



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

SAP Business Network for Logistics
March 2021

PUBLIC

THE BEST RUN



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



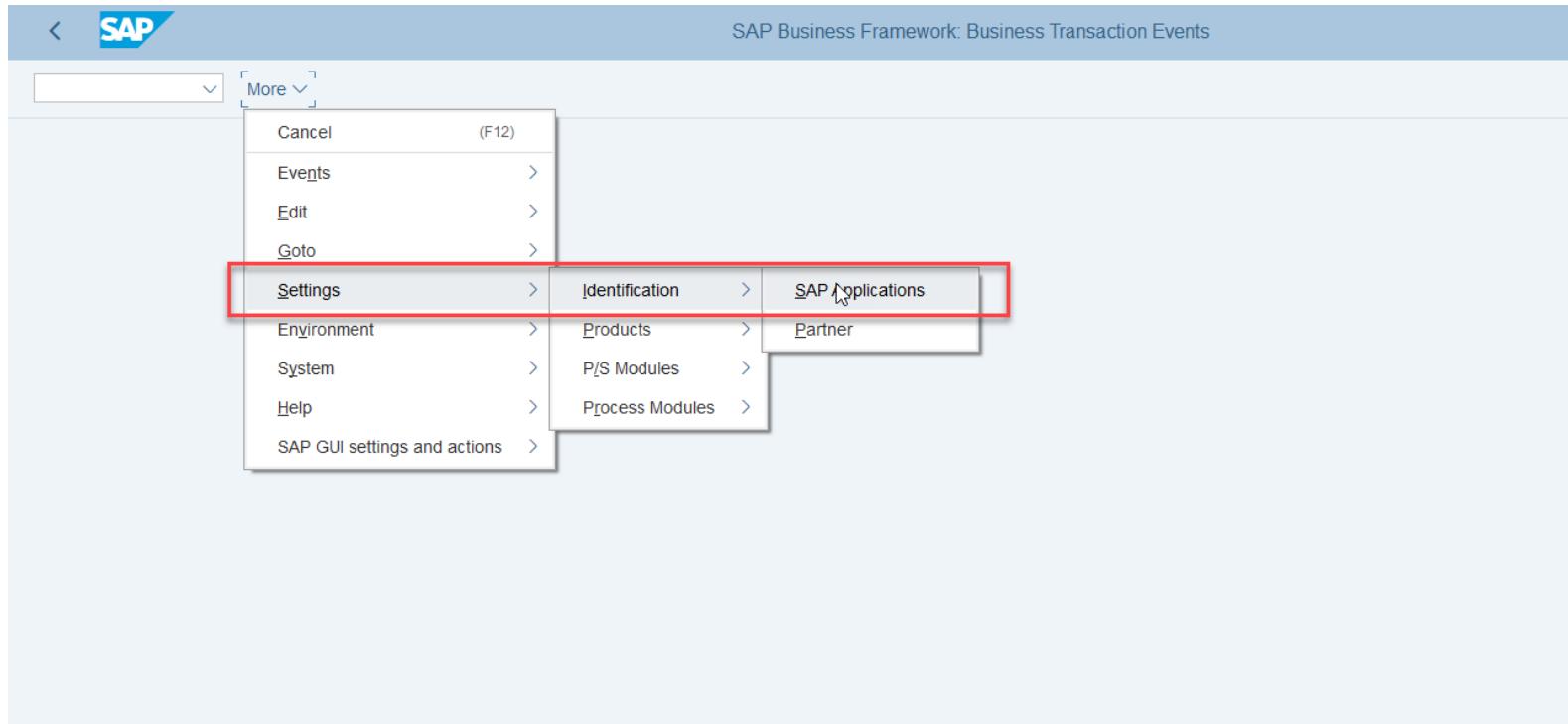
STEP 1: Check the SAP Product Version

1-1: Make sure that you have met the requirements for the product version mentioned in the “the “[Prerequisites](#)” section of *Appendix one: Connect to SAP ERP in Administration Guide for Version 2*. 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**

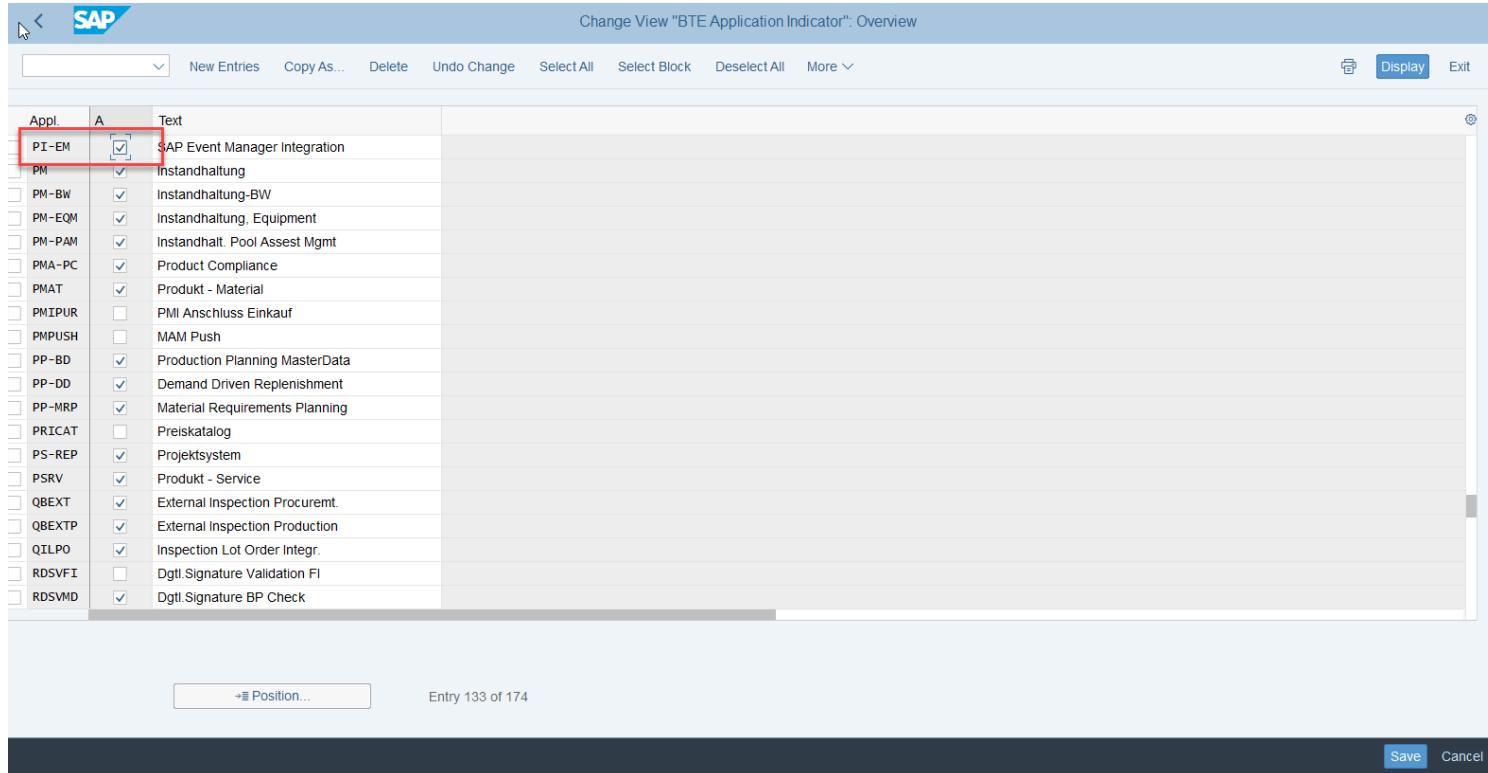


STEP 2: Activate SAP Event Manager Integration

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

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



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

1-1: Log on to the business client

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

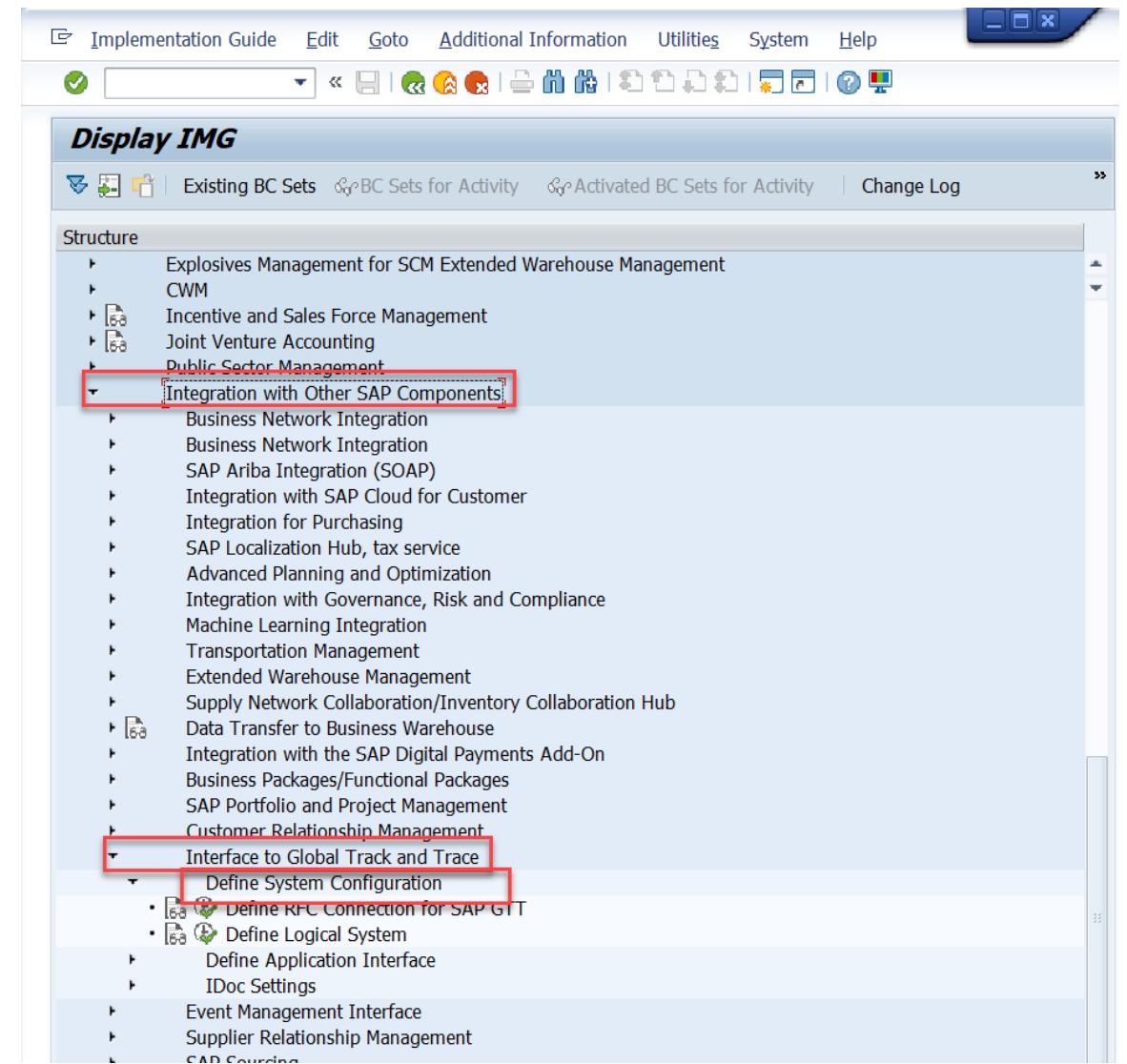
1-3: Click **Integration with Other SAP Components**

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

-> **Define System Configuration**

1-4: Choose activity:

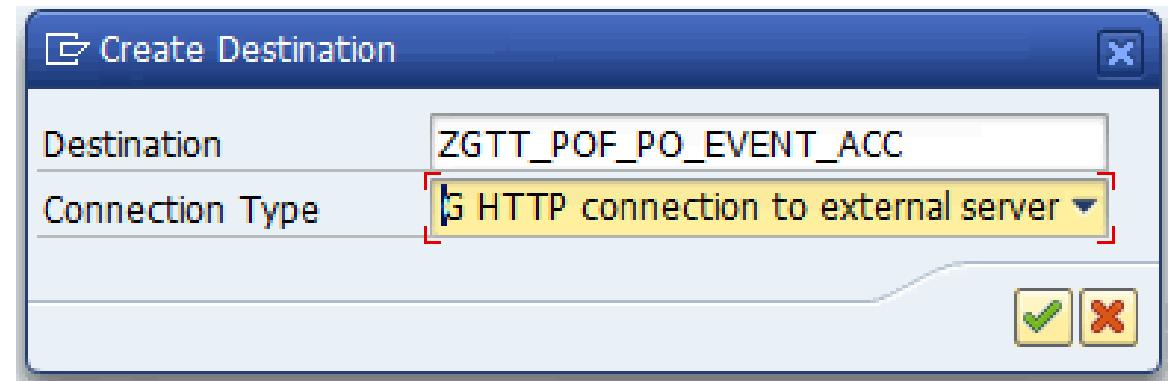
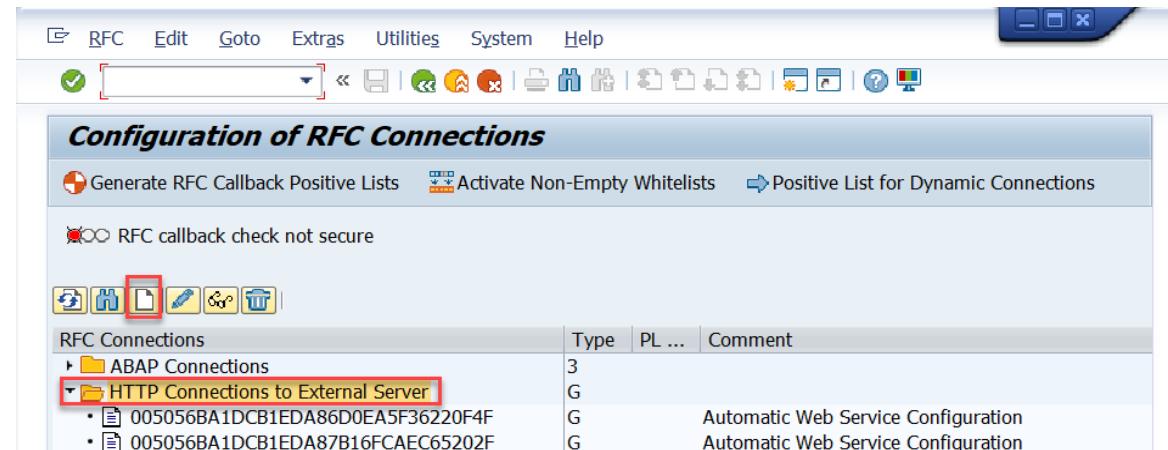
Define RFC Connection for SAP GTT



