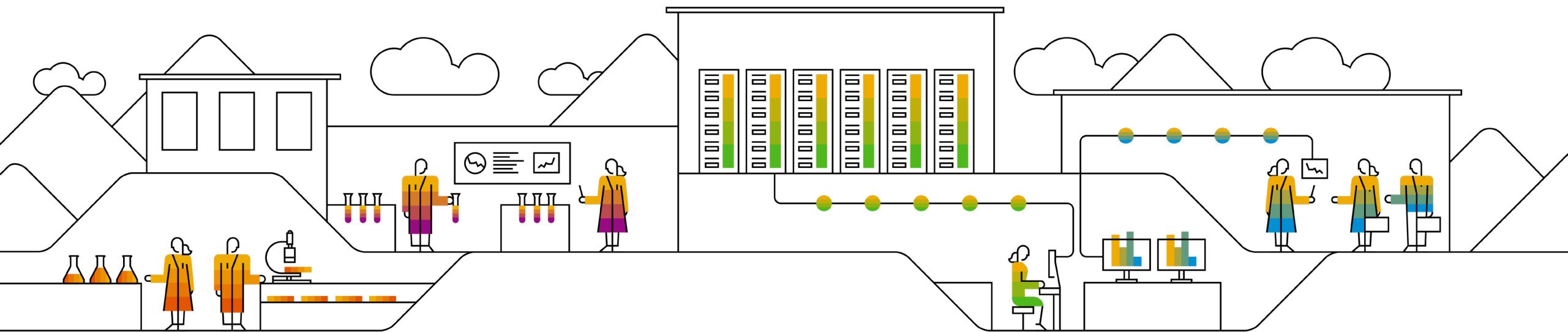


The screenshot shows the abapGit interface for managing ABAP code from GitHub. The repository selected is 'logistics-business-network-gtt-samples-1'. The 'Pull' button in the toolbar is highlighted with a red box. The main area displays a table of files categorized by type (Type) and name, along with their corresponding paths and diff status.

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

C) Download ABAP Code from GitHub

C3. Download Another ABAP Code from GitHub (TSOF)



STEP 1: Fork Sample Code Repository

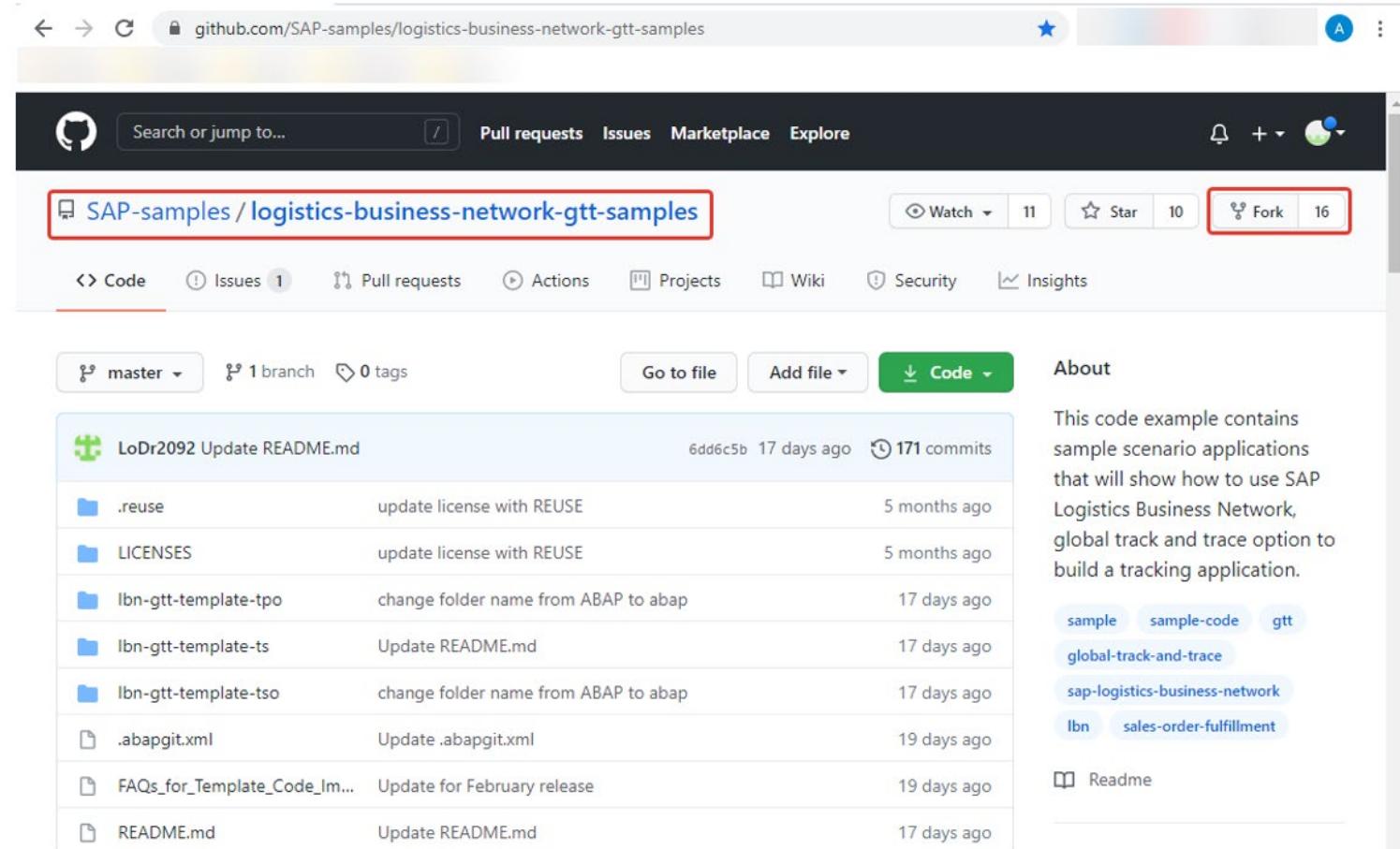
Prerequisite:

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

To install the TSOF 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 code into the user's account and meanwhile it will navigate to user's own repository.



The screenshot shows a GitHub repository page for the project "SAP-samples / logistics-business-network-gtt-samples". The URL in the address bar is "github.com/SAP-samples/logistics-business-network-gtt-samples". The repository has 11 watchers, 10 stars, and 16 forks. The "Fork" button in the top right is highlighted with a red box. The main content area displays a list of commits from a user named "LoDr2092". The commits are as follows:

Commit	Message	Date
LoDr2092 Update README.md	update license with REUSE	5 months 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	17 days ago
Ibn-gtt-template-ts	Update README.md	17 days ago
Ibn-gtt-template-tso	change folder name from ABAP to abap	17 days ago
.abapgit.xml	Update .abapgit.xml	19 days ago
FAQs_for_Template_Code_Im...	Update for February release	19 days ago
README.md	Update README.md	17 days ago

On the right side, there is an "About" section with a detailed description of the code example, a tag cloud, and a "Readme" link.

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

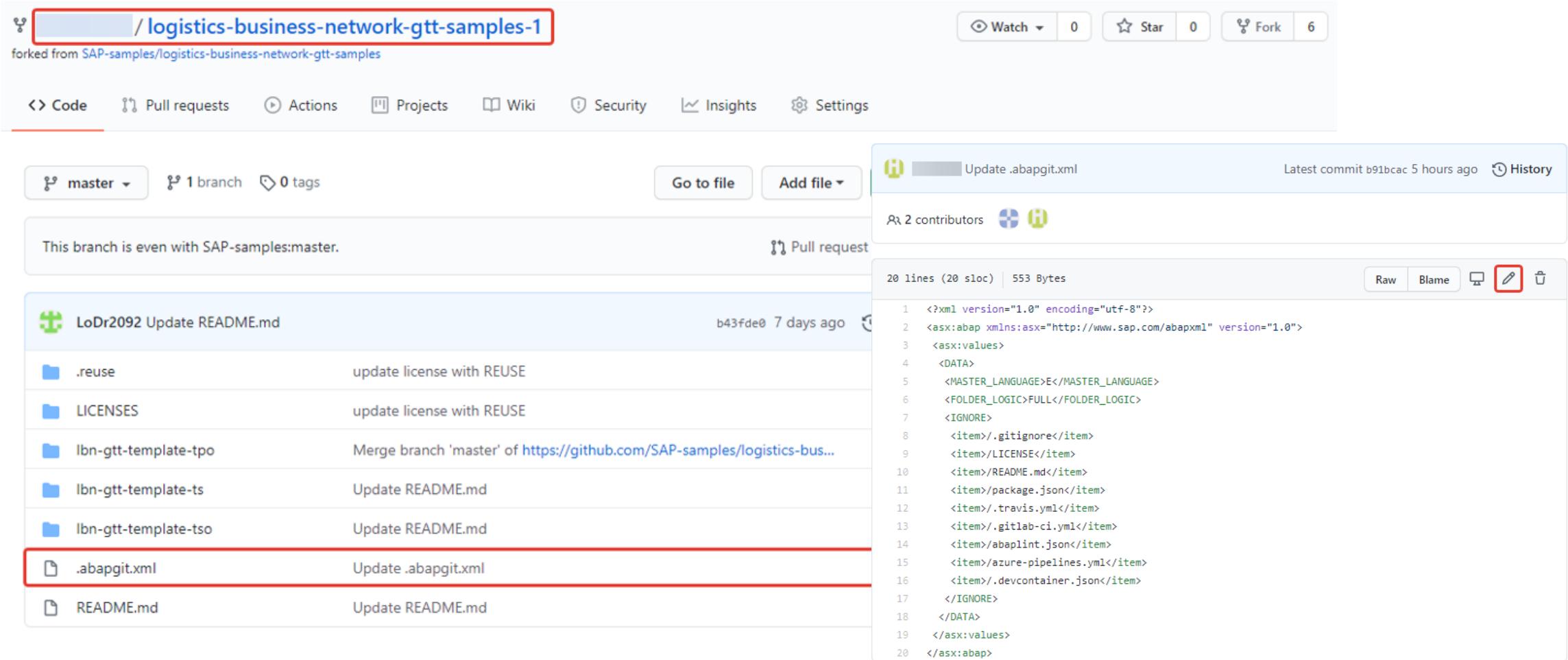
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The repository is forked from SAP-samples/logistics-business-network-gtt-samples. The 'Code' tab is selected. The repository has 1 branch and 0 tags. A message indicates the branch is even with SAP-samples:master. The commit history lists several changes, including updates to README.md, LICENSES, and template files. The '.abapgit.xml' file is highlighted with a red border. The repository has 0 stars, 6 forks, and 0 issues. The 'About' section describes the code example as containing sample scenarios applications for SAP Logistics Business Network, global track and trace options, and building a tracking application. It includes links to Readme, Releases, and Packages.

File / Action	Description	Date
.reuse	update license with REUSE	3 months ago
LICENSES	update license with REUSE	3 months ago
Ibn-gtt-template-tpo	Merge branch 'master' of https://github.com/SAP-samples/logistics-bus...	8 days ago
Ibn-gtt-template-ts	Update README.md	7 days ago
Ibn-gtt-template-tso	Update README.md	7 days ago
.abapgit.xml	Update .abapgit.xml	14 days ago
README.md	Update README.md	7 days ago

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

2-2: Click  button to edit the file.