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

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

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



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

1-7: Enter a description

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

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

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

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

Port: 443

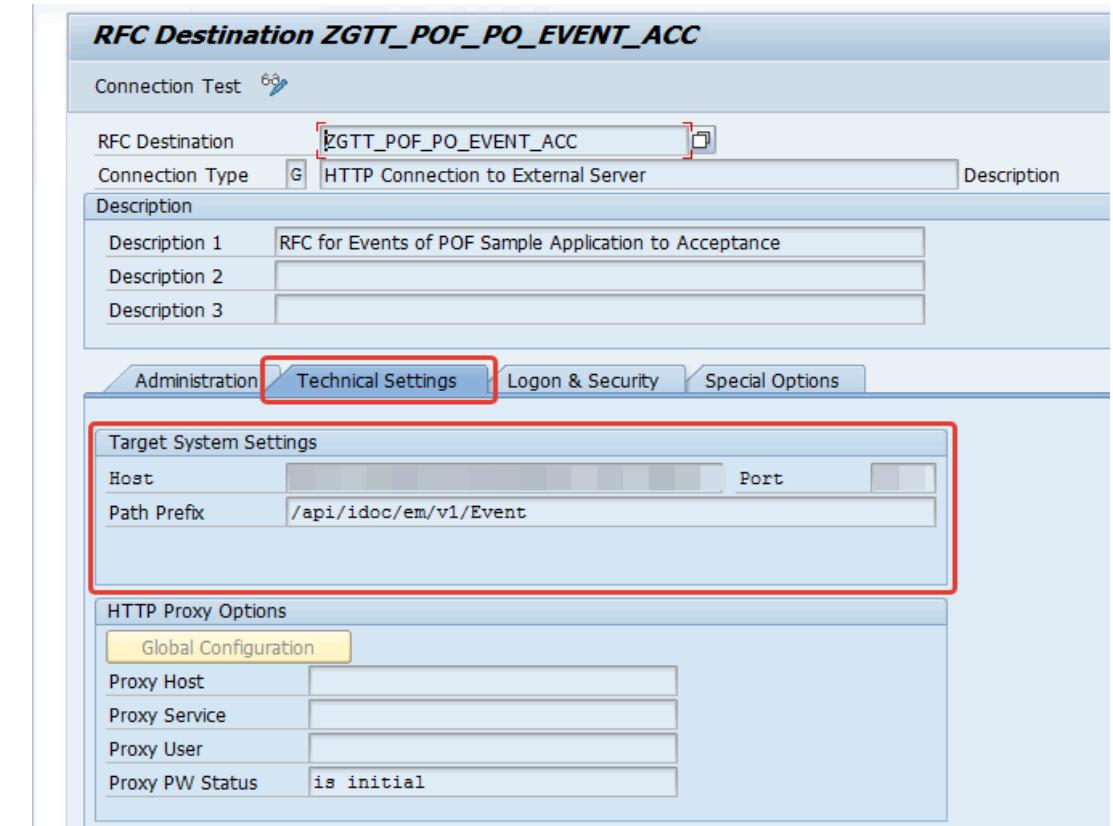
You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

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.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.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.

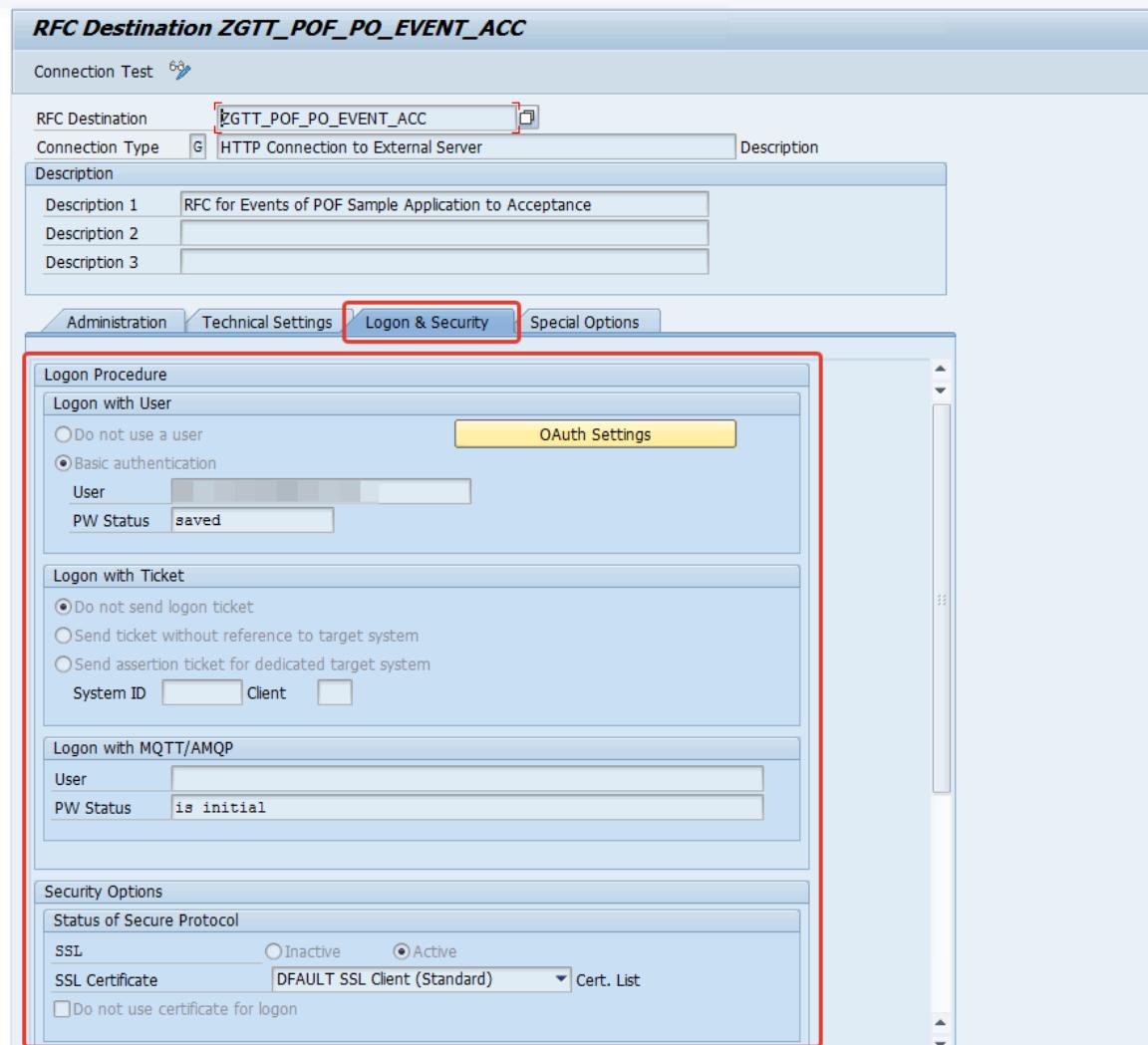
Also, SSL must be *Active*.

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

1-10: Save the configuration

Caution: You need to configure two RFC Connections:

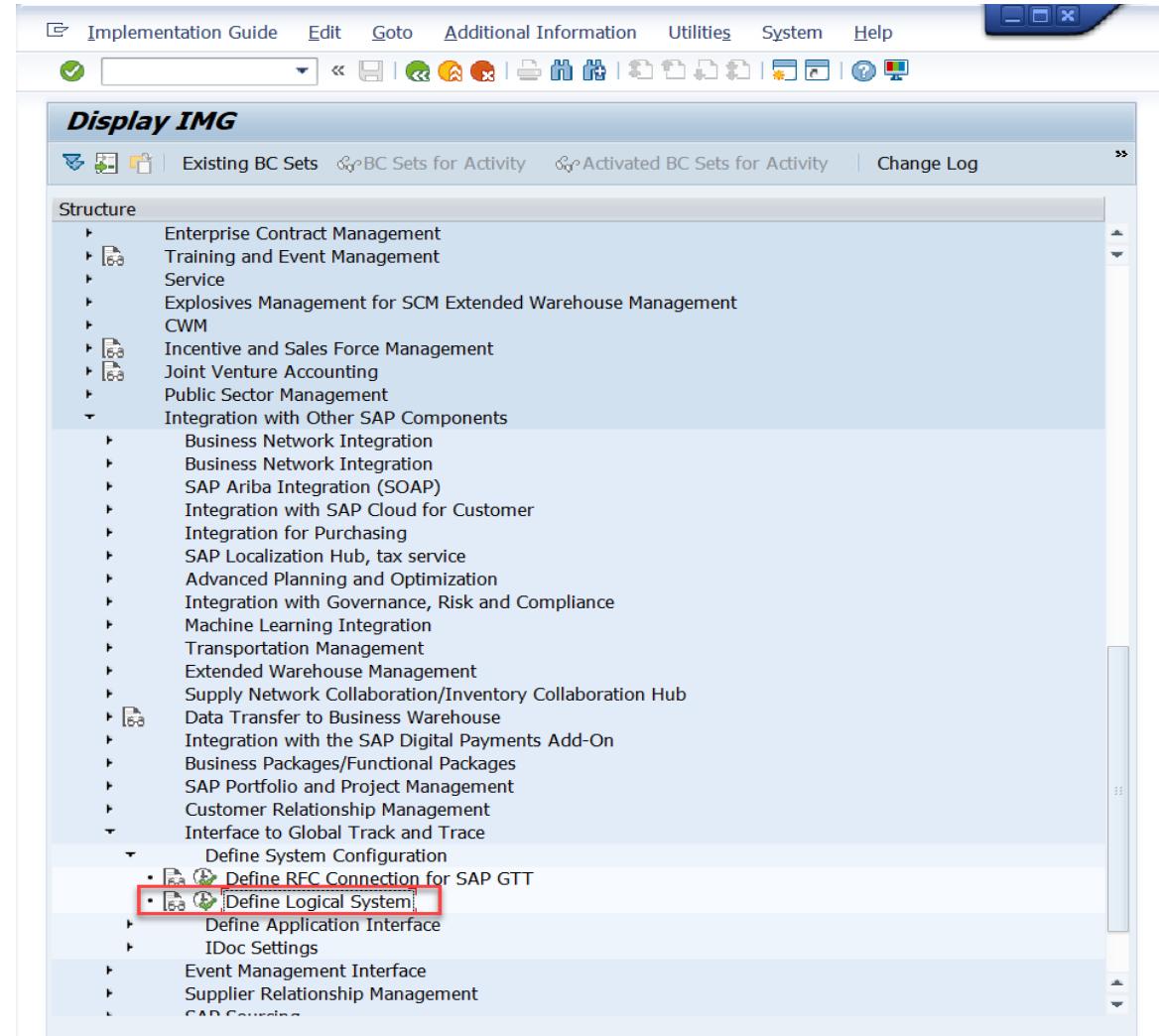
- one for event and
- the other for tracked Process.



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**

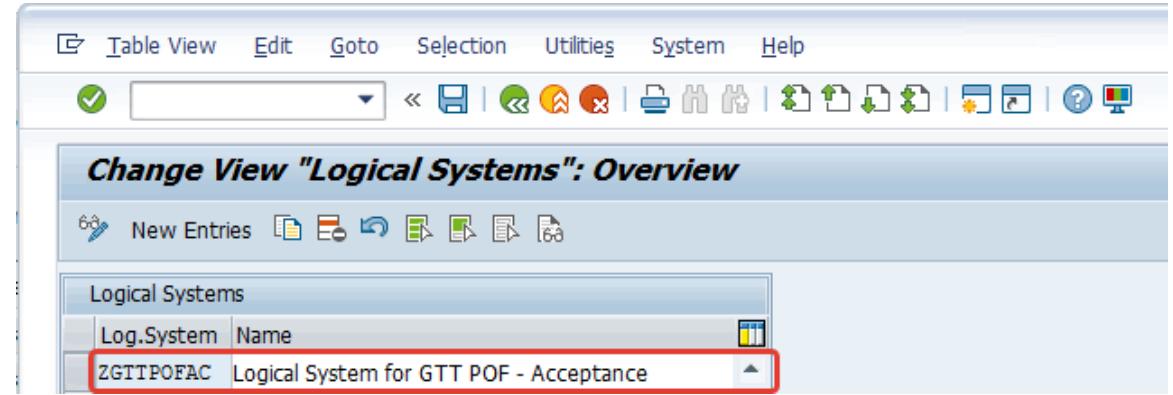


STEP 2: Define Logical System

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



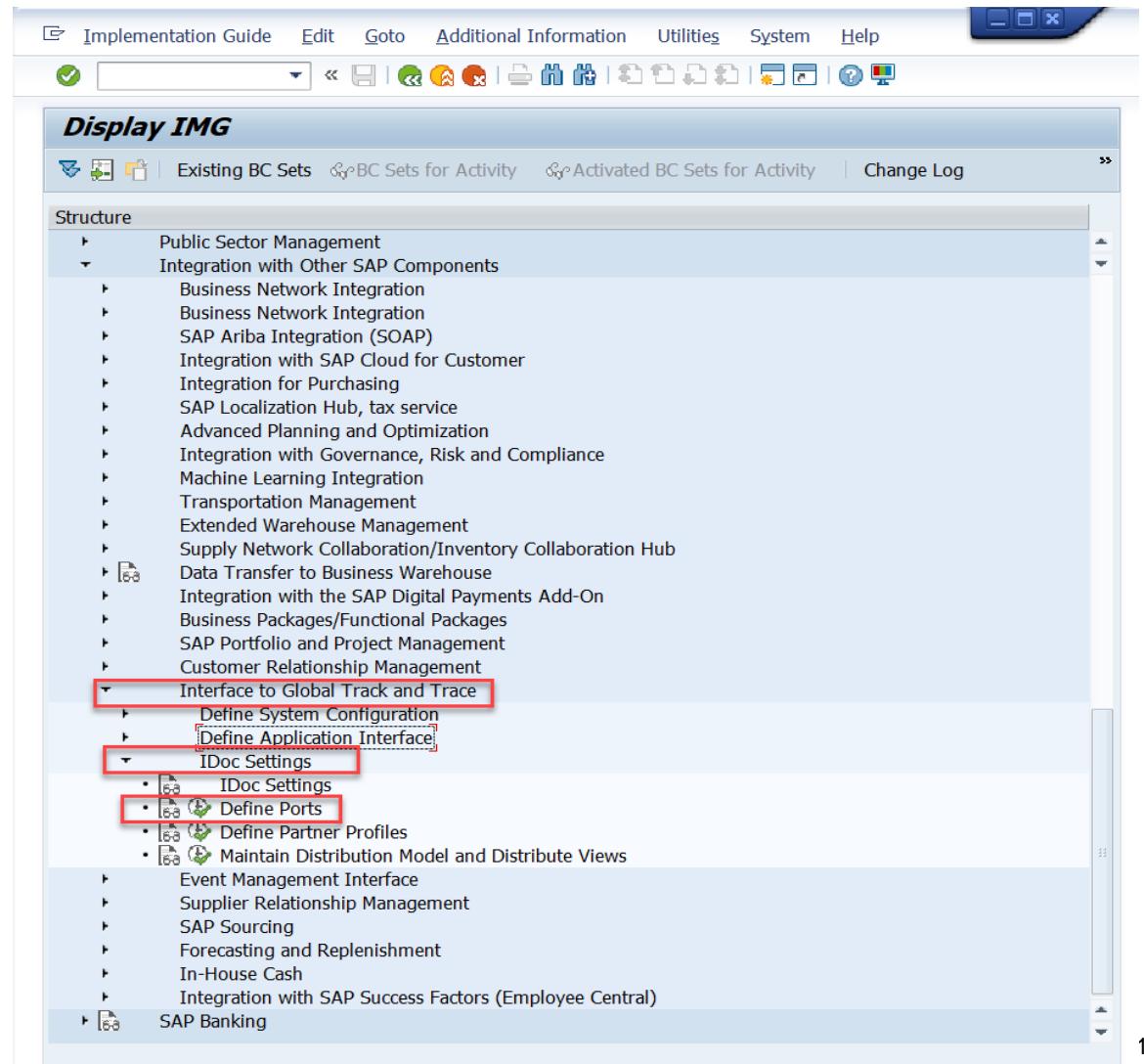
The screenshot shows the SAP GUI interface for defining logical systems. The title bar reads "Change View 'Logical Systems': Overview". Below the toolbar, there is a menu bar with "Table View", "Edit", "Goto", "Selection", "Utilities", "System", and "Help". The main area displays a table titled "Logical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

| 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

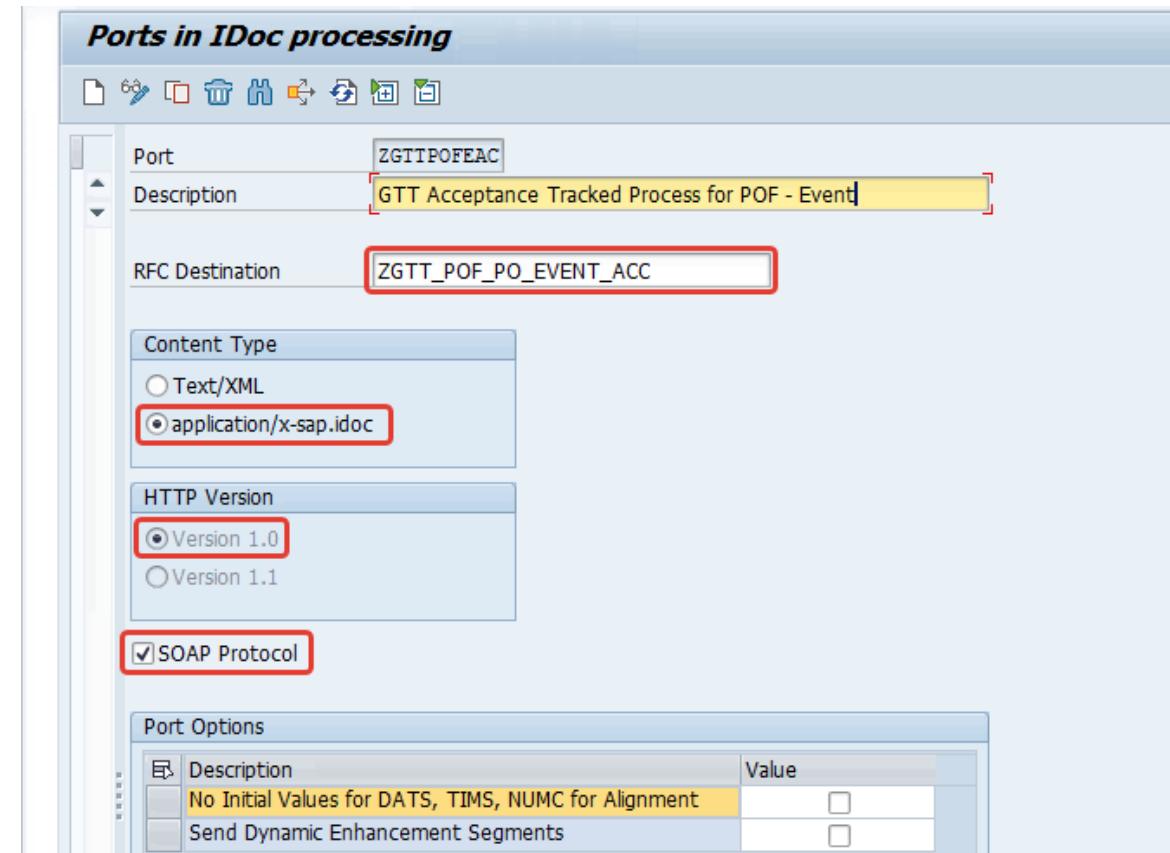
3-5: Choose **Content Type** as *application/x-sap.idoc*

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.

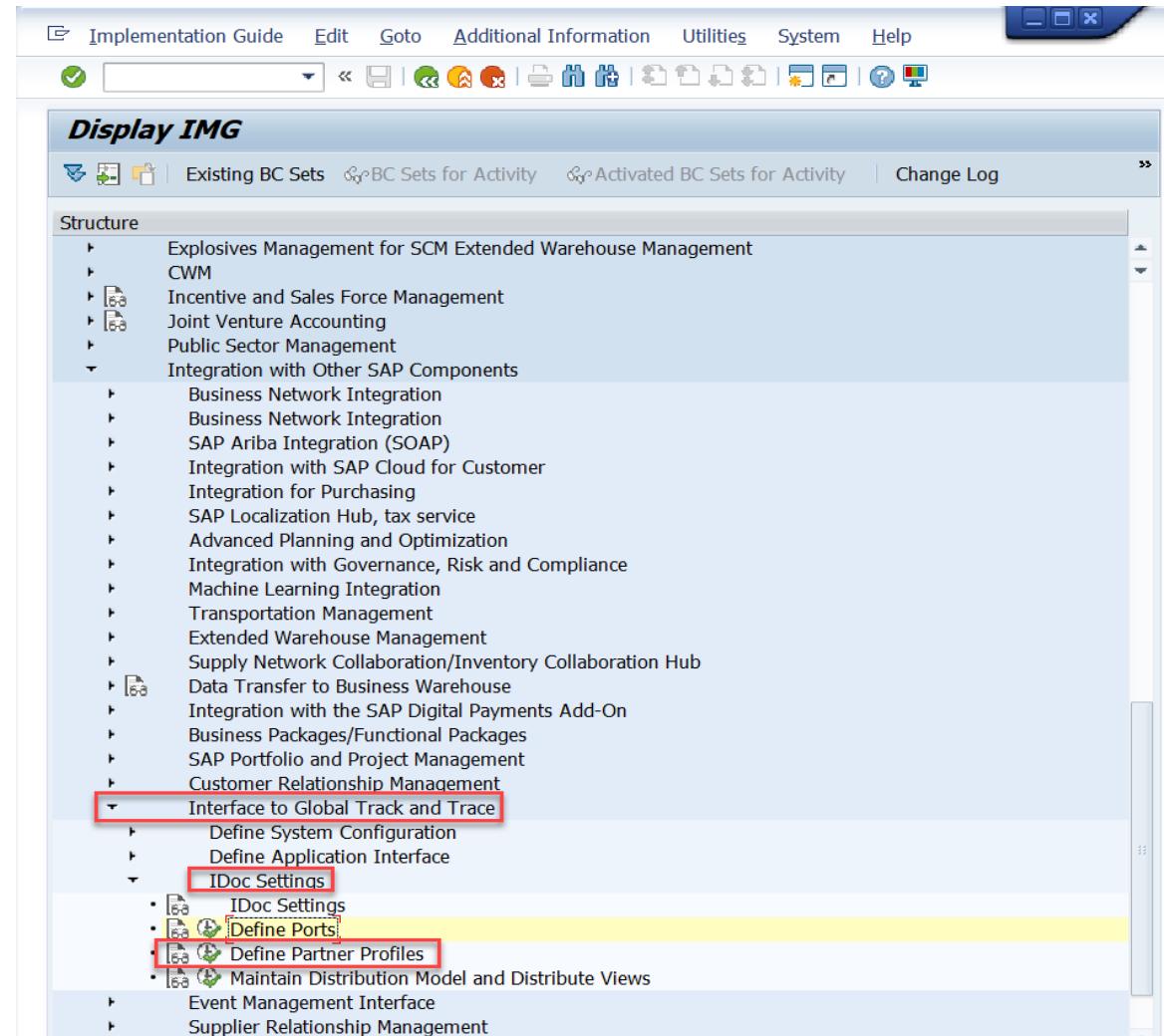


| Port | Description | RFC Destination | Content Type | HTTP Version | SOAP Protocol |
|------------|--|-----------------------|------------------------|--------------|---------------|
| ZGTTPOFEAC | GTT Acceptance Tracked Process for POF - Event | ZGTT_POF_PO_EVENT_ACC | application/x-sap.idoc | Version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

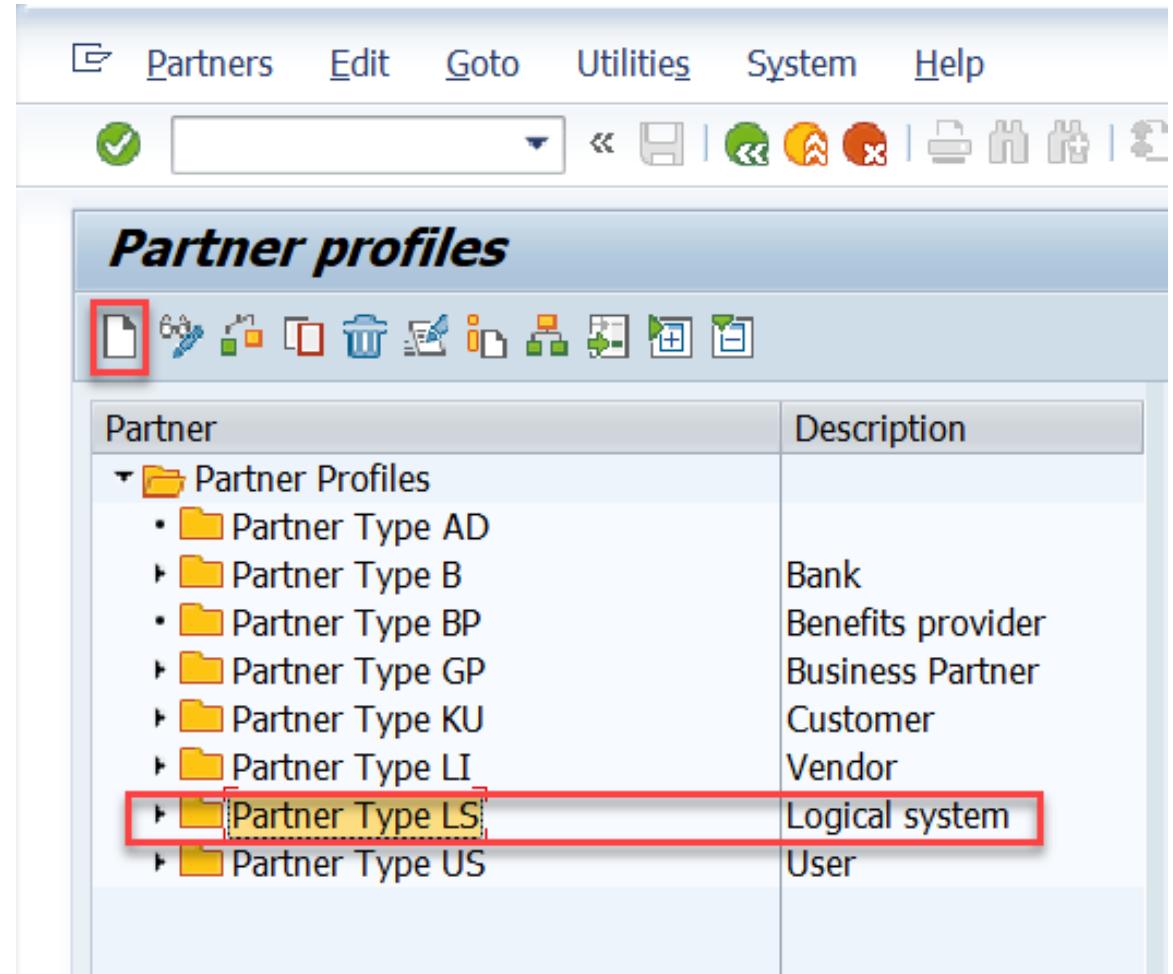
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**