The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The URL in the address bar is highlighted with a red box. The repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The main navigation bar includes 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, it shows 'master' branch, '1 branch', and '0 tags'. A message says 'This branch is even with SAP-samples:master.' There is a 'Pull request' button. The repository has 2 contributors. The latest commit was made 5 hours ago by user 'b91bcac'. The commit message is 'Update .abapgit.xml'. The code editor shows the 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  <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
7  <IGNORE>
8  <item>/.gitignore</item>
9  <item>/LICENSE</item>
10 <item>/README.md</item>
11 <item>/package.json</item>
12 <item>/travis.yml</item>
13 <item>/gitlab-ci.yml</item>
14 <item>/abaplint.json</item>
15 <item>/azure-pipelines.yml</item>
16 <item>/devcontainer.json</item>
17 </IGNORE>
18 </DATA>
19 </asx:values>
20 </asx:abap>
```

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

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

2-4: Commit changes.

The screenshot shows a GitHub commit dialog for the file '.abapgit.xml' in the repository 'logistics-business-network-gtt-samples-1'. The file content is displayed on the left, showing XML code. A specific line is highlighted with a red box: '<STARTING_FOLDER>/</STARTING_FOLDER>'. On the right, the 'Commit changes' dialog is open, containing fields for a commit message ('Update .abapgit.xml') and a description, along with radio buttons for committing directly or creating a new branch. The 'Commit changes' button is highlighted with a red border.

Code

Watch 0 Star 0 Fork 16

Pull requests Actions Projects Wiki Security Insights Settings

logistics-business-network-gtt-samples-1/.abapgit.xml in master

Cancel Changes

Edit file Preview changes

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <DATA>
5       <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6       <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</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>
14    </DATA>
15  </asx:values>
16 </asx:abap>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

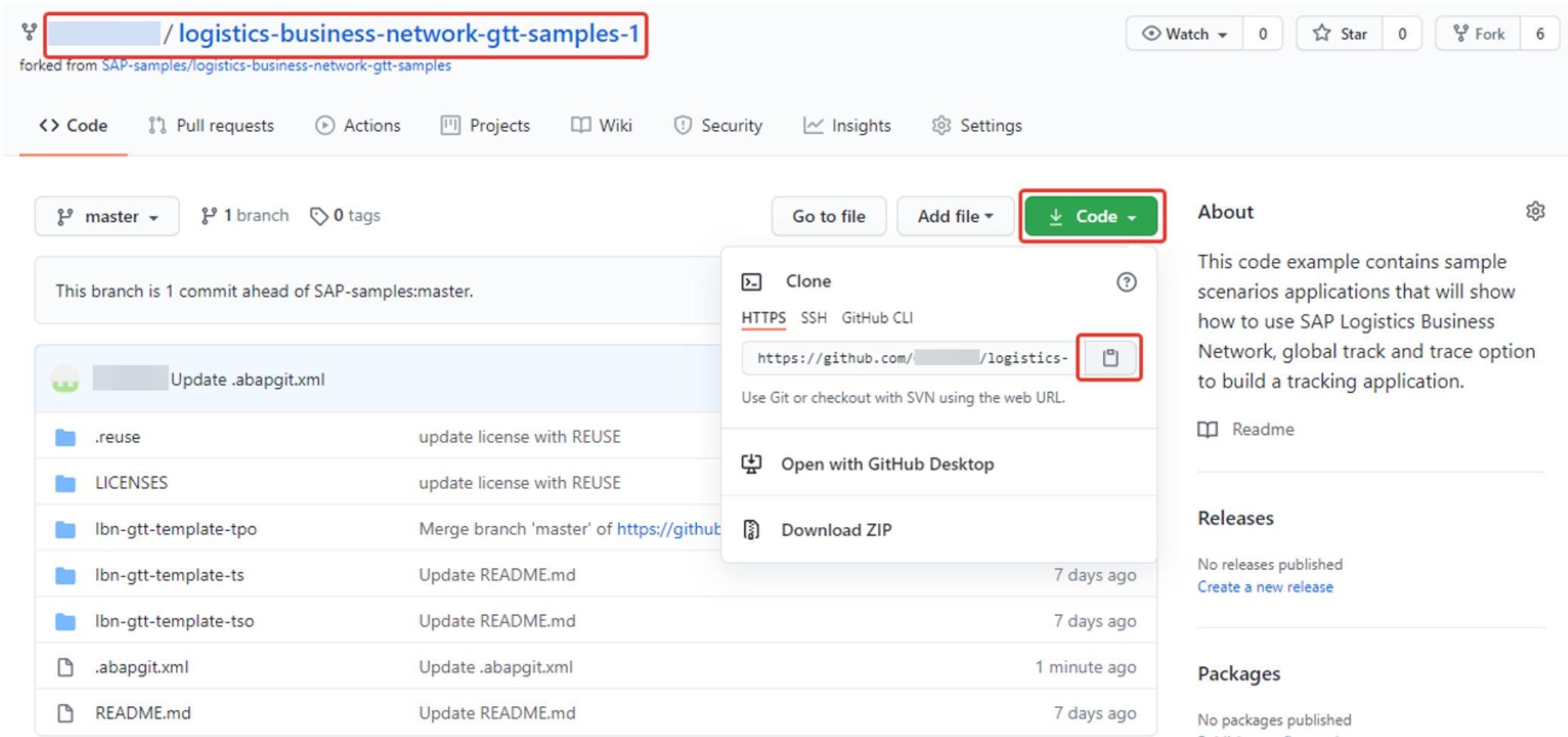
-o- Commit directly to the master branch.

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

Commit changes Cancel

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

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



The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The URL in the address bar is highlighted with a red box. The 'Code' dropdown menu is open, and the 'Clone' section is visible, with the copy icon (a clipboard with a plus sign) highlighted with a red box. The repository has 1 branch and 0 tags. The 'About' section describes the repository as containing sample scenarios applications for SAP Logistics Business Network. The 'Updates' section shows a recent commit to '.abapgit.xml'.

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

Update .abapgit.xml

.reuse update license with REUSE

LICENSES update license with REUSE

Ibn-gtt-template-tpo Merge branch 'master' of https://github.com/SAP-samples/logistics-business-network-gtt-samples-1 into Ibn-gtt-template-tpo

Ibn-gtt-template-ts Update README.md

Ibn-gtt-template-tso Update README.md

.abapgit.xml Update .abapgit.xml

README.md Update README.md

Go to file Add file **Code** 

Clone 

HTTPS SSH GitHub CLI

<https://github.com/SAP-samples/logistics-business-network-gtt-samples-1>

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](#)

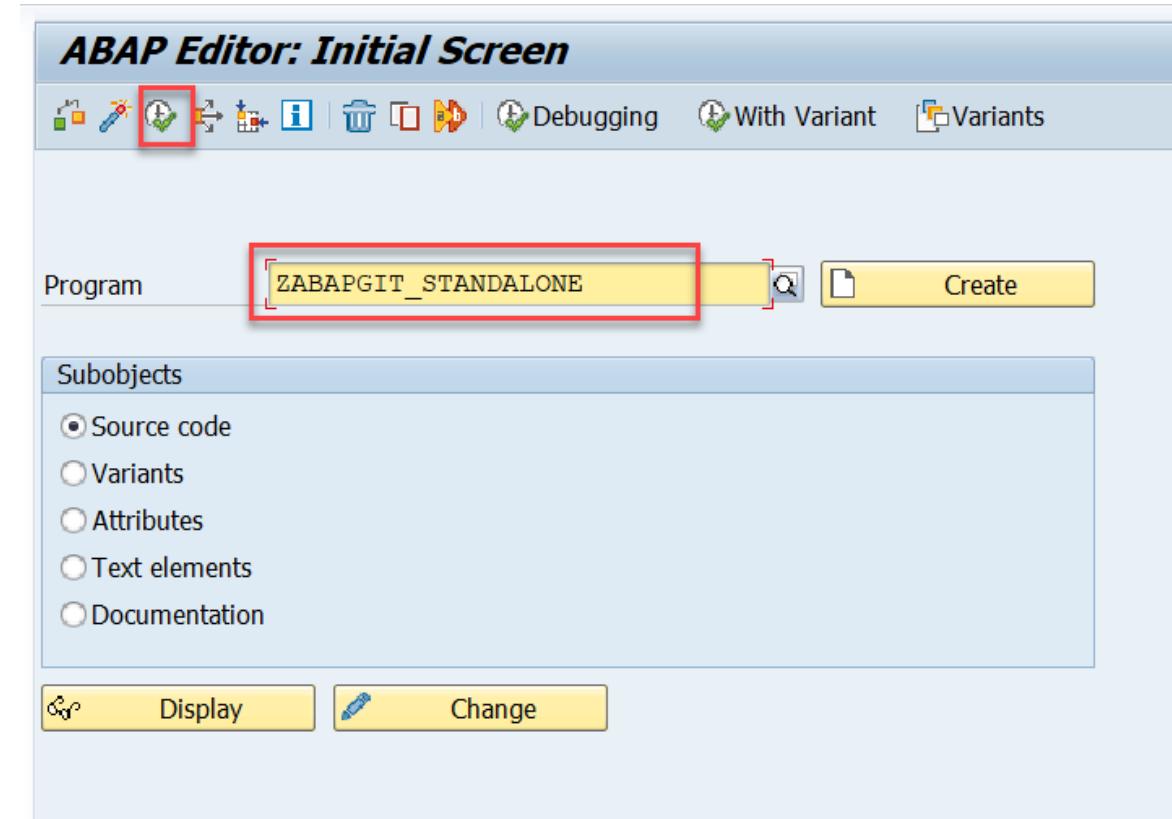
Packages

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

STEP 3: Remove TPOF Repository in ABAPGit

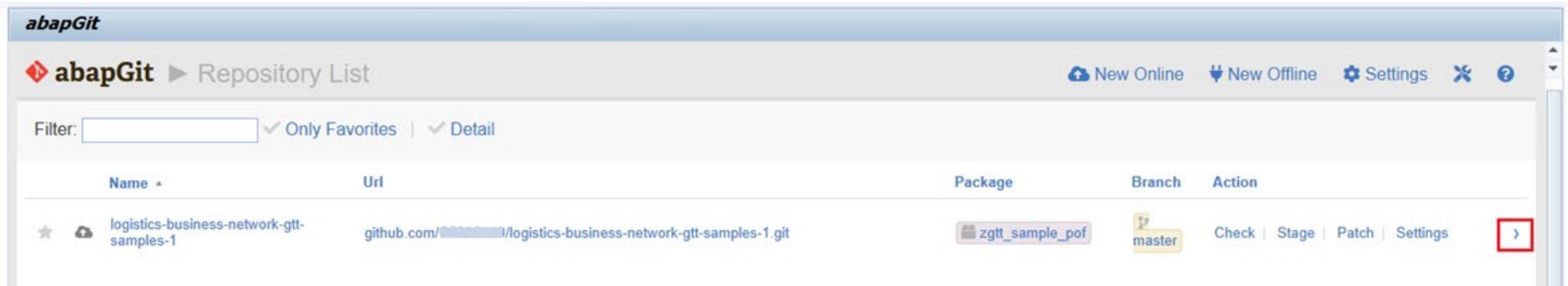
3-1: Enter T-code **SE38** and fill in the report name **ZABAPGIT_STANDALONE**.

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



STEP 3: Remove TPOF Repository in ABAPGit

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

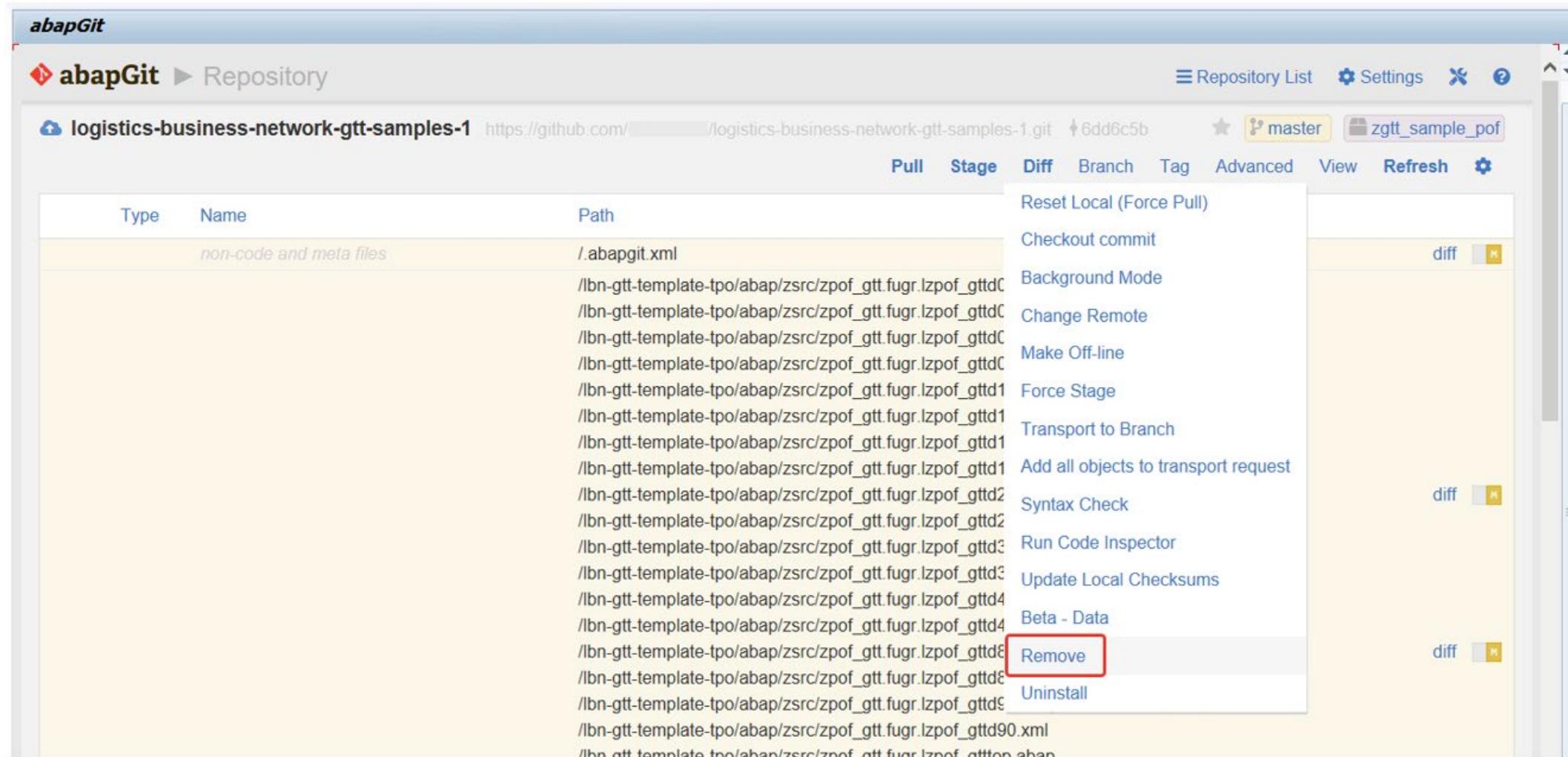


The screenshot shows the ABAPGit application interface. The title bar says "abapGit". The main area is titled "abapGit ► Repository List". There is a "New Online" button, a "New Offline" button, a "Settings" button, and a close button. Below the title, there is a "Filter:" input field, a "Only Favorites" checkbox, and a "Detail" checkbox. The main table has columns: Name, Url, Package, Branch, and Action. One row is visible: "logistics-business-network-gtt-samples-1" with URL "github.com/[REDACTED]/logistics-business-network-gtt-samples-1.git", package "zgtt_sample_pof", branch "master", and action buttons "Check", "Stage", "Patch", "Settings", and a red-bordered ">".

Name	Url	Package	Branch	Action
logistics-business-network-gtt-samples-1	github.com/[REDACTED]/logistics-business-network-gtt-samples-1.git	zgtt_sample_pof	master	Check Stage Patch Settings 

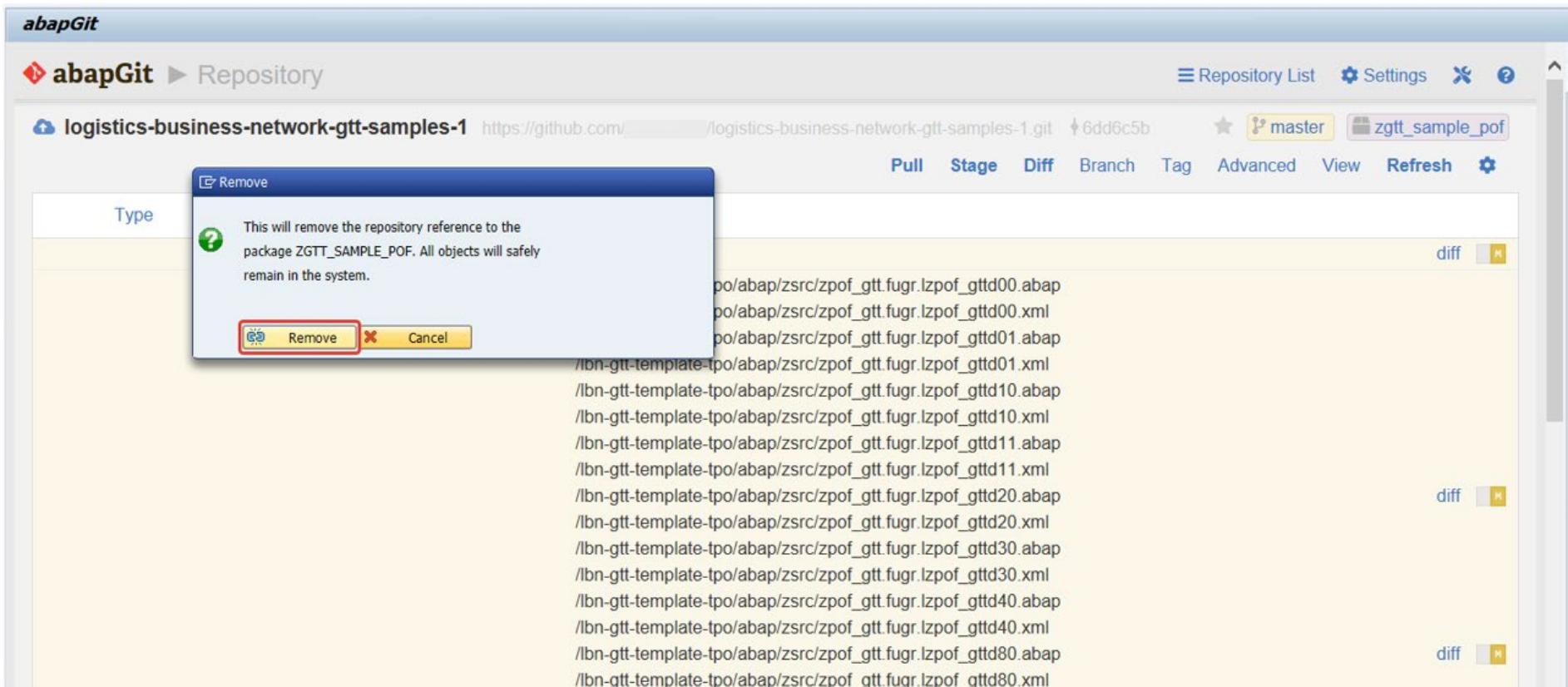
STEP 3: Remove TPOF Repository in ABAPGit

3-4: Under the ‘Advanced’ menu, choose and click ‘Remove’.

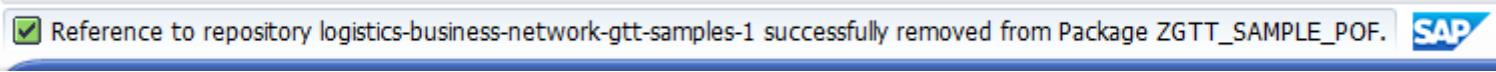


STEP 3: Remove TPOF Repository in ABAPGit

3-5: Click 'Remove' button in the popup window. The reference to TPOF repository will be removed.



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



STEP 4: Download TSOF Code from GitHub

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

The screenshot shows the abapGit application interface. At the top, there is a header bar with the title "abapGit" and a navigation menu with items like "New Online", "New Offline", "Settings", and others. A red box highlights the "New Online" button. Below the header is a search bar labeled "Filter:" and some filter options: "Only Favorites" and "Detail". The main area is a table with columns: "Name", "Url", "Package", "Branch", and "Action". The table is currently empty. At the bottom of the screen, there is a modal dialog box for a repository named "abapGit" version "1.105.0". The dialog shows a message: "Reference to repository logistics-business-network-gtt-samples-1 successfully removed from Package ZGTT_SAMPLE_POF." There is an SAP logo at the bottom right of the dialog. The status bar at the bottom of the application window shows some SAP navigation icons and the text "js: OK".