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



STEP 4: Define Partner Profiles

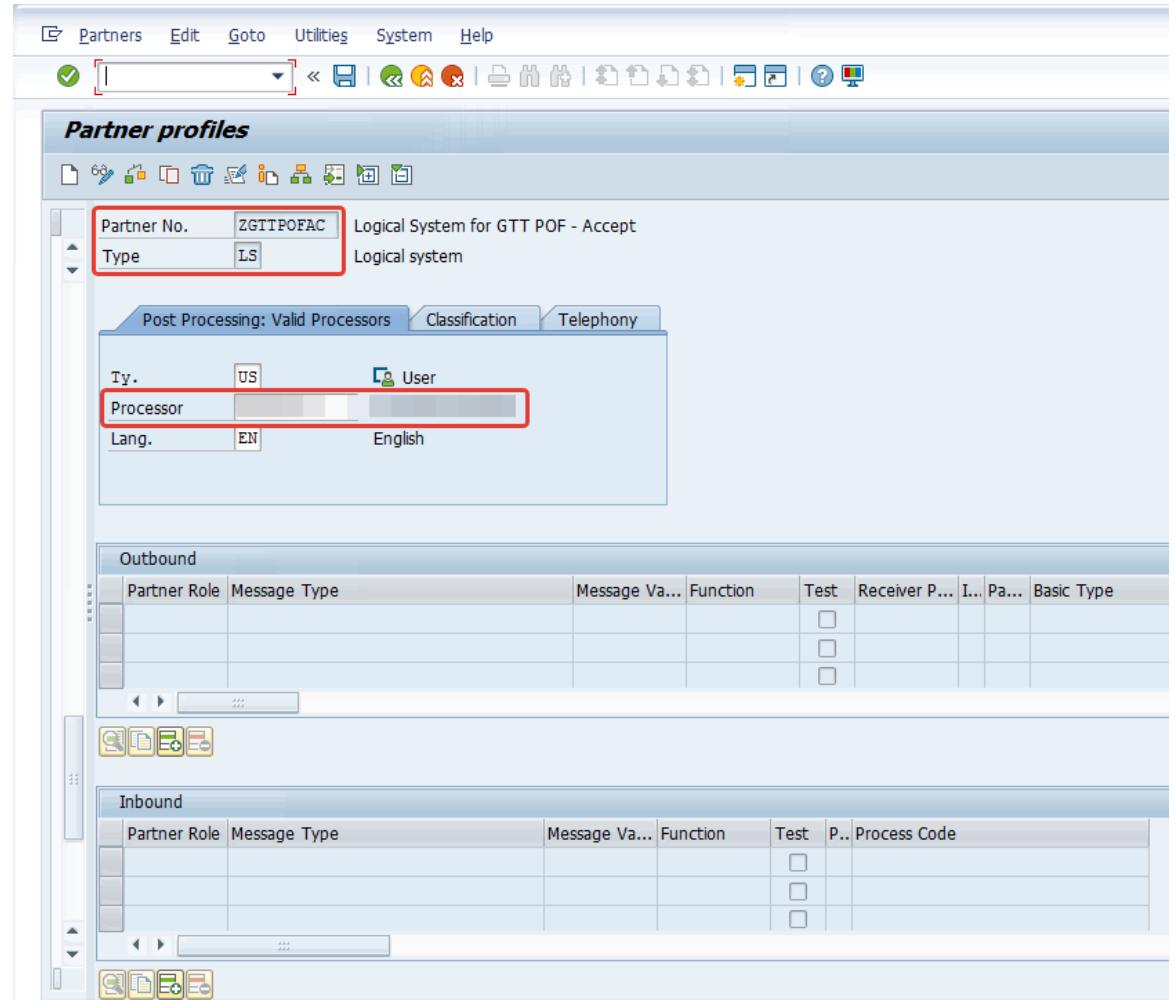
4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

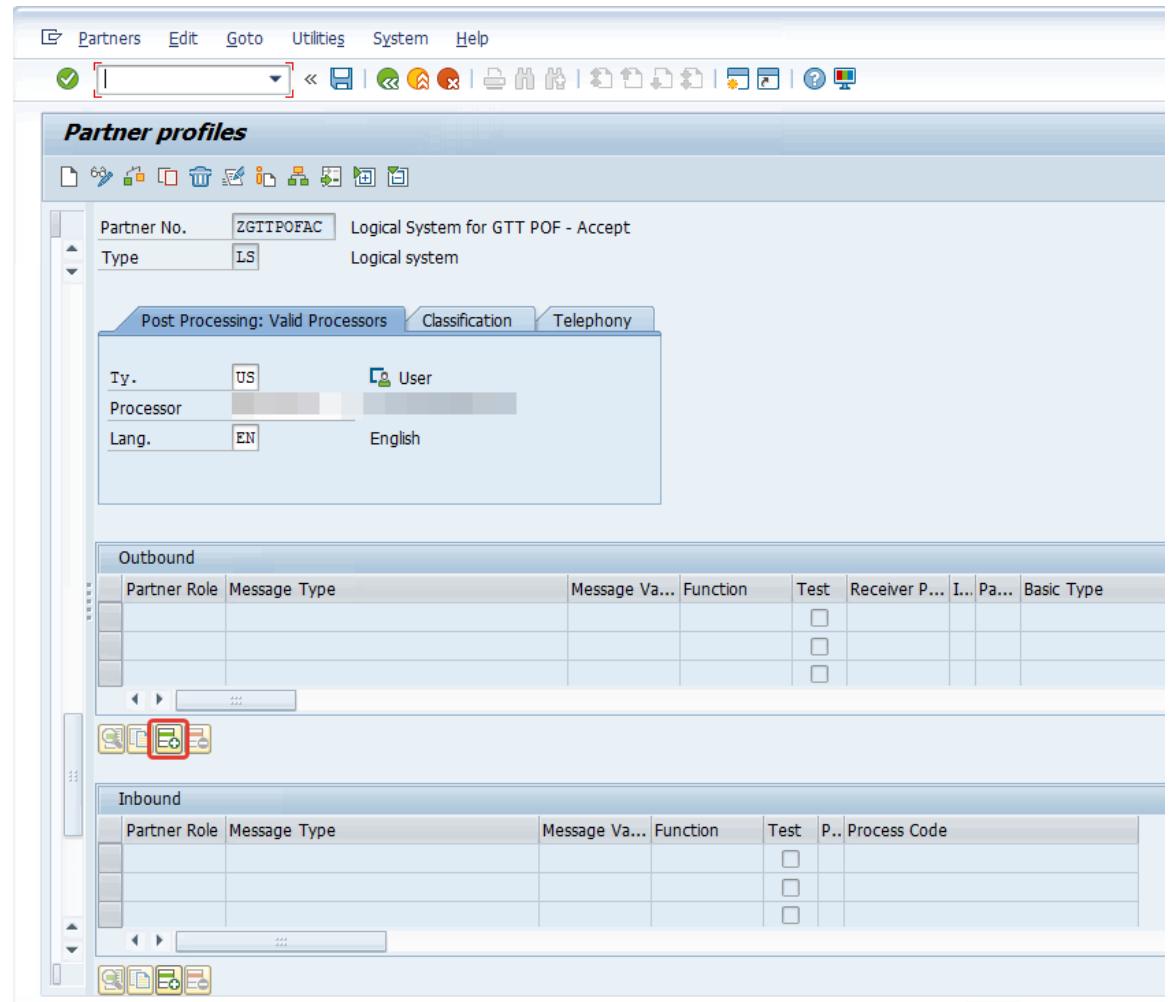
4-4: Fill in the **Partner No.** that you created in STEP 2

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

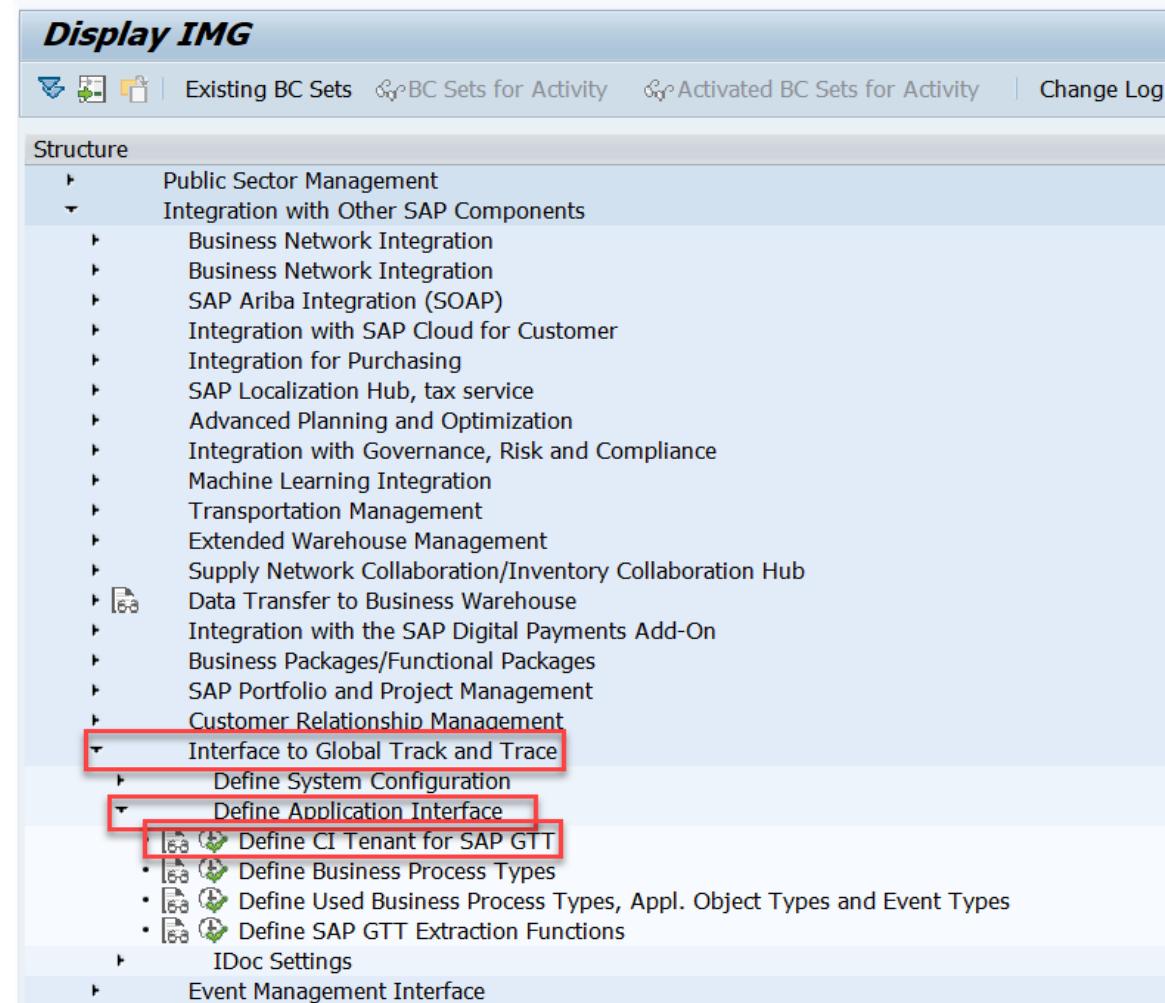
B2. Extractor Configuration



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

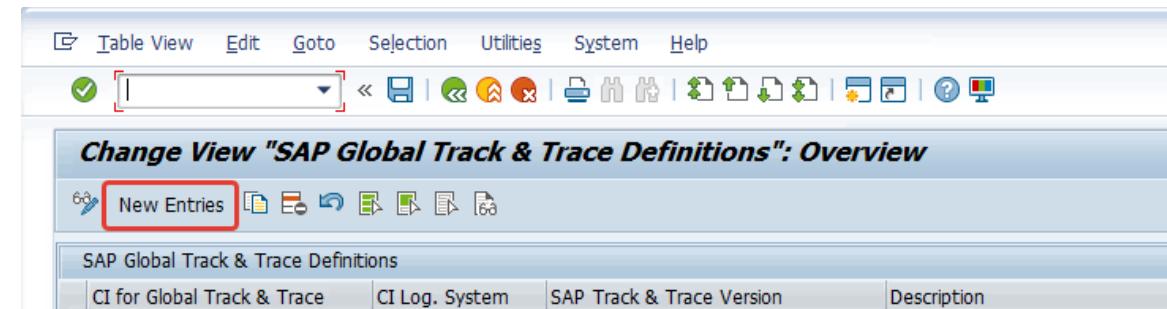
5-2: Choose activity
Define CI Tenant for SAP GTT



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.



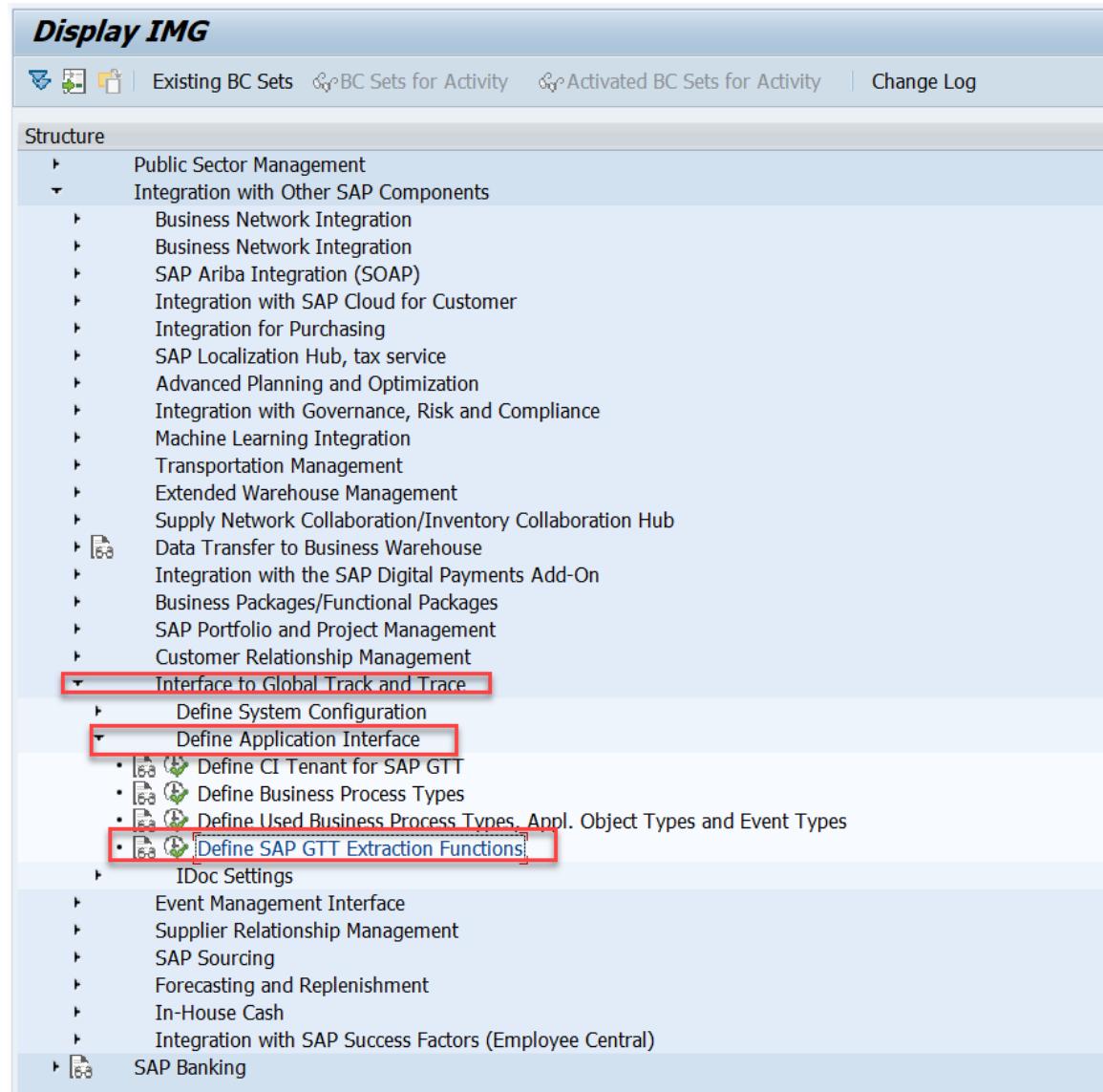
The screenshot shows the SAP Global Track & Trace Definitions overview screen after a new entry has been created. The top navigation bar and toolbar are identical to the previous screenshot. The main title is 'Change View "SAP Global Track & Trace Definitions": Overview'. The 'New Entries' button is visible in the toolbar. The table below now contains one row, which is highlighted with a red box. The row details a new CI tenant: '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

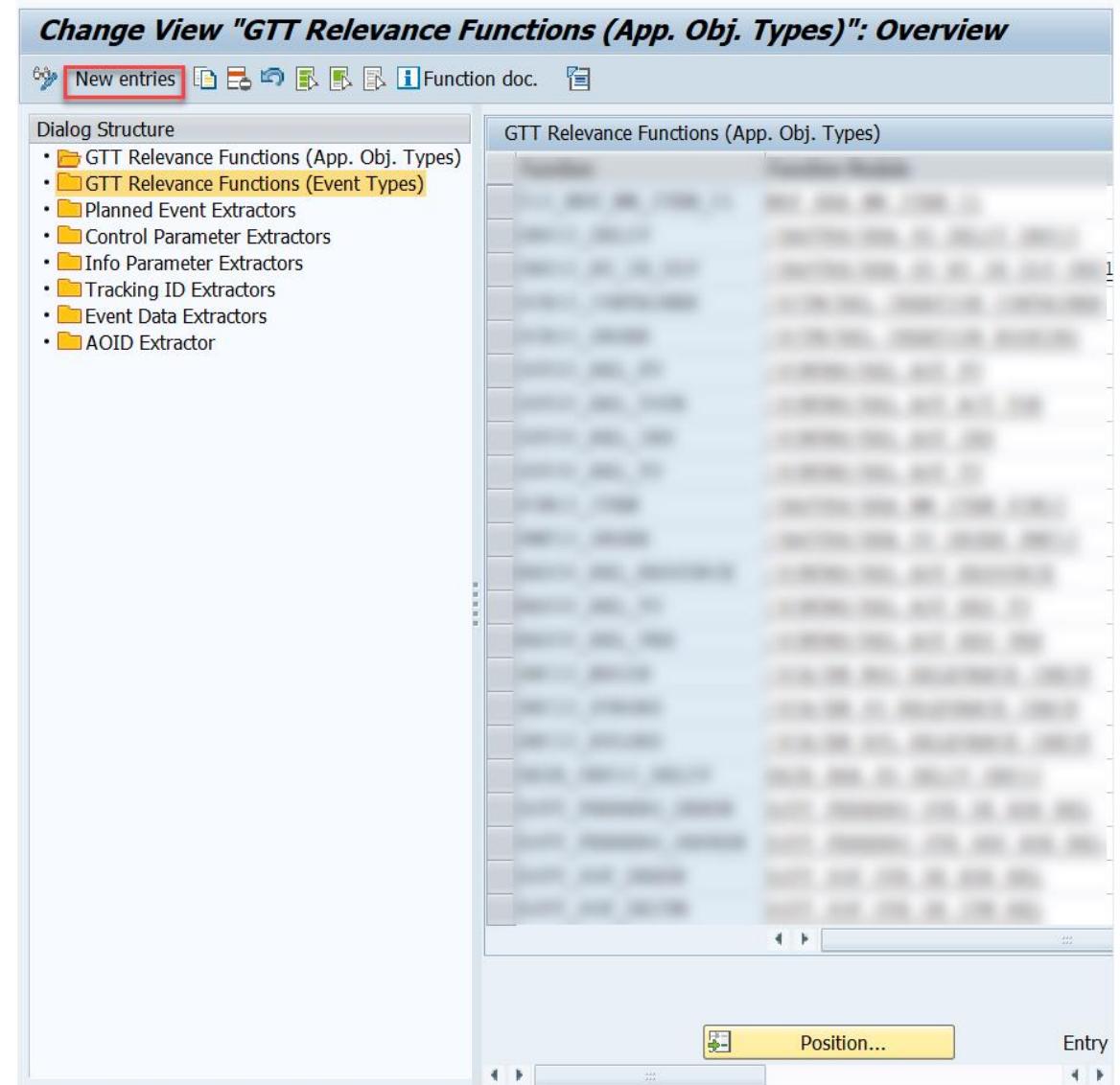


The screenshot shows the SAP Display IMG interface. The top navigation bar includes icons for search, refresh, and help, followed by links for 'Existing BC Sets', 'BC Sets for Activity', 'Activated BC Sets for Activity', and 'Change Log'. The main area is titled 'Structure' and contains a hierarchical list of SAP components and their sub-components. A red box highlights the path: 'Interface to Global Track and Trace' > 'Define Application Interface' > 'Define SAP GTT Extraction Functions'. The 'Define SAP GTT Extraction Functions' node is specifically highlighted with a red border.

- 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

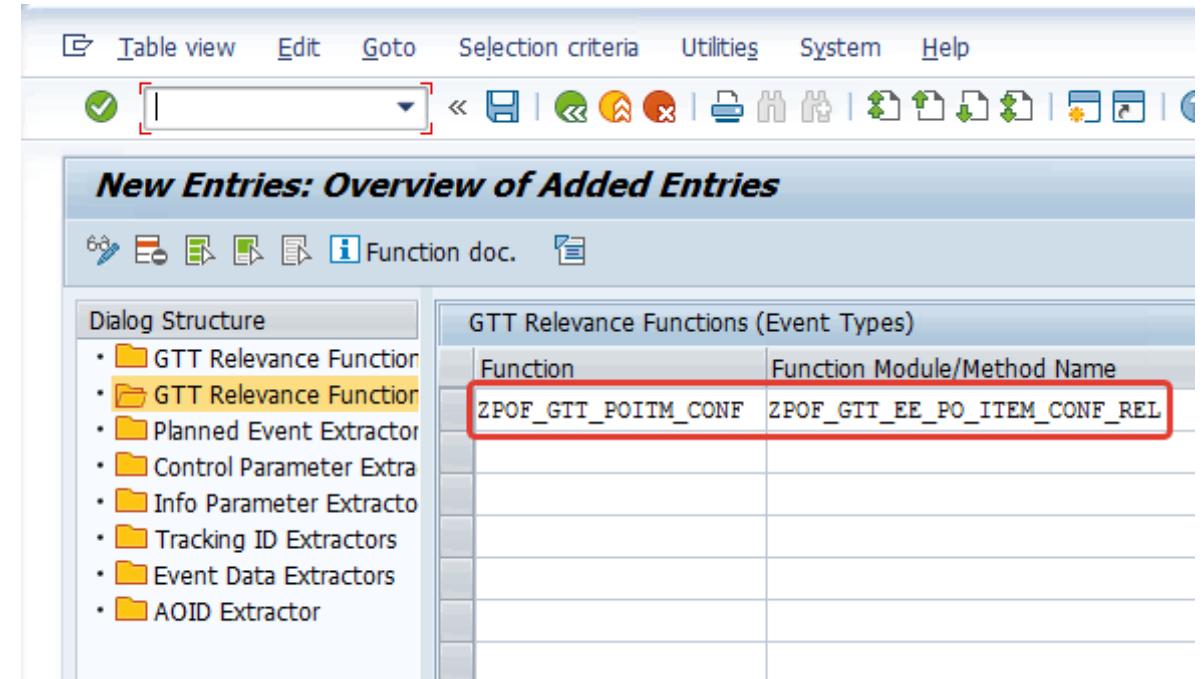
STEP 6: Define GTT Extraction Functions

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

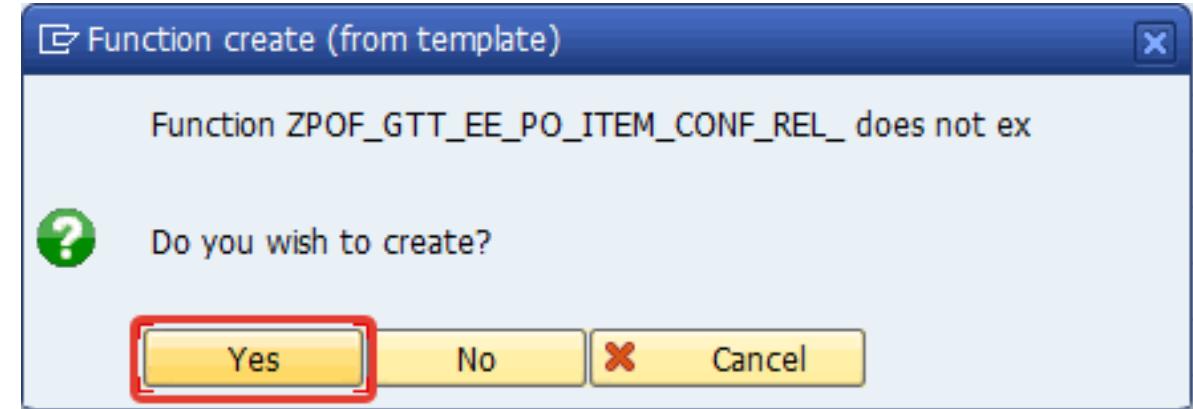
6-5: Click Save

New Entries: Overview of Added Entries

| Function | Function Module/Method Name |
|---------------------|------------------------------|
| ZPOF_GTT_POITM_CONF | ZPOF_GTT_EE_PO_ITEM_CONF_REL |

STEP 6: Define GTT Extraction Functions

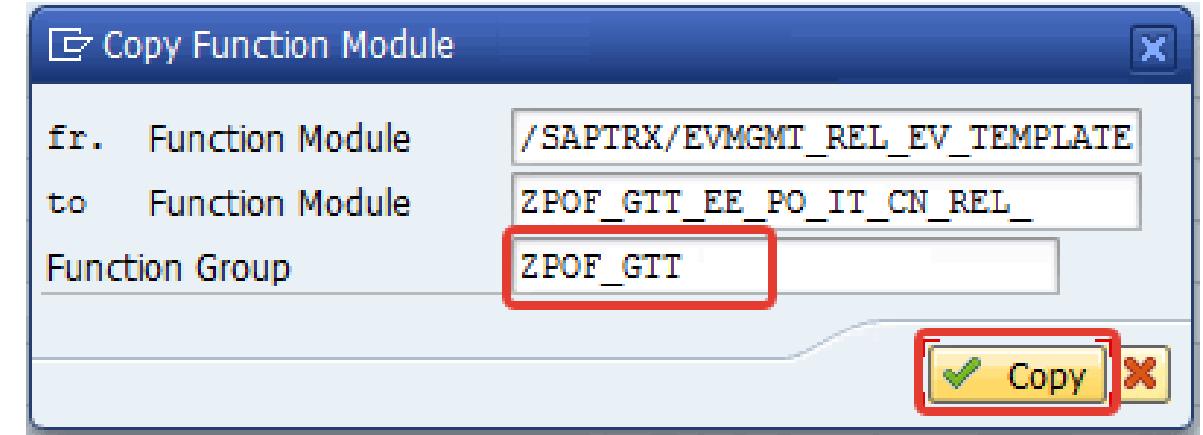
6-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



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.