STEP 4: Download TSOF Code from GitHub

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

<https://github.com/xxxxx/logistics-business-network-gtt-samples-1.git>

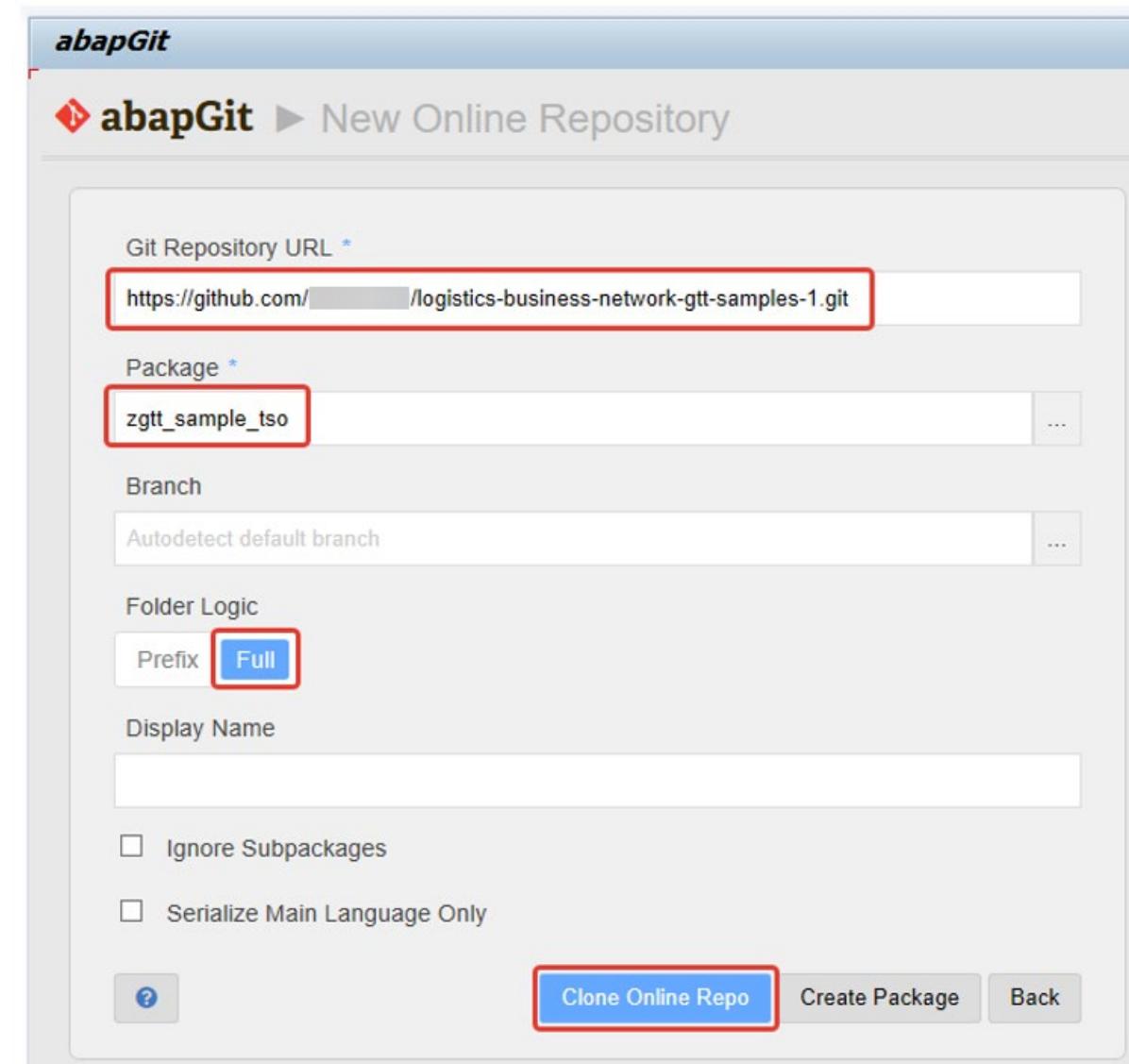
Caution:

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.

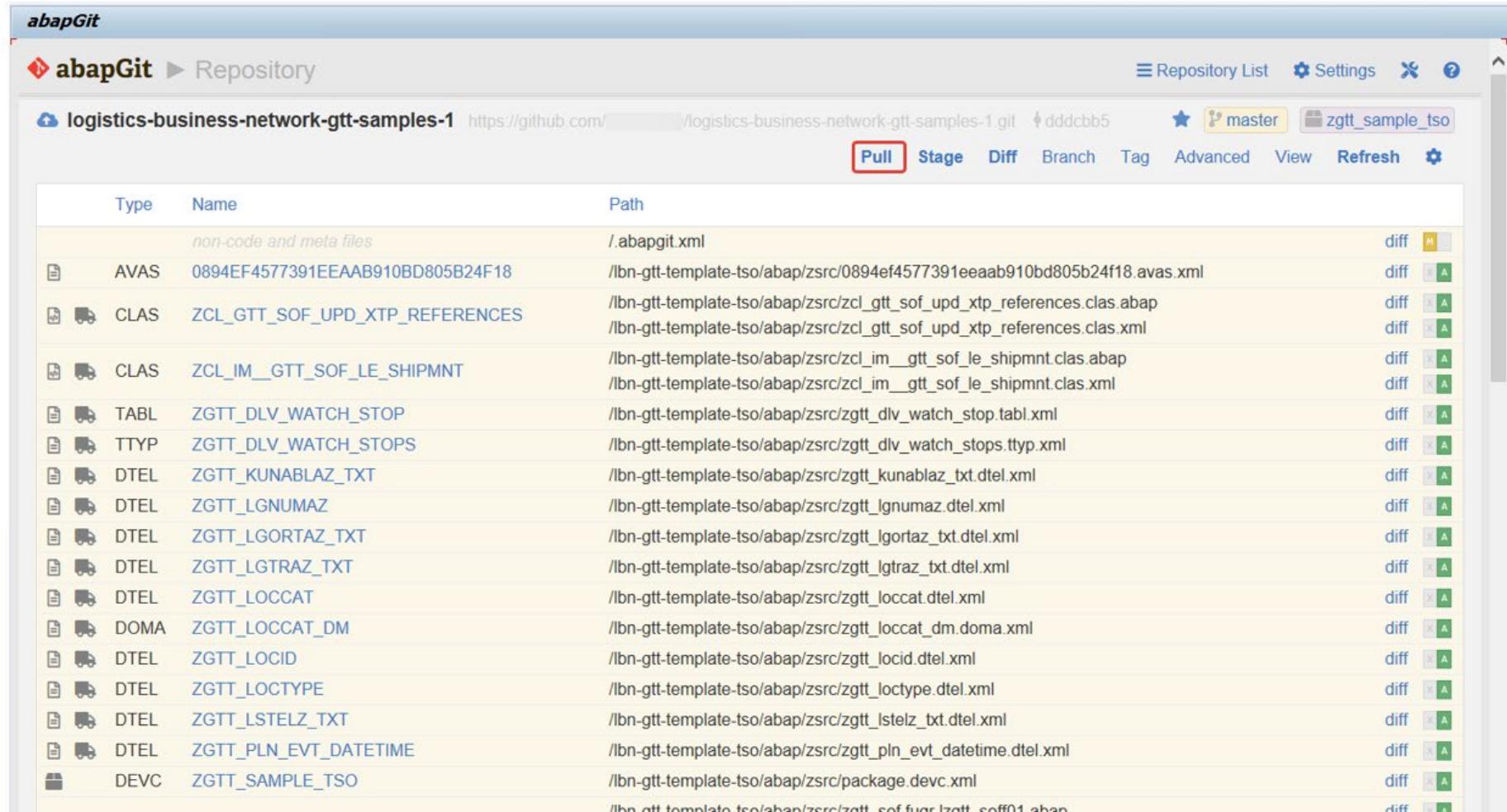
4-4: Set **Full** for **Folder Logic**

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



STEP 4: Download TSOF Code from GitHub

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

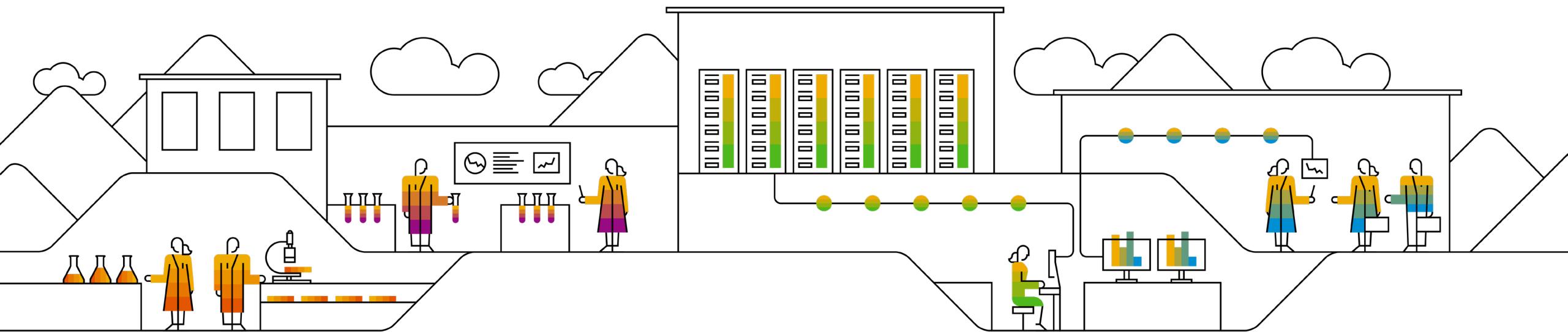


The screenshot shows the abapGit interface for the 'logistics-business-network-gtt-samples-1' repository. The 'Pull' button is highlighted with a red box. The table lists various files and their paths:

Type	Name	Path	diff	A
	non-code and meta files	/abapgit.xml	diff	M
AVAS	0894EF4577391EEAAB910BD805B24F18	/lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml	diff	A
CLAS	ZCL_GTT_SOF_UPD_XTP_REFERENCES	/lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.abap /lbn-gtt-template-tso/abap/zsrc/zcl_gtt_sof_upd_xtp_references.clas.xml	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
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_LGORTAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml	diff	A
DTEL	ZGTT_LGTRAZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml	diff	A
DTEL	ZGTT_LOCCAT	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat.dtel.xml	diff	A
DOMA	ZGTT_LOCCAT_DM	/lbn-gtt-template-tso/abap/zsrc/zggt_loccat_dm.doma.xml	diff	A
DTEL	ZGTT_LOCID	/lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml	diff	A
DTEL	ZGTT_LOCTYPE	/lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml	diff	A
DTEL	ZGTT_LSTELZ_TXT	/lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml	diff	A
DTEL	ZGTT_PLN_EVT_DATETIME	/lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml	diff	A
DEVC	ZGTT_SAMPLE_TSO	/lbn-gtt-template-tso/abap/zsrc/package.devc.xml /lbn-gtt-template-tso/abap/zsrc/zggt_sample_tso.gbsn	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**.

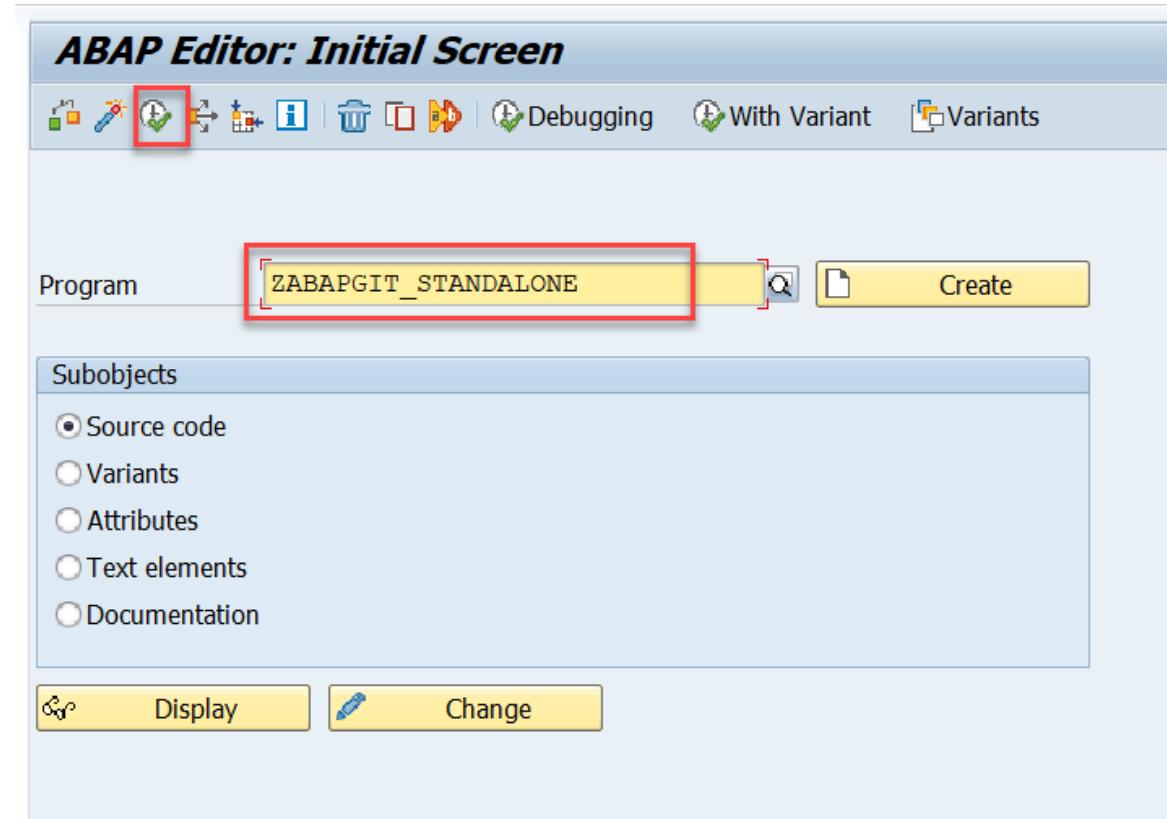
Note: ABAPGit version 1.105.0 is used to create this guide. In case you use different version of ABAPGit, you may face distinctions in interface of the app.

The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". The left sidebar has sections for "Getting Started" (Installation, Upgrading, Uninstalling, UI features), "Setup" (SSL setup, Proxy configuration, Development version), "Online Projects" (Installing online repo, Keeping code up to date, Uninstall repository, First project, Moving package into git, Contributing to a project), "Offline Projects" (Import zip, Export zip), and "Reference" (Repo Settings (abapgit.xml), Supported object types, Icon Legend, User Exits, Authorizations, Namespaces). The main content area starts with a "Summary" section stating that abapGit exists in two flavours: standalone or developer version. It then describes the standalone version as targeted at users and the developer version as targeted at developers. Below this is a "Prerequisites" section requiring SAP BASIS version 702 or higher. The "Install standalone version" section is highlighted with a red border and contains four numbered steps: 1. Download the ABAP code (right click -> save-as) to a file. 2. Via SE38 or SE80, create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version. 3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload. 4. Activate. A note below says typically abapGit will only be used in the development system so it can be installed in a local \$ package (e.g. \$ZABAPGIT). A final note says now you can use abapGit by executing the report in transaction SE38.

STEP 2: Download ABAP Code

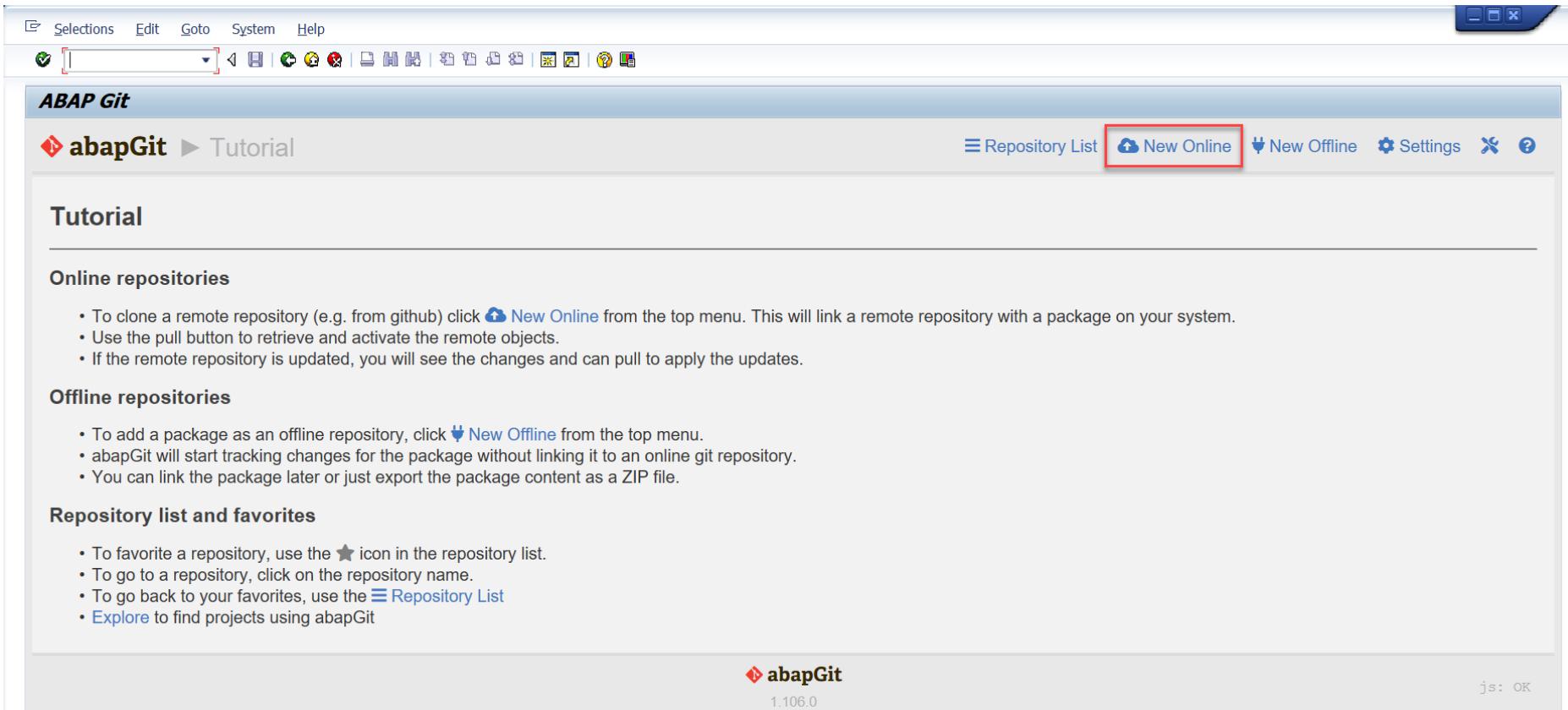
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 2: Download ABAP Code

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



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

STEP 2: Download ABAP Code

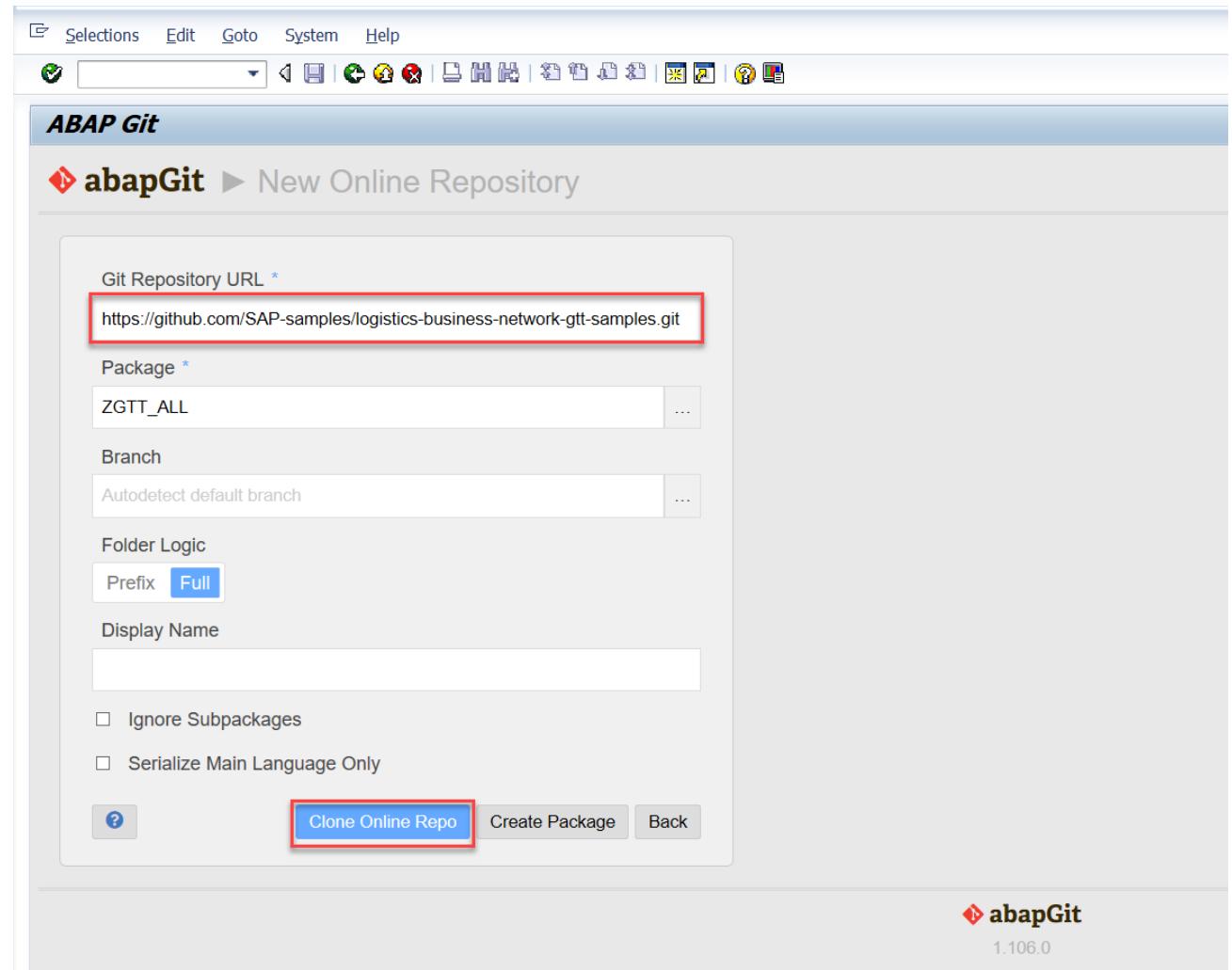
2-4: Fill in the **Git Repository URL**:

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

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

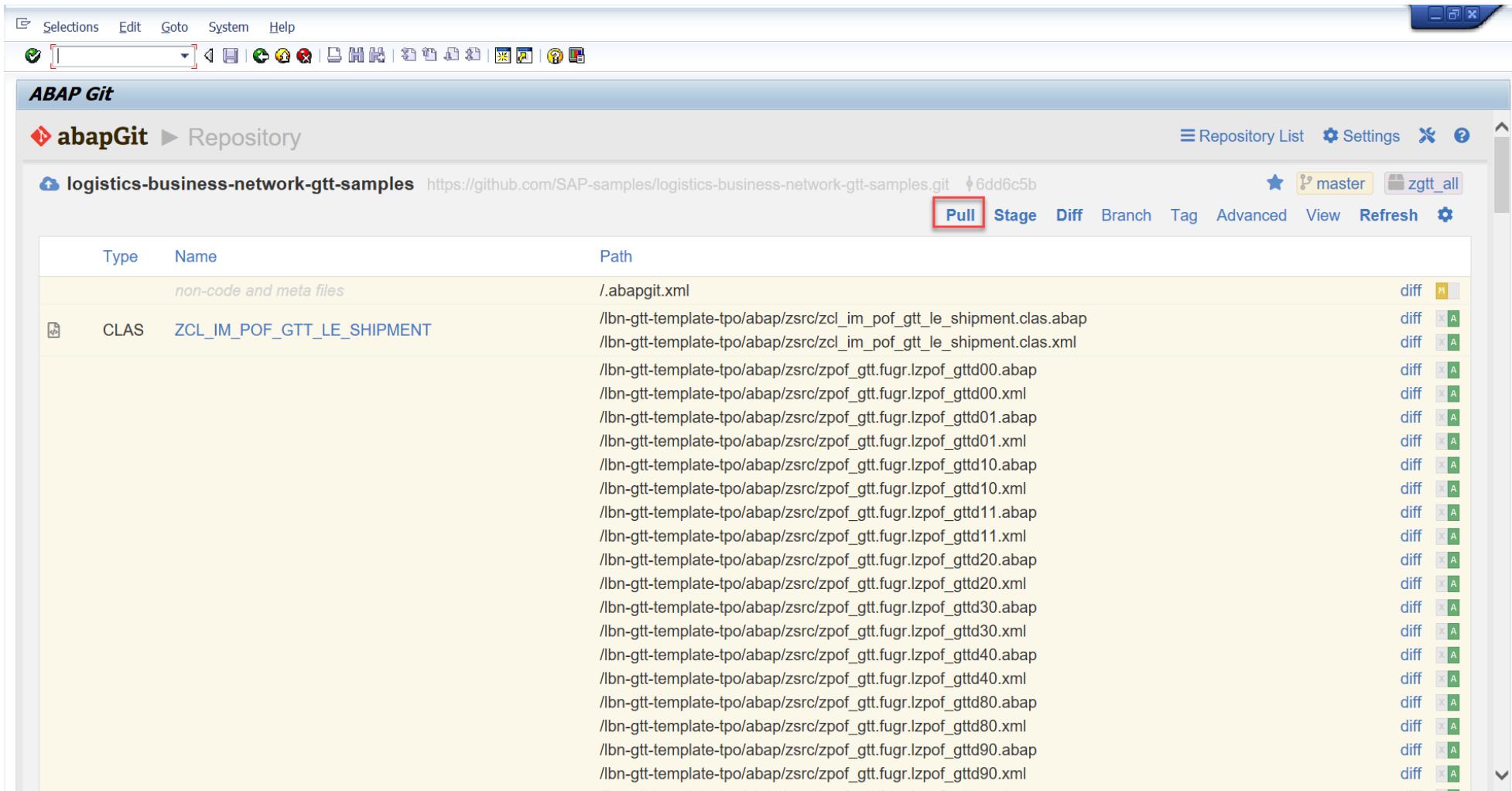
2-6: Set *Full* for **Folder Logic**

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



STEP 2: Download ABAP Code

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

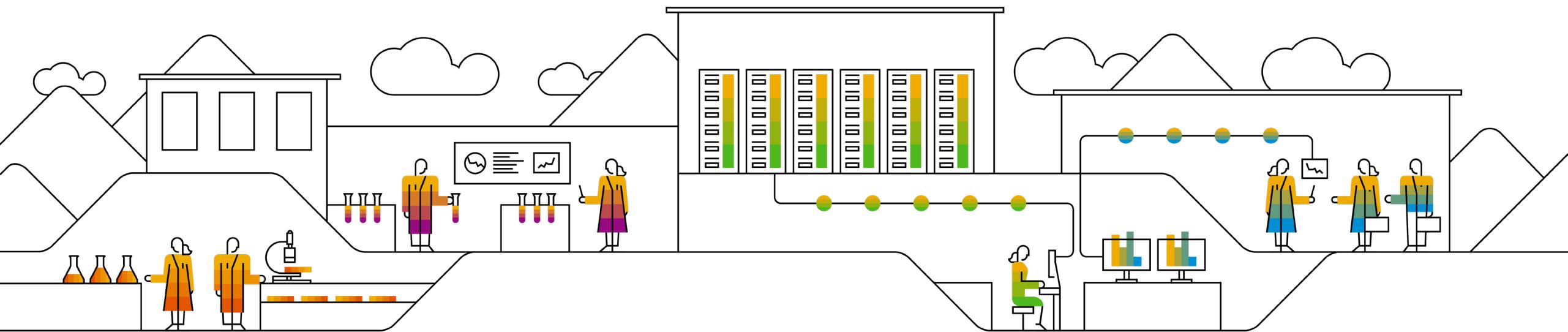


The screenshot shows the ABAP Git interface within SAP. The title bar includes 'Selections', 'Edit', 'Goto', 'System', and 'Help'. Below the title bar is a toolbar with various icons. The main area is titled 'ABAP Git' and shows the path 'abapGit > Repository'. A repository card for 'logistics-business-network-gtt-samples' is displayed, including its URL and a commit hash. The 'Pull' button is highlighted with a red box. Below the card is a table with columns 'Type', 'Name', and 'Path'. The table lists several files and classes, with 'diff' buttons and status indicators (M, A) for each. The table has a vertical scrollbar on the right side.

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

D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

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

The image displays two screenshots illustrating the configuration of Application Object Types (AOT) in SAP and their integration with the SAP Business Network Global Track and Trace (GTT) system.

SAP AOT Configuration: On the left, the "Change View 'Define Application Object Types': Details" screen shows the configuration of an Application Object Type (Appl. Obj. Type) named "ZPOF_GTT_AC_HD". This type is associated with a Business Process Type ("Bus. Proc. Type") of "ESC_PURORD". The "Text" field indicates it is for "Purchase Order Head for Procurement Visibility (GTT) - Acceptance". The "General Data" tab is selected, showing details like sequencing (Seq. No. 10, CI for GTT: ZGTTPOFAC), business object reference (Object Type: BUS2012, PurchaseOrder), and other parameters.