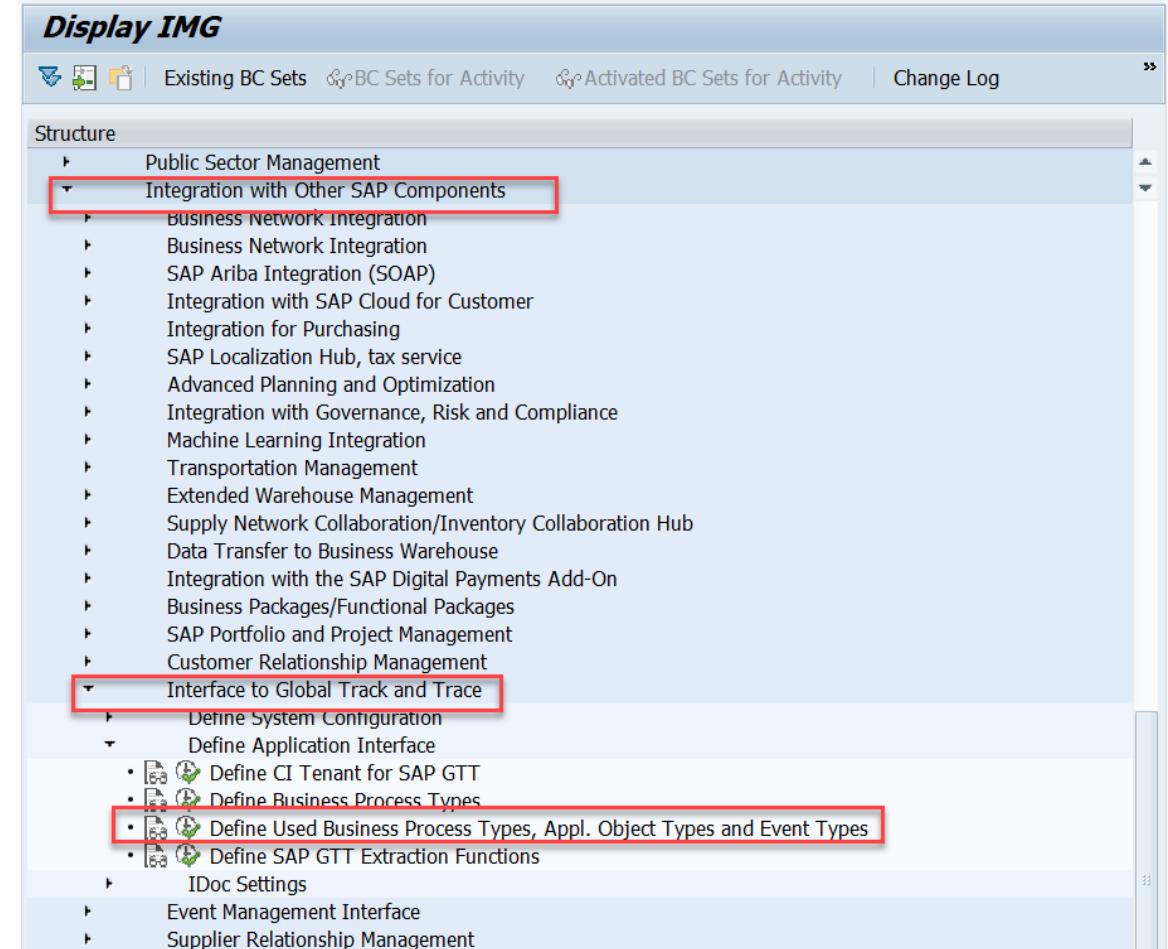
The screenshot shows the SAP SE80 Function Builder interface. The title bar reads "Function Builder: Display ZPOF_GTT_EE_PO_IT_CN_REL_". The top menu includes "Function Module", "Edit", "Goto", "Utilities", "Environment", "System", and "Help". The toolbar has various icons for file operations like Open, Save, Print, and Copy. The left pane is a "Repository Browser" with a "Function Group" dropdown set to "ZPOF_GTT" and a list of function modules under "Object Name". One item, "ZPOF_GTT_EE_PO_IT_CN_REL_", is highlighted with a red box. The right pane displays the source code for the selected function module:

```
1 FUNCTION ZPOF_GTT_EE_PO_IT_CN_REL_.
2   *--> Local Interface:
3   *--> IMPORTING
4   *-->   REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM
5   *-->   REFERENCE(I_EVENT_TYPES) TYPE /SAPTRX/EVTYPES
6   *-->   REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER
7   *-->   REFERENCE(I_EVENTTYPE_TAB) TYPE TRXAS_EVENTTYPE_TABS_WA
8   *-->   REFERENCE(I_EVENT) TYPE TRXAS_EVT_CTAB_WA
9   *--> EXPORTING
10  *-->   VALUE(E_RESULT) LIKE SY-BINPT
11  *--> TABLES
12  *-->   C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL
13  *--> EXCEPTIONS
14  *-->   PARAMETER_ERROR
15  *-->   RELEVANCE_DETERM_ERROR
16  *-->   STOP_PROCESSING
17  *-->
18  *-->*
19  *-->* Top Include
20  *-->* TYPE-POOLS:trxas.
21  *-->
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 | | |
|--|--------------------|--------------------------|
| New Entries | | |
| Dialog Structure | | |
| • Define Used Business Process Types | Bus. Proc. Type | Update Mode |
| • Define Application Object Types | EPL_NOTIF | Update Task (▼ Active |
| • Define Event Types | ESC_DELIV | Update Task ... ▼ Active |
| | ESC_FI_CLEARING | Update Task ... ▼ Active |
| | ESC_MATDOC | Update Task ... ▼ Active |
| | ESC_MM_INVOICE | Update Task ... ▼ Active |
| | ESC_PURORD | Update Task ... ▼ Active |
| | ESC_PURORD_FASHION | Update Task ... ▼ Active |
| | ESC_SHIPMT | Update Task ... ▼ Active |
| | ESC_SORDER | Update Task ... ▼ Active |
| | ESC_WRKORD | Update Task ... ▼ Active |
| | OCB10_ORDER | Dialog Update ▼ Active |
| | SNC_MSGIN | Dialog Update ▼ Active |
| | SNC_PURORD | Dialog Update ▼ Active |
| | SNC_RPLORD | Dialog Update ▼ Active |
| | TMS_INS | Update Task ... ▼ Active |
| | TMS_RES | Update Task ... ▼ Active |
| | TMS_TOR | Update Task ... ▼ Active |

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

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

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

| 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 Fiori Change View "Define Event Types": Overview. The top navigation bar includes Table View, Edit, Goto, Selection, Utilities, System, and Help. Below the navigation is a toolbar with various icons. The main title is "Change View 'Define Event Types': Overview". On the left, there is a "Dialog Structure" tree with nodes for "Define Used Business Pro", "Define Application Ot", and "Define Event Types". A red box highlights the "New Entries" button in the toolbar. The main area displays a table titled "Define Event Types" with columns: Business Process Type, Event Type, and Description. The table contains three rows:

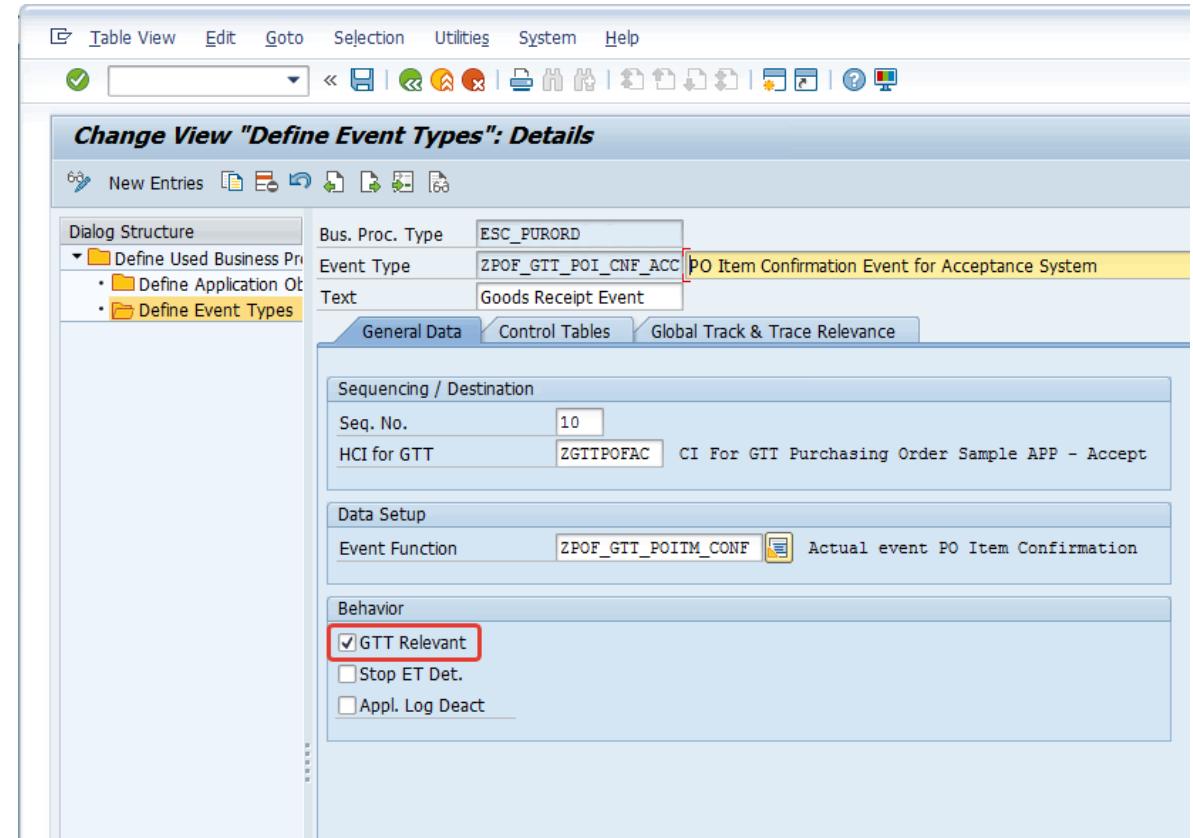
| Business Process Type | Event Type | Description |
|-----------------------|----------------------|---|
| ESC_PURORD | ZPOF_GTT_POI_DEL_ACC | PO Item Confirmation Event for Acceptance System |
| ESC_PURORD | ZPOF_GTT_POI_DEL_ACZ | PO Item Confirmation Event for Azure System |
| ESC_PURORD | ZPOF_GTT_POI_DEL_INT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

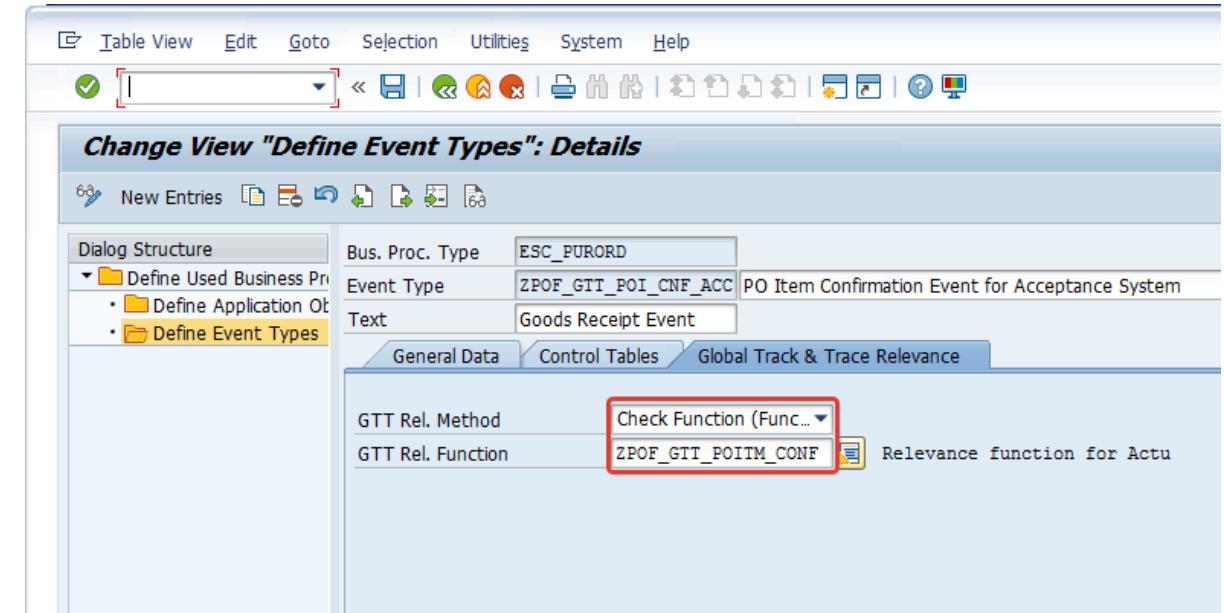
| | | |
|--|---------------------------|---|
| Bus. Proc. Type | ESC_PURORD | |
| Event Type | ZPOF_GTT_POT_CNF_ACC | PO Item Confirmation Event for Acceptance System |
| Text | Goods Receipt Event | |
| General Data Control Tables Global Track & Trace Relevance | | |
| Data Source for Events | | |
| 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"/> R |
| 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

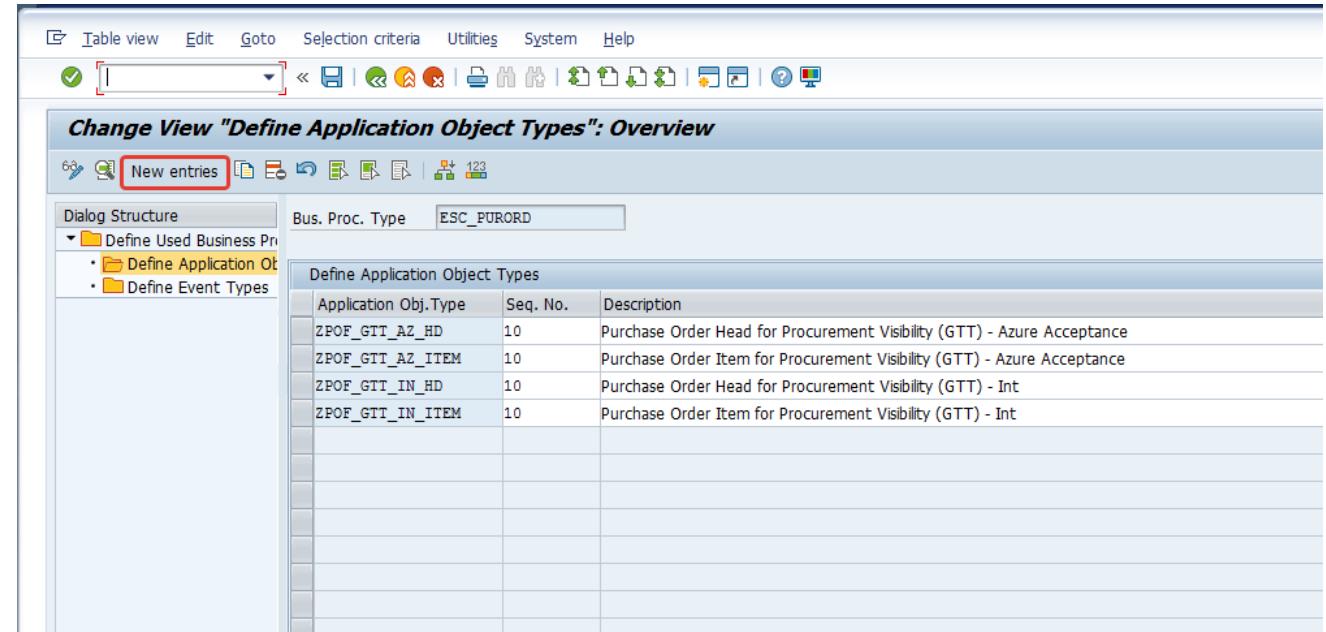
7-11: Choose the business process type from the **Define Used Business Process Types** on the right side

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**

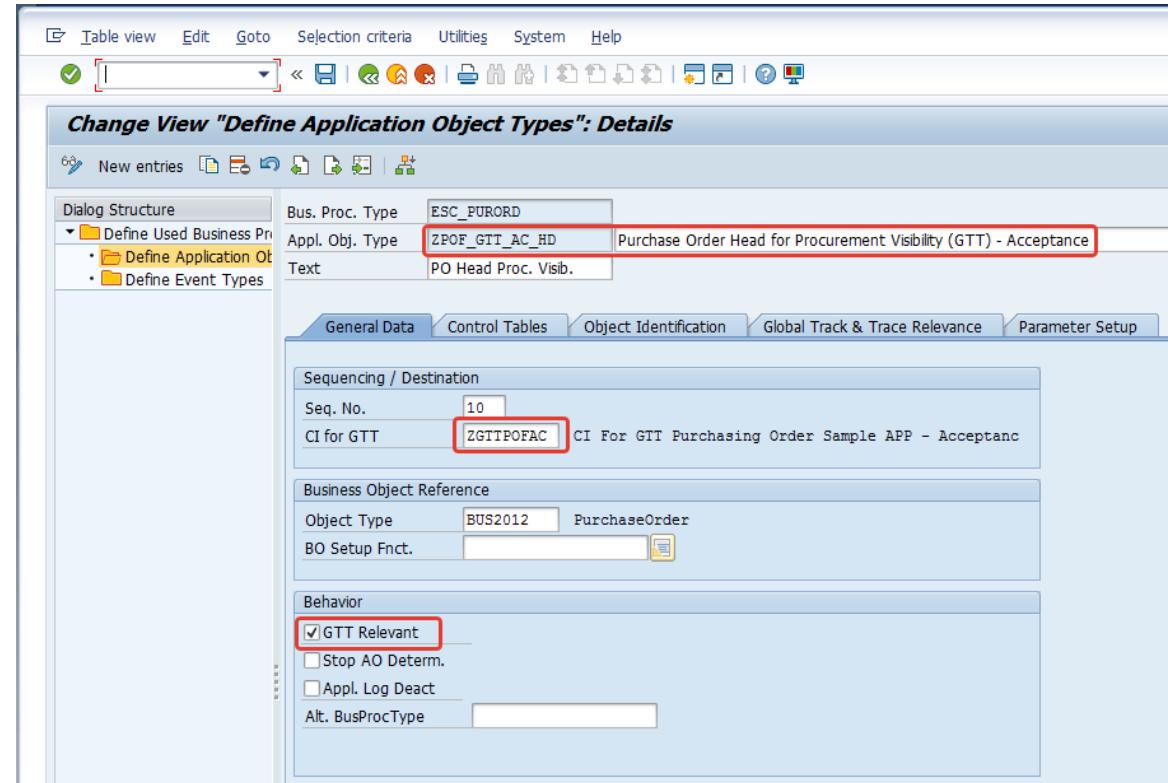


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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**

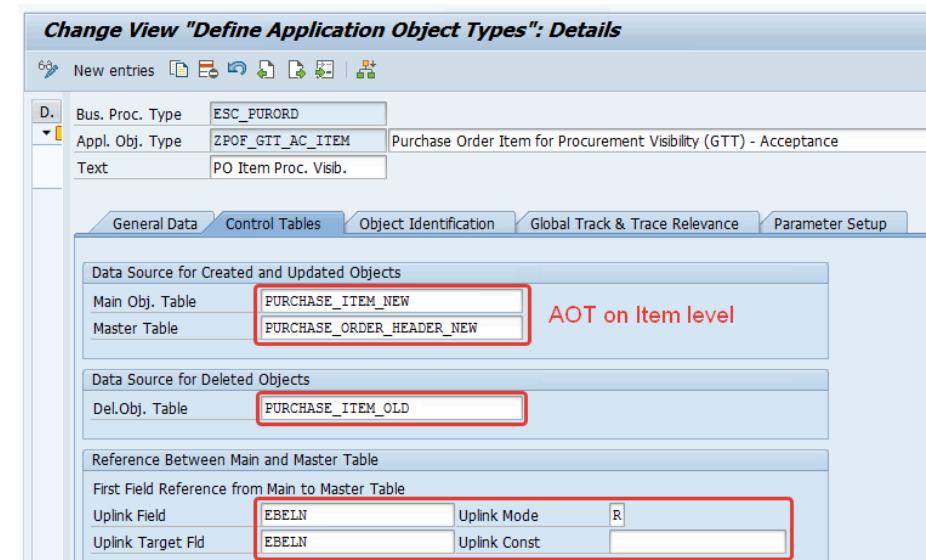
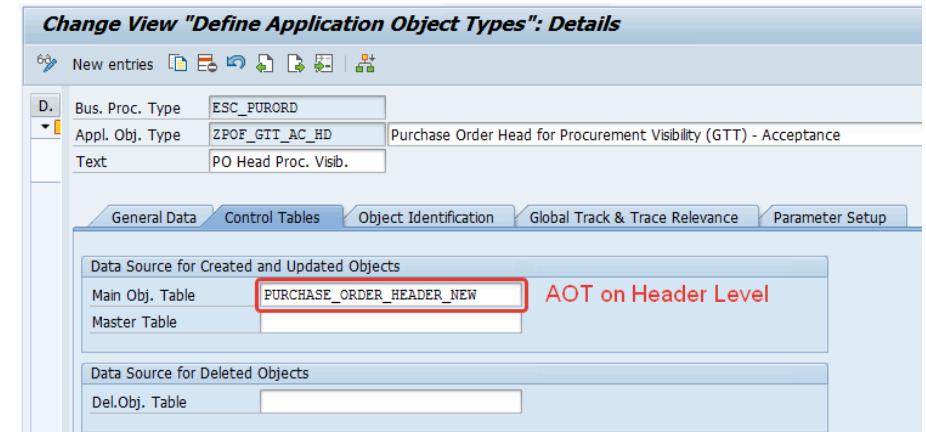


STEP 7: Define Used Business Process Types, Appl. Object Types and Event Types

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

Caution:

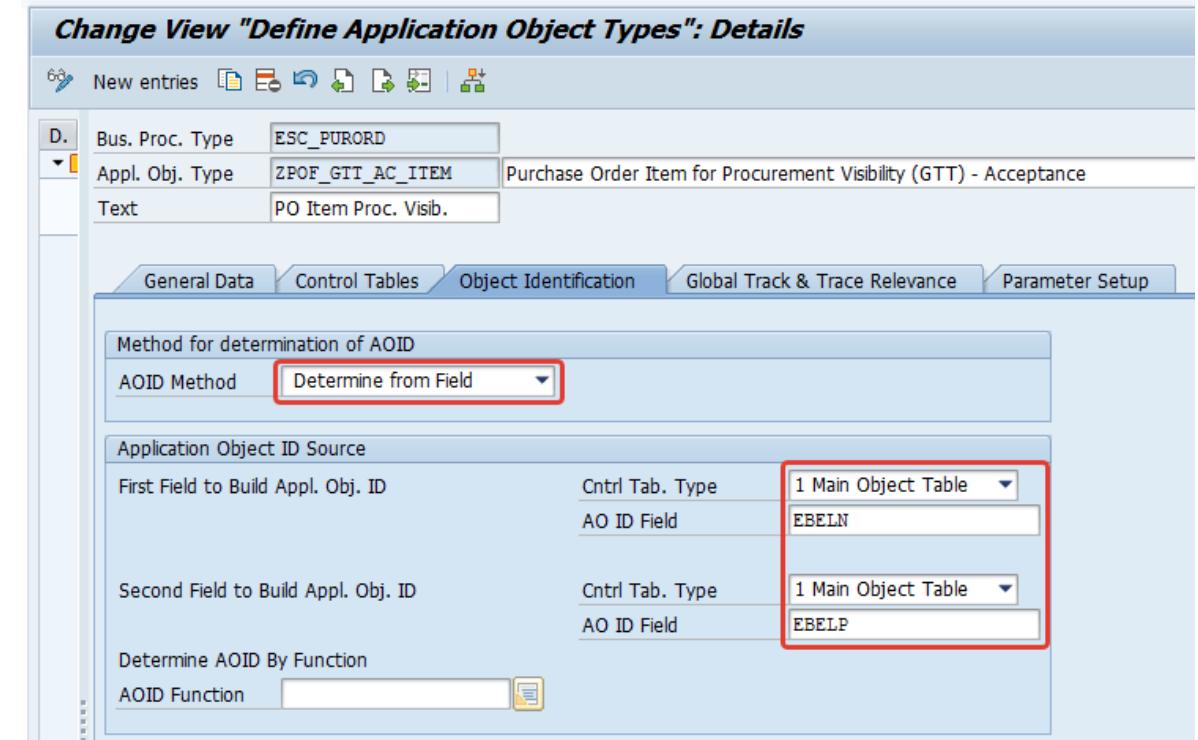
If the event type or application object type is on the header level, then you only need to assign the **Main Object Table**. Otherwise, if the event type or application object type is on the item level, then you need to assign the **Main Object Table** and **Master Table**, and assign the reference between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

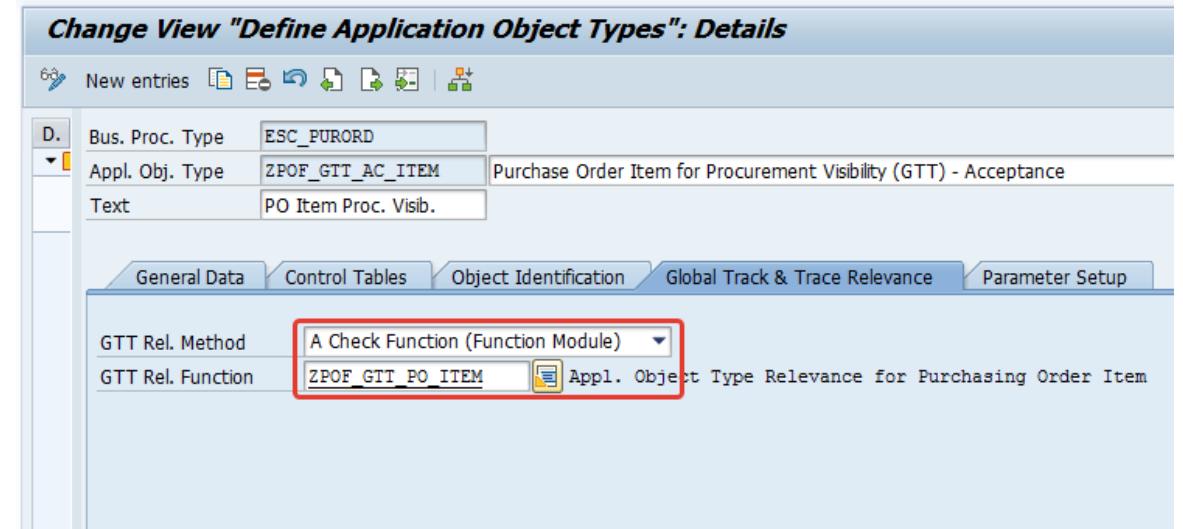