SAP Business Network GTT Integration: On the right, the "Model Details" screen for the "pof" model (Active) shows the "IDOC Integration" tab selected. It maps the "PurchaseOrder" tracked process to the "Others" ERP object type. The "Application Object Type" is set to "ZPOF_GTT_AC_HD". The "Tracked Process / Events" table lists "purchaseOrderEvent" with IDOC "E1EHPAO". The "Fields" table maps fields from the PurchaseOrder entity to IDOC segments:

Field	IDOC Segment	IDOC Field
purchaseOrderNo	E1EHPCP	YN_PO_NUMBER
supplierId	E1EHPCP	YN_PO_SUPPLIER_ID
plannedDeliveryDate	E1EHPCP	YN_PO_DELIVERY_DATE
netValue	E1EHPCP	YN_PO_NET_VALUE
currency	E1EHPCP	YN_PO_CURRENCY
incotermsVersion	E1EHPCP	YN_PO_INCOTERMS_VERSION
incoterms	E1EHPCP	YN_PO_INCOTERMS

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 the Manage Models app in SAP Business Network Global Track and Trace Version 2.

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

The image displays two screenshots illustrating the configuration of Application Object Types (AOT) and the creation of a Tracked Process in the Manage Models app.

AOT Configuration: The left screenshot shows the "Change View 'Define Application Object Types': Details" screen. Under the "Bus. Proc. Type" dropdown, "ESC_PURORD" is selected. In the "Appl. Obj. Type" dropdown, "ZPOF_GTT_AC_HD" is selected, with the description "Purchase Order Head for Procurement Visibility (GTT) - Acceptance" highlighted. The "Tr. ID Code Set" field is set to "PURCHASE_ORDER".

Manage Models App: The right screenshot shows the "Model Details" screen for "pof" (Active). The "Tracked Process" tab is selected. A modal dialog titled "Create Tracked Process" is open, showing the "Name" field set to "PurchaseOrder" and the "Tracking Id Type" field set to "PURCHASE_ORDER".

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 categories:

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

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

For customization of Tracking ID Type and AOT ID, you need to enable the *Determine by Function* option.

For customization of GTT relevance, you need to enable the *Check Function (Function Module)* option.

Extractor	Description
510_WRF_CONTR_01	Control Parameters for Purchase Order (Seasonal Procurement)
CONTR_PARAM_DELIV	Selection of Control parameters for Deliveries in Shipment
OBP10_DELIV	Selection of CPs for Delivery - Outbound Delivery Visibility Process
OBP10_HU_IN_DLV	Selection of CPs for HUs in Delivery - Outbound Delivery Visibility Process
OCB10_CONTAINER	Selection of CPs for Containers in Ocean Carrier Booking Process
OCB10_ORDER	Selection of CPs for Booking Orders in Ocean Carrier Booking Process
ODT20_TOR	Selection of Control Parameters - Transportation Execution Visib. Proc.
ODT30_INS	Selection of Cntrl Parameters - Instruction Execution Visibility Procoress
ODT40_TOR	Selection of Control Parameters - Transportation Execution Visib. Proc.
PCM10_ITEM	Selection of CPs for Purchase Order Item - Procurement Visibility Process
PMF10_NOTIF	Selection of CPs for Notification - Production Malfunction Visibility Process
PMF10_ORDER	Selection of CPs for Manuf. Order - Production Malfunction Visibility Process
RES30_CPARAM	Selection of Control Parameters - Resource Tracking Visibility Process
SNC10_MSGIN	Control Parameter Extractor for SNC Messages
SNC10_PURORD	Control Parameter Extractor for SNC Purchase Order
SNC10_RPLORD	Control Parameter Extractor for SNC Replenishment Order
TRA10_DELIV	Selection of CPs for Deliveries in Road Shipment - Transp. Visibility Process
TRA10_ROADSEA	Selection of CPs for Road/Sea Shipment - Transp. Visibility Process
ZGTT_OBP10_DELIV	Selection of CPs for Delivery - Outbound Delivery Visibility Process
ZGTT_OTE_DE_HDR	Control Parameter Extractor for Outbound Delivery Header
ZGTT_OTE_DE_ITEM	Control Parameter Extractor for Outbound Delivery Item
ZGTT_OTE_SHP_HDR	Control Parameter Extractor for Shipment Header
ZGTT_OTE_SO_HDR	Control Parameter Extractor for Sales Order Header
ZPOF_GTT_OTE_DL_HDR	Control Parameter Extractor for Inbound Delivery Header
ZPOF_GTT_OTE_DL_ITEM	Control Parameter Extractor for Inbound Delivery Item
ZPOF_GTT_OTE_PO_HDR	Control Parameter Extractor for Purchasing Order Header
ZPOF_GTT_OTE_PO_ITEM	Control Parameter Extractor for Purchasing Order Item
ZPOF_GTT_OTE_SH_HDR	Control Parameter Extractor for Shipment Header
ZSST_GTT_OTE_FO_HDR	Control Parameter Extractor for Freight Order

4: Sample Code for Track PO Fulfillment Template App

4-1 To support the Track PO Fulfillment template app, the sample code covers the following cases by function group ZPOF_GTT:

Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_HDR	Control Parameter Extractor for Inbound Delivery Header
Control Parameter Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM	Control Parameter Extractor for Inbound Delivery Item
Control Parameter Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_HDR	Control Parameter Extractor for Purchasing Order Header
Control Parameter Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM	Control Parameter Extractor for Purchasing Order Item
Control Parameter Extractors	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR	Control Parameter Extractor for Shipment Header
Event Data Extractors	ESC_MATDOC	ZPOF_GTT_EE_DL_HDR_GR	Actual event DLV Header Goods Receipt
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA	Actual event DLV Item Put Away
Event Data Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG	Actual event DLV Item Packing
Event Data Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_CONF	Actual event PO Item Confirmation
Event Data Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_DEL	Actual event PO Item Deletion
Event Data Extractors	ESC_MATDOC	ZPOF_GTT_EE_PO_ITEM_GR	Actual event PO Item Goods Receipt
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_ARR	Actual event Shipment Header Arrival
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_CI	Actual event Shipment Header Check In
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_DEP	Actual event Shipment Header Departure
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LE	Actual event Shipment Header Load End
Event Data Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LS	Actual event Shipment Header Load Start
Planned Event Extractors	ESC_DELIV	ZPOF_GTT_EE_DL_HDR	Selection of EEs for Inbound Delivery Item - Procurement Visibility Process
Planned Event Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_HDR	Selection of EEs for Purchasing Order Header - Procurement Visibility Process
Planned Event Extractors	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM	Selection of EEs for Purchasing Order Item - Procurement Visibility Process
Planned Event Extractors	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR	Selection of EEs for Shipment Header - Procurement Visibility Process
Tracking ID Extractors	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM_TID	Tracking ID Extractor for Inbound Delivery Item
Tracking ID Extractors	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM_TID	Tracking ID Extractor for Purchasing Order Item
Tracking ID Extractors	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR_TID	Tracking ID Extractor for Shipment Header

4: Sample Code for Track PO Fulfillment Template App

Continued from the previous table:

Category	Business Process Type	Function Module Name	Description
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_HDR_REL	Appl. Object Type Relevance for Inbound Delivery Header
GTT relevance function of AOT	ESC_DELIV	ZPOF_GTT_OTE_DL_ITEM_REL	Appl. Object Type Relevance for Inbound Delivery Item
GTT relevance function of AOT	ESC_PURORD	ZPOF_GTT_OTE_PO_HDR_REL	Appl. Object Type Relevance for Purchasing Order Header
GTT relevance function of AOT	ESC_PURORD	ZPOF_GTT_OTE_PO_ITEM_REL	Appl. Object Type Relevance for Purchasing Order Item
GTT relevance function of AOT	ESC_SHIPMT	ZPOF_GTT_OTE_SH_HDR_REL	Appl. Object Type Relevance for Shipment Header
GTT relevance function of Event Type	ESC_MATDOC	ZPOF_GTT_EE_DL_HDR_GR_REL	Relevance function for Actual event DLV Header Goods Receipt
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PA_REL	Relevance function for Actual event DLV Item Put Away
GTT relevance function of Event Type	ESC_DELIV	ZPOF_GTT_EE_DL_ITEM_PKNG_REL	Relevance function for Actual event DLV Item Packing
GTT relevance function of Event Type	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_CONF_REL	Relevance function for Actual event PO Item Confirmation
GTT relevance function of Event Type	ESC_PURORD	ZPOF_GTT_EE_PO_ITEM_DEL_REL	Relevance function for Actual event PO Item Deletion
GTT relevance function of Event Type	ESC_MATDOC	ZPOF_GTT_EE_PO_ITEM_GR_REL	Relevance function for Actual event PO Item Goods Receipt
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_ARR_REL	Relevance function for Actual event Header Arrival
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_CI_REL	Relevance function for Actual event Header Check In
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_DEP_REL	Relevance function for Actual event Header Departure
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LE_REL	Relevance function for Actual event Header Load End
GTT relevance function of Event Type	ESC_SHIPMT	ZPOF_GTT_EE_SH_HDR_LS_REL	Relevance function for Actual event Header Load Start
Cross TP Update Function	ESC_DELIV	ZPOF_GTT_CTP_SH_TO_DL	Cross TP Update from Shipment to Delivery

4: Sample Code for Track PO Fulfillment Template App

4-2 To support the Track PO Fulfillment template app, the following extractors should be set up:

Business Process Type	Extractor Type	Extraction Level	Description	Control Tables	Used Function Modules
ESC_PURORD	AOT Type	PO Header	Purchase Order Head for Procurement Visibility (GTT)	PURCHASE_ORDER_HEADER_NEW	ZPOF_GTT_OTE_PO_HDR ZPOF_GTT_OTE_PO_HDR_REL
ESC_PURORD	AOT Type	PO Item	Purchase Order Item for Procurement Visibility (GTT)	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD	ZPOF_GTT_OTE_PO_ITEM ZPOF_GTT_OTE_PO_ITEM_REL ZPOF_GTT_EE_PO_ITEM ZPOF_GTT_OTE_PO_ITEM_TID
ESC_PURORD	Event Type	PO Item	Purchase Order Item Confirmation Event	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD PURCHASE_ORDER_HEADER_OLD	ZPOF_GTT_EE_PO_ITEM_CONF ZPOF_GTT_EE_PO_ITEM_CONF_REL
ESC_PURORD	Event Type	PO Item	Purchase Order Item Deletion Event	PURCHASE_ITEM_NEW PURCHASE_ORDER_HEADER_NEW PURCHASE_ITEM_OLD PURCHASE_ORDER_HEADER_OLD	ZPOF_GTT_EE_PO_ITEM_DEL ZPOF_GTT_EE_PO_ITEM_DEL_REL
ESC_MATDOC	Event Type	PO Item	Purchase Order Item Goods Receipt Event	MATERIAL_SEGMENT MATERIAL_HEADER	ZPOF_GTT_EE_PO_ITEM_GR ZPOF_GTT_EE_PO_ITEM_GR_REL
ESC_DELIV	AOT Type	DLV Header	Inbound Delivery Head for Procurement Visibility (GTT)	DELIVERY_HEADER_NEW DELIVERY_HEADER_OLD	ZPOF_GTT_OTE_DL_HDR ZPOF_GTT_OTE_DL_HDR_REL ZPOF_GTT_EE_DL_HDR
ESC_DELIV	AOT Type	DLV Item	Inbound Delivery Item for Procurement Visibility (GTT)	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD	ZPOF_GTT_OTE_DL_ITEM ZPOF_GTT_OTE_DL_ITEM_REL ZPOF_GTT_EE_DL_ITEM ZPOF_GTT_OTE_DL_ITEM_TID
ESC_DELIV	Event Type	DLV Item	Inbound Delivery Item Put Away Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_ITEM_PA ZPOF_GTT_EE_DL_ITEM_PA_REL

4: Sample Code for Track PO Fulfillment Template App

Continued from the previous table:

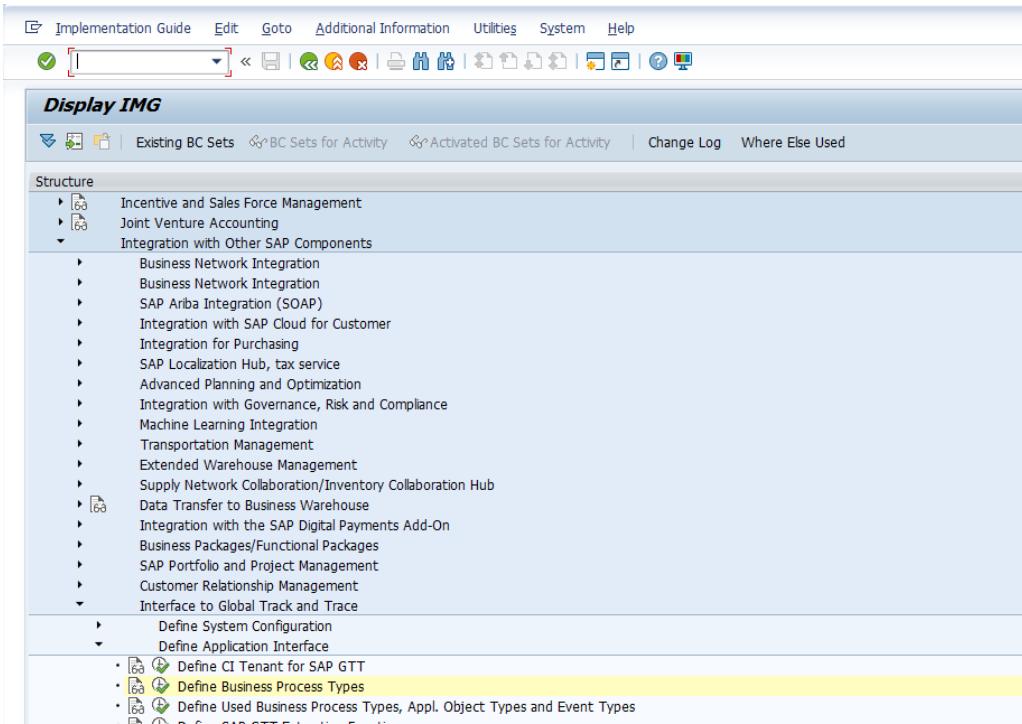
Business Process Type	Extractor Type	Extraction Level	Description	Control Tables	Used Function Modules
ESC_DELIV	Event Type	DLV Item	Inbound Delivery Item Packing Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_ITEM_PKNG ZPOF_GTT_EE_DL_ITEM_PKNG_REL
ESC_MATDOC	Event Type	DLV Header	Inbound Delivery Header Goods Receipt Event	DELIVERY_ITEM_NEW DELIVERY_HEADER_NEW DELIVERY_ITEM_OLD DELIVERY_HEADER_OLD	ZPOF_GTT_EE_DL_HDR_GR ZPOF_GTT_EE_DL_HDR_GR_REL
ESC_SHIPMT	AOT Type	Shipment	Shipment for Procurement Visibility (GTT)	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_OTE_SH_HDR ZPOF_GTT_OTE_SH_HDR_REL ZPOF_GTT_EE_SH_HDR ZPOF_GTT_OTE_SH_HDR_TID
ESC_SHIPMT	Event Type	Shipment	Shipment Arrival Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_ARR ZPOF_GTT_EE_SH_HDR_ARR_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Check In Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_CI ZPOF_GTT_EE_SH_HDR_CI_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Departure Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_DEP ZPOF_GTT_EE_SH_HDR_DEP_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Load End Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_LE ZPOF_GTT_EE_SH_HDR_LE_REL
ESC_SHIPMT	Event Type	Shipment	Shipment Load Start Event	SHIPMENT_HEADER_NEW SHIPMENT_HEADER_OLD	ZPOF_GTT_EE_SH_HDR_LS ZPOF_GTT_EE_SH_HDR_LS_REL

5: Available Contexts for the Extractors' Modules

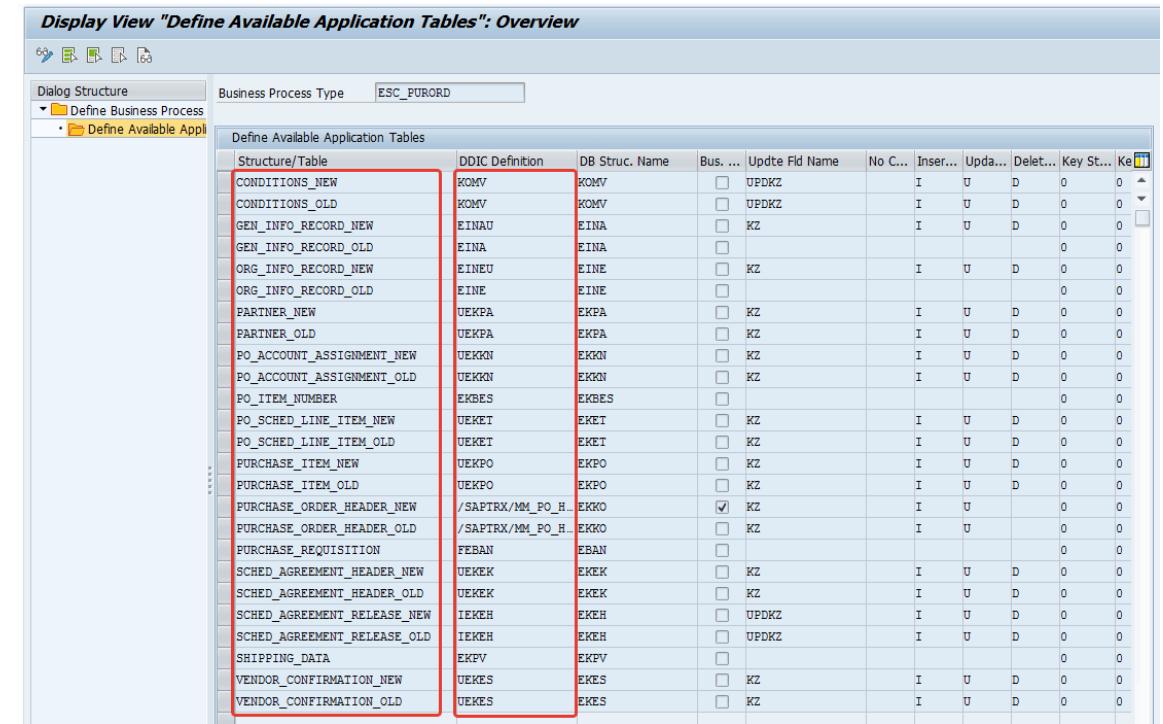
5-1: In Display IMG page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

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

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



The screenshot shows the SAP Display IMG interface. The top menu bar includes Implementation Guide, Edit, Goto, Additional Information, Utilities, System, and Help. Below the menu is a toolbar with various icons. The main area is titled "Display IMG" and contains a breadcrumb trail: Existing BC Sets > BC Sets for Activity > Activated BC Sets for Activity. Below the breadcrumb are tabs for Change Log and Where Else Used. A large tree view under "Structure" lists several categories, including "Integration with Other SAP Components" which is expanded to show sub-options like "Business Network Integration", "SAP Ariba Integration (SOAP)", and "Interface to Global Track and Trace". Under "Interface to Global Track and Trace", there are two items: "Define System Configuration" and "Define Application Interface", with "Define Application Interface" being highlighted by a yellow background. At the bottom of the tree view, there is a list of three specific actions: "Define CI Tenant for SAP GTT", "Define Business Process Types", and "Define Used Business Process Types, Appl. Object Types and Event Types".



The screenshot shows the SAP Display View "Define Available Application Tables": Overview. The title bar indicates the view is for "ESC_PURORD". The left side shows a "Dialog Structure" tree with "Define Business Process" selected, and "Define Available Appl" is expanded. The main area is a grid table with the following columns: Structure/Table, DDIC Definition, DB Struc. Name, Bus. ..., Updt Fld Name, No C..., Inser..., Upda..., Delet..., Key St..., and Ke. The table lists numerous tables, many of which are highlighted with red boxes. Some examples include: CONDITIONS_NEW (DDIC: KOMV, DB: KOMV), CONDITIONS_OLD (DDIC: KOMV, DB: KOMV), GEN_INFO_RECORD_NEW (DDIC: EINAU, DB: EINA), GEN_INFO_RECORD_OLD (DDIC: EINA, DB: EINA), ORG_INFO_RECORD_NEW (DDIC: EINEU, DB: EINE), ORG_INFO_RECORD_OLD (DDIC: EINE, DB: EINE), PARTNER_NEW (DDIC: UEKPA, DB: EKPA), PARTNER_OLD (DDIC: UEKPA, DB: EKPA), PO_ACCOUNT_ASSIGNMENT_NEW (DDIC: UKKN, DB: EKKN), PO_ACCOUNT_ASSIGNMENT_OLD (DDIC: UKKN, DB: EKKN), PO_ITEM_NUMBER (DDIC: EKBES, DB: EKES), PO_SCHED_LINE_ITEM_NEW (DDIC: UKET, DB: EKET), PO_SCHED_LINE_ITEM_OLD (DDIC: UKET, DB: EKET), PURCHASE_ITEM_NEW (DDIC: UKPO, DB: EKPO), PURCHASE_ITEM_OLD (DDIC: UKPO, DB: EKPO), PURCHASE_ORDER_HEADER_NEW (DDIC: /SAPTRX/MM_PO_H, DB: EKKO), PURCHASE_ORDER_HEADER_OLD (DDIC: /SAPTRX/MM_PO_H, DB: EKKO), PURCHASE_REQUSITION (DDIC: FEBAN, DB: EBAN), SCHED_AGREEMENT_HEADER_NEW (DDIC: UEKEK, DB: EKEK), SCHED_AGREEMENT_HEADER_OLD (DDIC: UEKEK, DB: EKEK), SCHED_AGREEMENT_RELEASE_NEW (DDIC: IEKEH, DB: EKEH), SCHED_AGREEMENT_RELEASE_OLD (DDIC: IEKEH, DB: EKEH), SHIPPING_DATA (DDIC: EKPV, DB: EKPV), VENDOR_CONFIRMATION_NEW (DDIC: UKES, DB: EKES), and VENDOR_CONFIRMATION_OLD (DDIC: UKES, DB: EKES).

Structure/Table	DDIC Definition	DB Struc. Name	Bus. ...	Updt Fld Name	No C...	Inser...	Upda...	Delet...	Key St...	Ke
CONDITIONS_NEW	KOMV	KOMV	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
CONDITIONS_OLD	KOMV	KOMV	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
GEN_INFO_RECORD_NEW	EINAU	EINA	<input type="checkbox"/>	KZ	I	U	D	0	0	
GEN_INFO_RECORD_OLD	EINA	EINA	<input type="checkbox"/>					0	0	
ORG_INFO_RECORD_NEW	EINEU	EINE	<input type="checkbox"/>	KZ	I	U	D	0	0	
ORG_INFO_RECORD_OLD	EINE	EINE	<input type="checkbox"/>					0	0	
PARTNER_NEW	UEKPA	EKPA	<input type="checkbox"/>	KZ	I	U	D	0	0	
PARTNER_OLD	UEKPA	EKPA	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ACCOUNT_ASSIGNMENT_NEW	UKKN	EKKN	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ACCOUNT_ASSIGNMENT_OLD	UKKN	EKKN	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_ITEM_NUMBER	EKBES	EKES	<input type="checkbox"/>		I	U	D	0	0	
PO_SCHED_LINE_ITEM_NEW	UKET	EKET	<input type="checkbox"/>	KZ	I	U	D	0	0	
PO_SCHED_LINE_ITEM_OLD	UKET	EKET	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ITEM_NEW	UKPO	EKPO	<input type="checkbox"/>		I	U	D	0	0	
PURCHASE_ITEM_OLD	UKPO	EKPO	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ORDER_HEADER_NEW	/SAPTRX/MM_PO_H	EKKO	<input checked="" type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_ORDER_HEADER_OLD	/SAPTRX/MM_PO_H	EKKO	<input type="checkbox"/>	KZ	I	U	D	0	0	
PURCHASE_REQUSITION	FEBAN	EBAN	<input type="checkbox"/>					0	0	
SCHED_AGREEMENT_HEADER_NEW	UEKEK	EKEK	<input type="checkbox"/>	KZ	I	U	D	0	0	
SCHED_AGREEMENT_HEADER_OLD	UEKEK	EKEK	<input type="checkbox"/>	KZ	I	U	D	0	0	
SCHED_AGREEMENT_RELEASE_NEW	IEKEH	EKEH	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
SCHED_AGREEMENT_RELEASE_OLD	IEKEH	EKEH	<input type="checkbox"/>	UPDKZ	I	U	D	0	0	
SHIPPING_DATA	EKPV	EKPV	<input type="checkbox"/>					0	0	
VENDOR_CONFIRMATION_NEW	UKES	EKES	<input type="checkbox"/>	KZ	I	U	D	0	0	
VENDOR_CONFIRMATION_OLD	UKES	EKES	<input type="checkbox"/>	KZ	I	U	D	0	0	

6: Coding Tips in the GTT Relevance Function Modules

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

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: *ZPOF_GTT_OTE_PO_ITEM_REL*

The image shows two SAP ABAP development environments side-by-side. On the left is the 'Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_REL' window, which displays the source code for the function module. On the right is the 'ABAP Editor: Display Include LZPOF_GTTD20' window, which displays the source code for the include module.

Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_REL

Function Module: ZPOF_GTT_OTE_PO_ITEM_REL active

```
DATA: lt_app_objects TYPE trxas_apppobj_ctabs,  
      io_udm_message TYPE REF TO cx_udm_message,  
      ls_bapiret    TYPE bapiret2.  
  
lt_app_objects = VALUE #( ( i_app_object ) ).  
  
TRY.  
  e_result = lcl_ef_performer->check_relevance(  
    is_definition = VALUE #(  
      maintab = lif_pof_constants->cs_tabledef-po_item_new  
      mastertab = lif_pof_constants->cs_tabledef-po_header_new  
    ).  
    io_bo_factory = NEW lcl_factory_po_item( )  
    iv_appsystypes = i_appsystypes  
    it_all_appl_tables = i_all_appl_tables  
    it_app_objects = lt_app_objects ).  
  
  CATCH cx_udm_message INTO lo_udm_message.  
    lcl_tools->get_errors_log(  
      EXPORTING  
        io_udm_message = lo_udm_message  
        iv_appsystypes = i_appsystypes  
      IMPORTING  
        es_bapiret = ls_bapiret ).  
  
    " add error message  
    APPEND ls_bapiret TO c_logtable.  
  
    " throw corresponding exception  
    CASE lo_udm_message->textid.  
      WHEN lif_ef_constants->cs_errors-stop_processing.  
        RAISE stop_processing.  
      WHEN lif_ef_constants->cs_errors-table_determination.  
        RAISE parameter_error.  
    ENDCASE.  
  ENDTRY.
```

ABAP Editor: Display Include LZPOF_GTTD20

Include: LZPOF_GTTD20 Active

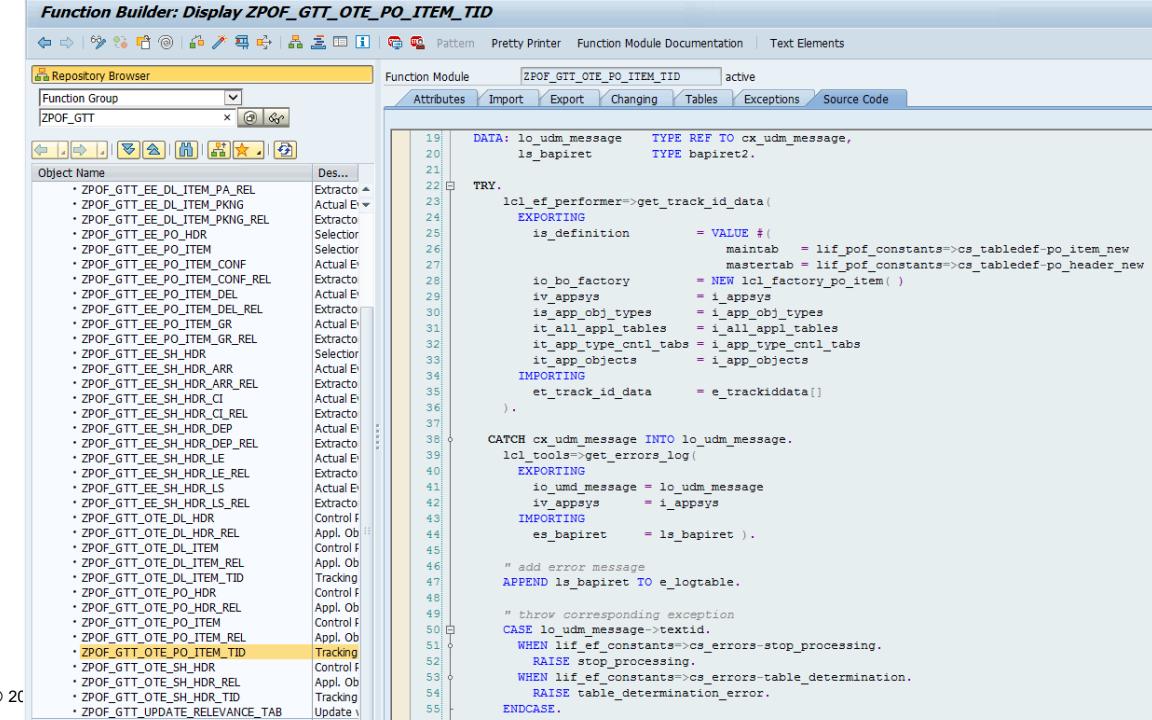
```
METHOD lif_bo_reader->check_relevance.  
  " 1. Basic check of main table which shall be following  
  " the AOT configuration  
  " 2. Check that only 1 PO type is relevance for GTT,  
  " which could be the standard PO type: NB  
  " 3. If it's CREATING PO, always flag TRUE  
  " 4. If it's UPDATING PO, check whether there is any  
  " change for all the above fields or not, comparing  
  " their NEW / OLD value pairs  
  " 5. Don't need to consider DELETING PO, which will be  
  " considered by standard logic of EM framework and  
  " extractors cannot impact this case  
  
  rv_result = lif_ef_constants=>cs_condition-false.  
  
  " is_app_object-maintabdef = lif_pof_constants->cs_tabledef-po_item_new AND  
 IF lcl_po_tools->is_appropriate_po_type( ir_ekko = is_app_object-maintabref ) = abap_true AND  
 lcl_po_tools->is_appropriate_po_item( ir_ekpo = is_app_object-maintabref ) = abap_true AND  
 is_object_changed = is_app_object = is_app_itemobject ) = abap_true.  
  
  CASE is_app_object-update_indicator.  
    WHEN lif_ef_constants=>cs_change_mode-insert.  
      rv_result = lif_ef_constants=>cs_condition-true.  
    WHEN lif_ef_constants=>cs_change_mode-update OR  
      lif_ef_constants=>cs_change_mode-undefined.  
      rv_result = lcl_tools->are_structures_different(  
        ir_data1 = lif_bo_reader->get_data( )  
        is_app_object = is_app_object )  
        ir_data2 = lif_bo_reader->get_data_old( )  
        is_app_object = is_app_object ) ).  
    ENDCASE.  
  ENDIF.  
 ENDMETHOD.  
  
 METHOD lif_bo_reader->get_data.  
   FIELD-SYMBOLS: <ls_item> TYPE ts_po_item.  
   rr_data = NEW ts_po_item( ).  
   ASSIGN rr_data->* TO <ls_item>.  
   IS_CREDIT_CHANGED
```

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

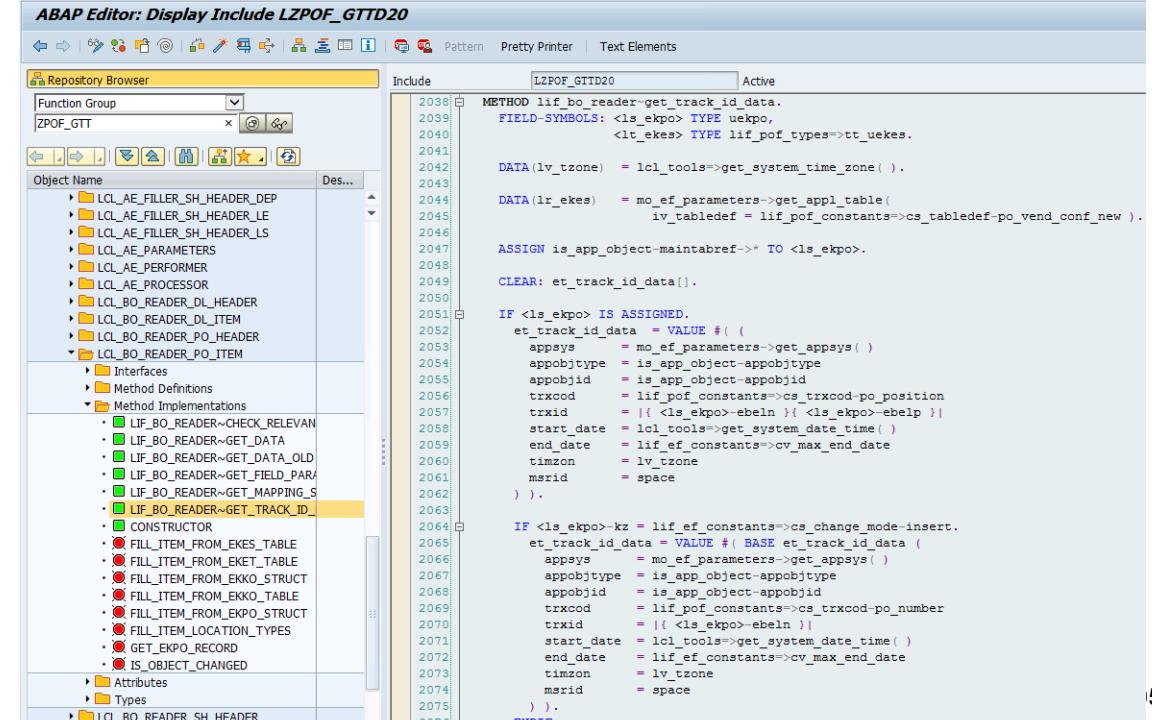
See sample code of function: *ZPOF_GTT_OTE_PO_ITEM_TID*



The screenshot shows the SAP ABAP Function Builder interface with the function module *ZPOF_GTT_OTE_PO_ITEM_TID* selected. The code implements a function with several parameters and handles errors. It uses the *LCL_BO_READER* class to get tracking ID data and the *LCL_BO_FILDER* class to handle header data. The code includes sections for TRY, CATCH, and ENDTRY.

```
Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_TID
Function Module ZPOF_GTT_OTE_PO_ITEM_TID active
Attributes Import Export Changing Tables Exceptions Source Code

19 DATA: lo_udm_message TYPE REF TO cx_udm_message,
20      ls_bapiret TYPE bapiret2.
21
22 TRY.
23   lcl_ef_performer->get_track_id_data(
24     EXPORTING
25       is_definition = VALUE #( maintab = lif_pof_constants->cs_tabledef-po_item_new
26                               mastertab = lif_pof_constants->cs_tabledef-po_header_new )
27     IMPORTING
28       io_bo_factory = NEW lcl_factory_po_item()
29       iv_appsyst = i_appsyst
30       is_app_obj_types = i_app_obj_types
31       it_all_appl_tables = i_all_appl_tables
32       it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
33       it_app_objects = i_app_objects
34   ).
35
36   CATCH cx_udm_message INTO lo_udm_message.
37   lcl_tools->get_errors_log(
38     EXPORTING
39       ic_udm_message = lo_udm_message
40       iv_appsyst = i_appsyst
41     IMPORTING
42       es_bapiret = ls_bapiret .
43
44   " add error message
45   APPEND ls_bapiret TO e_logtable.
46
47   " throw corresponding exception
48   CASE lo_udm_message->textid.
49     WHEN lif_ef_constants=cs_errors-stop_processing.
50       RAISE stop_processing.
51     WHEN lif_ef_constants=cs_errors-table_determination.
52       RAISE table_determination_error.
53   ENDCASE.
54
55 ENDTRY.
```



The screenshot shows the SAP ABAP Editor interface with the include *LZPOF_GTTD20* selected. The code defines methods for reading tracking ID data and handling system time zones. It uses the *LCL_BO_READER* class and its subclasses *LCL_BO_READER_PO_HEADER* and *LCL_BO_READER_PO_ITEM*. The code includes sections for ASSIGN, CLEAR, IF, and ENDIF.

```
ABAP Editor: Display Include LZPOF_GTTD20
Include LZPOF_GTTD20 Active
METHOD lcl_bo_reader->get_track_id_data.
  FIELD-SYMBOLS: <ls_ekpo> TYPE ukpo,
                  <lt_ekes> TYPE lif_pof_types->tt_uekes.

  DATA(lv_tzone) = lcl_tools->get_system_time_zone( ).

  DATA(lr_ekes) = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_pof_constants->cs_tabledef-po_vend_conf_new ).

  ASSIGN is_app_object-maintabref-> TO <ls_ekpo>.

  CLEAR: et_track_id_data[].

  IF <ls_ekpo> IS ASSIGNED.
    et_track_id_data = VALUE #( (
      appsyst = mo_ef_parameters->get_appsyst()
      appobjtype = is_app_object-appobjtype
      appobjid = is_app_object-appobjid
      trxcod = lif_pof_constants->cs_trxcod-po_position
      trxid = ||<ls_ekpo>-ebeln ||<ls_ekpo>-ebelp || )
      start_date = lcl_tools->get_system_date_time()
      end_date = lif_ef_constants=cv_max_end_date
      timzon = lv_tzone
      msrid = space
    ) ).

    IF <ls_ekpo>-kz = lif_ef_constants=cs_change_mode-insert.
      et_track_id_data = VALUE #( BASE et_track_id_data (
        appsyst = mo_ef_parameters->get_appsyst()
        appobjtype = is_app_object-appobjtype
        appobjid = is_app_object-appobjid
        trxcod = lif_pof_constants->cs_trxcod-po_number
        trxid = ||<ls_ekpo>-ebeln || )
        start_date = lcl_tools->get_system_date_time()
        end_date = lif_ef_constants=cv_max_end_date
        timzon = lv_tzone
        msrid = space
      ) ).

    IF <ls_ekpo>-kz = lif_ef_constants=cs_change_mode-update.
      et_track_id_data = VALUE #( BASE et_track_id_data (
        appsyst = mo_ef_parameters->get_appsyst()
        appobjtype = is_app_object-appobjtype
        appobjid = is_app_object-appobjid
        trxcod = lif_pof_constants->cs_trxcod-po_number
        trxid = ||<ls_ekpo>-ebeln || )
        start_date = lcl_tools->get_system_date_time()
        end_date = lif_ef_constants=cv_max_end_date
        timzon = lv_tzone
        msrid = space
      ) ).
```

8: Coding Tips in the Control Parameter Function Modules

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

See sample code of function: *ZPOF_GTT_OTE_PO_ITEM*

8: Coding Tips in the Control Parameter Function Modules

Field mapping is set up in the IDOC Integration tab of the Manage Models app:

The screenshot shows the SAP Manage Models app interface. At the top, there's a navigation bar with tabs: Tracked Process, Field Type Pool, Event Type Pool, Code List, **IDOC Integration** (which is highlighted with a red box), Visibility Provider Integration, Planned Event Extension, and Event to Action. Above the tabs, there are buttons for 'Edit' and 'Draft View'. Below the tabs, there are two dropdown menus: 'Tracked Process' set to 'PurchaseOrder' and 'Integration Switch' set to 'ON'. The main content area is divided into sections: 'Tracked Process Mapping' (with 'ERP Object Type: Others' and 'Application Object Type: ZPOF_GTT_AC_HD'), 'Tracked Process / Events (1)' (listing 'PurchaseOrderEvent' with IDOC 'E1EHPAO'), and 'Fields' (a table listing field mappings). A red box highlights the 'Fields' table.

Field	IDOC Segment	IDOC Field
purchaseOrderNo	E1EHPCP	YN_PO_NUMBER
supplierId	E1EHPCP	YN_PO_SUPPLIER_ID
plannedDeliveryDate	E1EHPCP	YN_PO_DELIVERY_DATE
netValue	E1EHPCP	YN_PO_NET_VALUE
currency	E1EHPCP	YN_PO_CURRENCY
incotermsVersion	E1EHPCP	YN_PO_INCOTERMS_VERSION
incoterms	E1EHPCP	YN_PO_INCOTERMS
incotermsLocation	E1EHPCP	YN_PO_INCOTERMS_LOCATION

8: Coding Tips in the Control Parameter Function Modules

Main logic of Purchase Order Item is implemented in class LCL_BO_READER_PO_ITEM

Function Module ZPOF_GTT_OTE_PO_ITEM active

Attributes Import Export Changing Tables Exceptions Source Code

```
19 DATA: lo_udm_message      TYPE REF TO cx_udm_message,
20      ls_bapiret        TYPE bapiret2.
21
22 TRY.
23   lcl_ef_performer->get_control_data(
24     EXPORTING
25       is_definition      = VALUE #(
26         maintab           = lif_pof_constants->cs_tabledef-po_item_new
27         mastertab          = lif_pof_constants->cs_tabledef-po_header_new )
28   io_bo_factory      = NEW lcl_factory_po_item( )
29   iv_appsps          = i_appsps
30   is_app_obj_types   = i_app_obj_types
31   it_all_appl_tables = i_all_appl_tables
32   it_app_type_cntl_tabs = i_app_type_cntl_tabs
33   it_app_objects     = i_app_objects
34
35 CHANGING
36   ct_control_data    = e_control_data[] ).
37
38 CATCH cx_udm_message INTO lo_udm_message.
39   lcl_tools->get_errors_log(
40     EXPORTING
41       io_udm_message = lo_udm_message
42       iv_appsps     = i_appsps
43     IMPORTING
44       es_bapiret    = ls_bapiret ).
45
46 " add error message
47 APPEND ls_bapiret TO e_logtable.
48
49 " throw corresponding exception
50 CASE lo_udm_message->textid.
51   WHEN lif_ef_constants->cs_errors-stop_processing.
52     RAISE stop_processing.
53   WHEN lif_ef_constants->cs_errors-table_determination.
54     RAISE table_determination_error.
55 ENDCASE.
56 ENDTRY.
57 ENDFUNCTION.
```

ABAP Editor: Display Include LZPOF_GTTD20

Repository Browser

Include LZPOF_GTTD20 Active

```
19841 METHOD lif_bo_reader~get_data.
19842   FIELD-SYMBOLS: <ls_item>      TYPE ts_po_item.
19843
19844   rr_data  = NEW ts_po_item( ).
19845
19846   ASSIGN rr_data->* TO <ls_item>.
19847
19848   fill_item_from_ekko_struct(
19849     EXPORTING
19850       ir_ekko      = is_app_object-mastertabref
19851     CHANGING
19852       cs_po_item   = <ls_item> .
19853
19854   fill_item_from_ekpo_struct(
19855     EXPORTING
19856       ir_ekpo      = is_app_object-maintabref
19857     CHANGING
19858       cs_po_item   = <ls_item> .
19859
19860   fill_item_from_eket_table(
19861     EXPORTING
19862       ir_ekpo      = is_app_object-maintabref
19863       ir_eket      = mo_ef_parameters->get_appl_table(
19864         iv_tabledef = lif_pof_constants->cs_tabledef-po_sched_new )
19865     CHANGING
19866       cs_po_item   = <ls_item> .
19867
19868   fill_item_from_ekes_table(
19869     EXPORTING
19870       ir_ekpo      = is_app_object-maintabref
19871       ir_ekes      = mo_ef_parameters->get_appl_table(
19872         iv_tabledef = lif_pof_constants->cs_tabledef-po_vend_conf_new )
19873     CHANGING
19874       cs_po_item   = <ls_item> .
19875
19876   fill_item_location_types(
19877     CHANGING
19878       cs_po_item   = <ls_item> .
19879
19880 ENDMETHOD.
```

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 in the output table *E_EXPEVENTDATA*.
3. As default when no change has been made on the model configuration, SAP Business Network Global Track and Trace Version 2 will ask for full transport for all the planned events, which means that all the events needs to be extracted in all cases, no matter whether their values have been changed. If nothing is transported, the planned events will be removed in SAP Business Network Global Track and Trace Version 2.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT_EXP_DATETIME* is optional, but needs to be filled in 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 in with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* app in SAP Business Network Global Track and Trace Version 2.
7. The field *LOCID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function: *ZPOF_GTT_EE_PO_ITEM*

pof Active

Purchase Order Fulfillment

Namespace: com.lbnrgttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List **IDOC Integration** Visibility Provider

Tracked Process: PurchaseOrderItem

Tracked Process Mapping

ERP Object Type: Others

Name	IDOC	Event Code
Tracked Process		
PurchaseOrderItemEvent	E1EHPAO	

Event Types		
ConfirmationEvent	E1EVMHDR02	CONFIRMATION
GoodsReceipt	E1EVMHDR02	GOODS_RECEIPT
DeletionEvent	E1EVMHDR02	DELETION
UndeletionEvent	E1EVMHDR02	UNDELETION

9: Coding Tips in the Planned Event Function Modules

Main logic of Purchase Order Item Planned Events is implemented in class LCL_PE_FILLER_PO_ITEM

The screenshot shows two SAP ABAP development environments. On the left, the 'Function Builder: Display ZPOF_GTT_EE_PO_ITEM' window displays the source code for the ZPOF_GTT_EE_PO_ITEM function module. The code implements planned events for purchase order items, utilizing the LCL_PE_FILLER_PO_ITEM class. On the right, the 'ABAP Editor: Display Include LZPOF_GTTD30' window shows the LZPOF_GTTD30 include, which contains the definition of the LCL_PE_FILLER_PO_ITEM class, including its method implementations for various planned events.

```
Function Builder: Display ZPOF_GTT_EE_PO_ITEM
Function Module ZPOF_GTT_EE_PO_ITEM active
Attributes Import Export Changing Tables Exceptions Source Code

21 DATA: lo_udm_message    TYPE REF TO cx_udm_message,
22      ls_bapiret     TYPE bapiret2.
23
24 CLEAR e_logtable[].
25
26 TRY.
27   lcl_ef_performer->get_planned_events(
28     EXPORTING
29       is_definition      = VALUE #(
30         maintab           = lif_pof_constants->cs_tabledef-po_item new
31         mastertab          = lif_pof_constants->cs_tabledef-po_header_new )
32
33   io_factory          = NEW lcl_factory_po_item( )
34   iv_appsyst          = i_appsyst
35   is_app_obj_types    = i_app_obj_types
36   it_all_appl_tables  = i_all_appl_tables
37   it_app_type_cntl_tabs = i_app_type_cntl_tabs
38   it_app_objects       = i_app_objects
39
40   CHANGING
41     ct_expeventdata    = e_expeventdata[]
42     ct_measrmntdata   = e_measrmntdata[]
43     ct_infodata        = e_infodata[]
44
45   CATCH cx_udm_message INTO lo_udm_message.
46   lcl_tools->get_errors_log(
47     EXPORTING
48       io_udm_message = lo_udm_message
49       iv_appsyst    = i_appsyst
50
51     IMPORTING
52       es_bapiret    = ls_bapiret .
53
54   " add error message
55   APPEND ls_bapiret TO e_logtable.
56
57   " throw corresponding exception
58 CASE lo_udm_message->txid.
59   WHEN lif_ef_constants=>cs_errors-stop_processing.
60     RAISE stop_processing.
61   WHEN lif_ef_constants=>cs_errors-table_determination.
62     RAISE table_determination_error.
63 ENDCASE.
64
65 ENDTRY.
66
67 ENDFUNCTION.
```



```
ABAP Editor: Display Include LZPOF_GTTD30
Include LZPOF_GTTD30 Active
425 ENDIF.
426 ENDMETHOD.
427
428 METHOD lif_pe_filler~get_planed_events.
429   add_confirmation_event(
430     EXPORTING
431       is_app_objects = is_app_objects
432
433     CHANGING
434       ct_expeventdata = ct_expeventdata .
435
436     add_goods_receipt_event(
437       EXPORTING
438         is_app_objects = is_app_objects
439
440     CHANGING
441       ct_expeventdata = ct_expeventdata .
442
443   ENDMETHOD.
444
445 ENDCCLASS.
```

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

See sample code of function: *ZPOF_GTT_EE_PO_ITEM_CONF*

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:

pof Active

Purchase Order Fulfillment

Namespace: com.lbngttsamples.gtt.app.pof Correlation Level: 4

Tracked Process Field Type Pool Event Type Pool Code List **IDOC Integration** Visibility Provider Integration Planned Event Extension Event to Action

Tracked Process: PurchaseOrderItem ▼ Integration Switch: ON

Tracked Process Mapping

ERP Object Type: Others Application Object Type: ZPOF_GTT_AC_ITEM

Tracked Process / Events (5)

Name	IDOC	Event Code
Tracked Process		
PurchaseOrderItemEvent	E1EHPAO	
Event Types		
ConfirmationEvent	E1EVMPAR02	CONFIRMATION
GoodsReceipt	E1EVMPAR02	GOODS_RECEIPT
DeletionEvent	E1EVMPAR02	DELETION
UndeletionEvent	E1EVMPAR02	UNDELETION

Fields

Field	IDOC Segment	IDOC Field
quantity	E1EVMPAR	QUANTITY
confirmType	E1EVMPAR	CONFIRM_TYPE

10: Coding Tips in the Event Data Function Modules

Main logic of Purchase Order Item Confirmation event is implemented in class LCL_AE_FILLER_PO_ITEM_CONF

Function Module ZPOF_GTT_EE_PO_ITEM_CONF active

Attributes Import Export Changing Tables Exceptions Source Code

```
59: DATA: lo_udm_message      TYPE REF TO cx_udm_message,
60:         ls_bapiret        TYPE bapiret2.
61:
62: TRY.
63:   lcl_ae_performer->get_event_data(
64:     EXPORTING
65:       is_definition      = VALUE #((
66:         maintab           = lif_pof_constants->cs_tabledef-po_item_new
67:         masterstab         = lif_pof_constants->cs_tabledef-po_header_new )
68:       io_ae_factory      = NEW lcl_ae_factory_po_item_conf( )
69:       iv_appsyst         = i_appsyst
70:       is_event_type      = i_event_type
71:       it_all_appl_tables = i_all_appl_tables
72:       it_event_type_cntl_tabs = i_event_type_cntl_tabs
73:       it_events          = i_events
74:     CHANGING
75:       ct_eventid_map     = c_eventid_map[]
76:       ct_trackingheader  = ct_trackingheader[]
77:       ct_tracklocation   = ct_tracklocation[]
78:       ct_trackreferences = ct_trackreferences[]
79:       ct_trackparameters = ct_trackparameters[]
80:   ).
81: CATCH cx_udm_message INTO lo_udm_message.
82:   lcl_tools->get_errors_log(
83:     EXPORTING
84:       io_udm_message    = lo_udm_message
85:       iv_appsyst         = i_appsyst
86:     IMPORTING
87:       es_bapiret        = ls_bapiret .
88:
89:   " add error message
90:   APPEND ls_bapiret TO ct_logtable.
91:
92:   " throw corresponding exception
93:   CASE lo_udm_message->textid.
94:     WHEN lif_ef_constants->cs_errors-stop_processing.
95:       RAISE stop_processing.
96:     WHEN lif_ef_constants->cs_errors-table_determination.
97:       RAISE event_data_error.
98:   ENDCASE.
99:
100: ENDTRY.
101: ENDFUNCTION.
```

ABAP Editor: Display Include LZPOF_GTTD40

Repository Browser

Object Name

- ZPOF_GTT_UPDATE_RELEVANCE_TAB
- Interfaces
- Classes
 - LCL_AE_FACTORY
 - LCL_AE_FACTORY_DL_ITEM_GR
 - LCL_AE_FACTORY_DL_ITEM_PA
 - LCL_AE_FACTORY_DL_ITEM_PKNG
 - LCL_AE_FACTORY_PO_ITEM_CONF
 - LCL_AE_FACTORY_PO_ITEM_DEL
 - LCL_AE_FACTORY_PO_ITEM_GR
 - LCL_AE_FACTORY_SH_HEADER_ARR
 - LCL_AE_FACTORY_SH_HEADER_CI
 - LCL_AE_FACTORY_SH_HEADER_DEP
 - LCL_AE_FACTORY_SH_HEADER_LE
 - LCL_AE_FACTORY_SH_HEADER_LS
 - LCL_AE_FILLER_DL_ITEM_GR
 - LCL_AE_FILLER_DL_ITEM_PA
 - LCL_AE_FILLER_DL_ITEM_PKNG
 - LCL_AE_FILLER_PO_ITEM_CONF
- Method Implementations
 - LIF_AE_FILLER~CHECK_RELEVANCE
 - LIF_AE_FILLER~GET_EVENT_DATA
 - CONSTRUCTOR
 - GET_CONFIRMATION_QUANTITY
 - GET_CONFIRMATION_QUANTITY_
 - HAS_CHANGES
 - IS_APPROPRIATE_CONF CONTRC
 - IS_APPROPRIATE_CONF_TYPE
- Attributes
- LCL_AE_FILLER_PO_ITEM_DEL
- LCL_AE_FILLER_PO_ITEM_GR
- LCL_AE_FILLER_SH_HEADER_ARR
- LCL_AE_FILLER_SH_HEADER_BH

Include LZPOF_GTTD40 Active

```
98: METHOD lif_ae_filler~get_event_data.
99:   DATA (lv_difference) = get_confirmation_quantity_diff(
100:         is_events = is_events ).
101:
102:   ct_trackingheader = VALUE #( BASE ct_trackingheader (
103:     language      = sy-langu
104:     trxid        = lcl_po_tools->get_tracking_id_po_item(
105:       ir_ekpo = is_events-maintabref )
106:     trxcod        = lif_pof_constants->cs_trxcod-po_position
107:     evtcnt        = is_events-eventid
108:     evtid         = lif_pof_constants->cs_milestone-po_confirmation
109:     evtdat        = sy-datum
110:     evttim        = sy-uzzeit
111:     evtzon        = lcl_tools->get_system_time_zone( )
112:   )).
113:
114:   ct_eventid_map = VALUE #( BASE ct_eventid_map (
115:     eventid      = is_events-eventid
116:     evtcnt        = is_events-eventid
117:   )).
118:
119:   ct_tracklocation = VALUE #( BASE ct_tracklocation (
120:     evtcnt        = is_events-eventid
121:     loccod        = lif_ef_constants->cs_loc_types-plant
122:     locidl        = lcl_tools->get_field_of_structure(
123:       ir_struct_data = is_events-maintabref
124:       iv_field_name = 'WERKS' )
125:   )).
126:
127:
128:   ct_trackparameters = VALUE #( BASE ct_trackparameters (
129:     evtcnt        = is_events-eventid
130:     param_name    = lif_pof_constants->cs_event_param-quantity
131:     param_value   = lcl_tools->get_pretty_value( iv_value = lv_difference )
132:   )).
133:
134:
135:   ct_trackparameters = VALUE #( BASE ct_trackparameters (
136:     evtcnt        = is_events-eventid
137:     param_name    = lif_pof_constants->cs_event_param-confirm_type
138:     param_value   = lif_pof_constants->cs_relevance-ebtyp
139:   )).
140: ENDMETHOD.
```

11: Enhancement Codes for Cross-process Tracking

The Track PO Fulfillment template app asks for cross-process tracking that is used in the following:

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

IMPORTANT: To enable cross-process tracking, update the sample code below after downloading:

- Update Inbound Delivery Header and Item AOT type Mask in Method GET_AOTYPE_RESTRICTIONS of LCL_CTP_SENDER_SH_TO_DL_HEAD and LCL_CTP_SENDER_SH_TO_DL_ITEM

```
ABAP Editor: Display Include LZPOF_GTTD80
Include LZPOF_GTTD80 Active
1551 CLASS lcl_ctp_sender_sh_to_dl_head IMPLEMENTATION.
1552 METHOD get_aotype_restrictions.
1553   et_aotype = VALUE #(
1554     low      = 'ZPOF_GTT_*_DL_HD'
1555     option   = 'CP'
1556     sign    = 'I'
1557   ) .
1558 ENDMETHOD.
1559
1560 METHOD get_instance.
1561   DATA(lt_trk_obj_type) = VALUE tt_trk_obj_type(
1562     ( lif_ef_constants=>cs_trk_obj_type-esc_shipmt )
1563     ( lif_ef_constants=>cs_trk_obj_type-esc_deliv )
1564   ).
```

11: Enhancement Codes for Cross-process Tracking

The cross-process tracking scenarios cover the following:

Shipment -> Inbound Delivery and Inbound 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 update IDOC. If the event IDOC approaches SAP Business Network Global Track and Trace prior to the order IDOC, it will lead to a processing failure.

This issue has been minimized now, see the solution provided in these topics:

- [8: Coding Tips in the Control Parameter Function Modules](#)
- [10: Coding Tips in the Event Data Function Modules](#)
- [13: Solution of IDOC Sequencing Issue](#)

13: Solution of IDOC Sequencing Issue

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

2. Create CI tenant.

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

SAP Global Track & Trace Definitions			
CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTPOFAC2	ZGTTPOFAC	GTT2.0 Logistics Business N...	CI For GTT Purchasing Order Sample APP - Acceptance

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`

RFC Destination ZGTT_POF_PO_TP_ACC2

Connection Test

RFC Destination	ZGTT_POF_PO_TP_ACC2
Connection Type	G HTTP Connection to External Server
Description	
Description 1	RFC for Tracted Process of POF Sample Application to Acceptance
Description 2	
Description 3	

Administration Technical Settings Logon & Security Special Options

Target System Settings

Host		Port	443
Path Prefix	<code>/api/idoc/em/v1/TrackedProcessAndEvent</code>		

Thank you.



Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.