7-19: When choosing *Determine by Function*, you must enter the customized information in the AOID Function field.



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

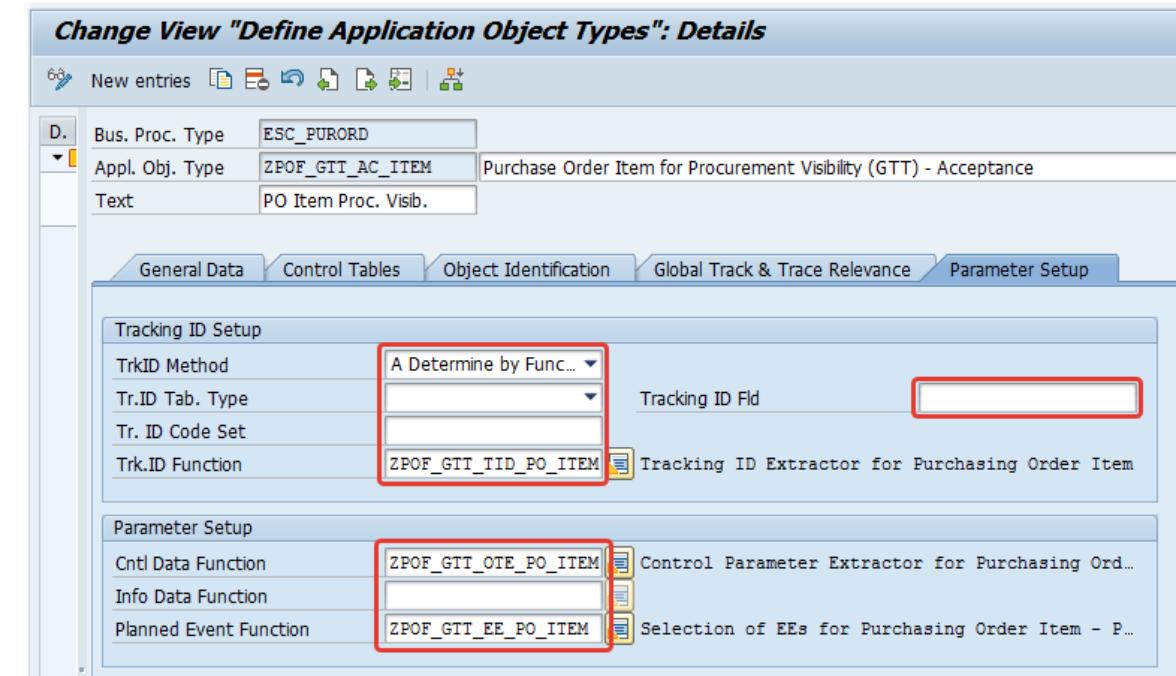
7-21: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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**.



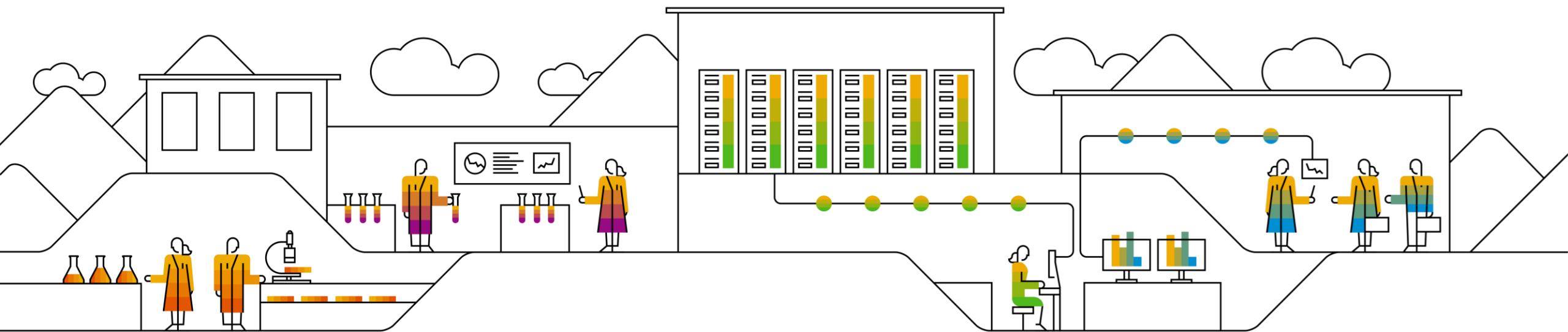
C) Download ABAP Code from GitHub

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

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

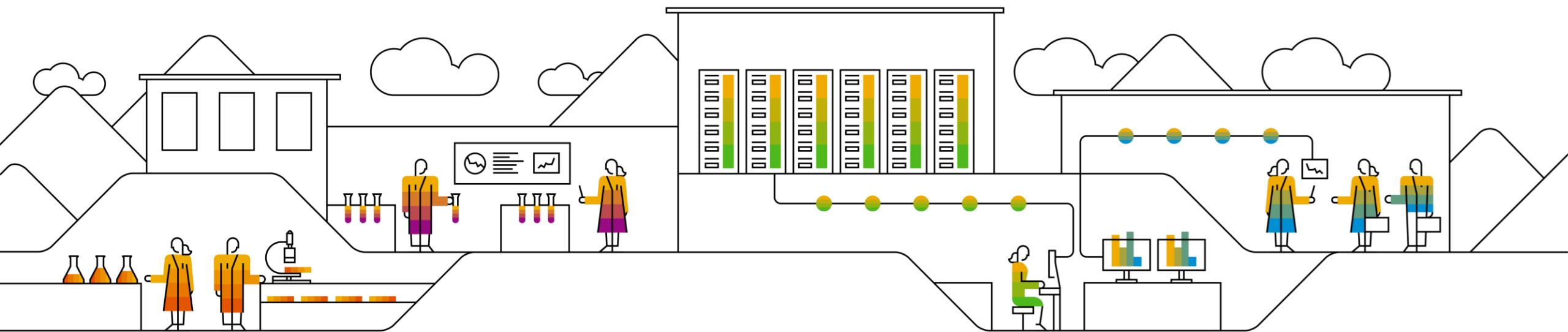
C3. Download Another ABAP Code from GitHub (TSOF)

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

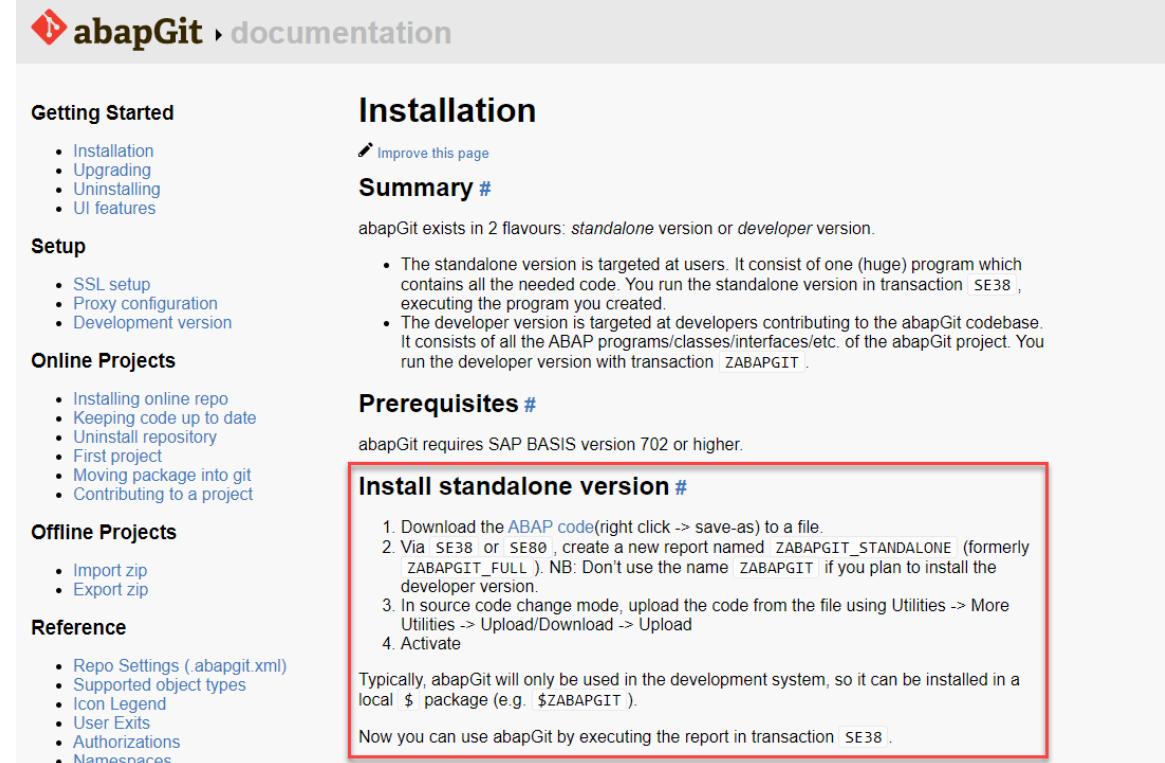
You need to install ABAPGit before downloading codes from GitHub.

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.

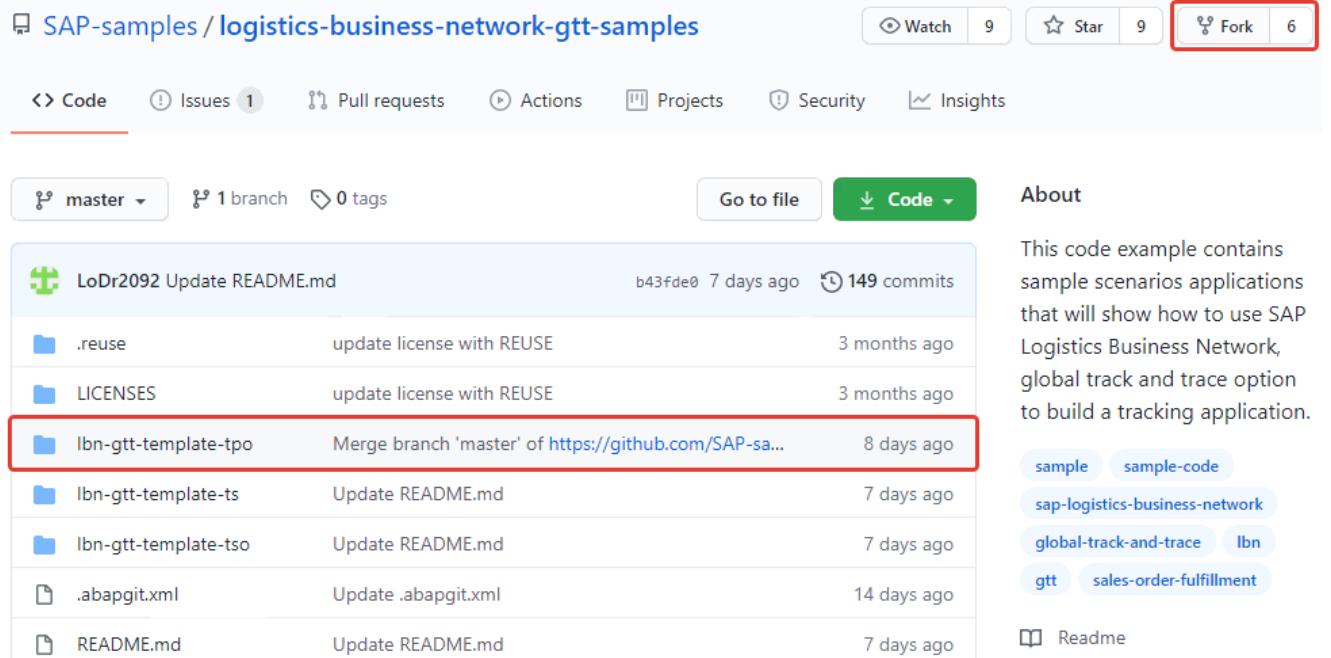


The screenshot shows the abapGit documentation page. The header reads "abapGit › documentation". On the left, there's a sidebar with links like "Getting Started", "Setup", "Online Projects", "Offline Projects", and "Reference". The main content area has a heading "Installation" with a link "Improve this page" and a "Summary" section. It explains that abapGit exists in two flavours: "standalone" and "developer". The "standalone" version is targeted at users and consists of one huge program, while the "developer" version is targeted at developers contributing to the codebase. Below this is a "Prerequisites" section stating "abapGit requires SAP BASIS version 702 or higher". A large red box highlights the "Install standalone version" section, which contains a numbered list of steps: 1. Download the ABAP code (right click -> save-as) to a file. 2. Via SE38 or SE80, create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version. 3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload. 4. Activate. Below this, it says "Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. \$ZABAPGIT)". Finally, it 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 the 'master' branch, showing '1 branch' and '0 tags'. To the right are 'Go to file' and 'Code' buttons. 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 | 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 |

On the right side, there is an 'About' section with a detailed description of the code example, followed by a list of tags: 'sample', 'sample-code', 'sap-logistics-business-network', 'global-track-and-trace', 'Ibn', 'gtt', and 'sales-order-fulfillment'. A 'Readme' link is also present.

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

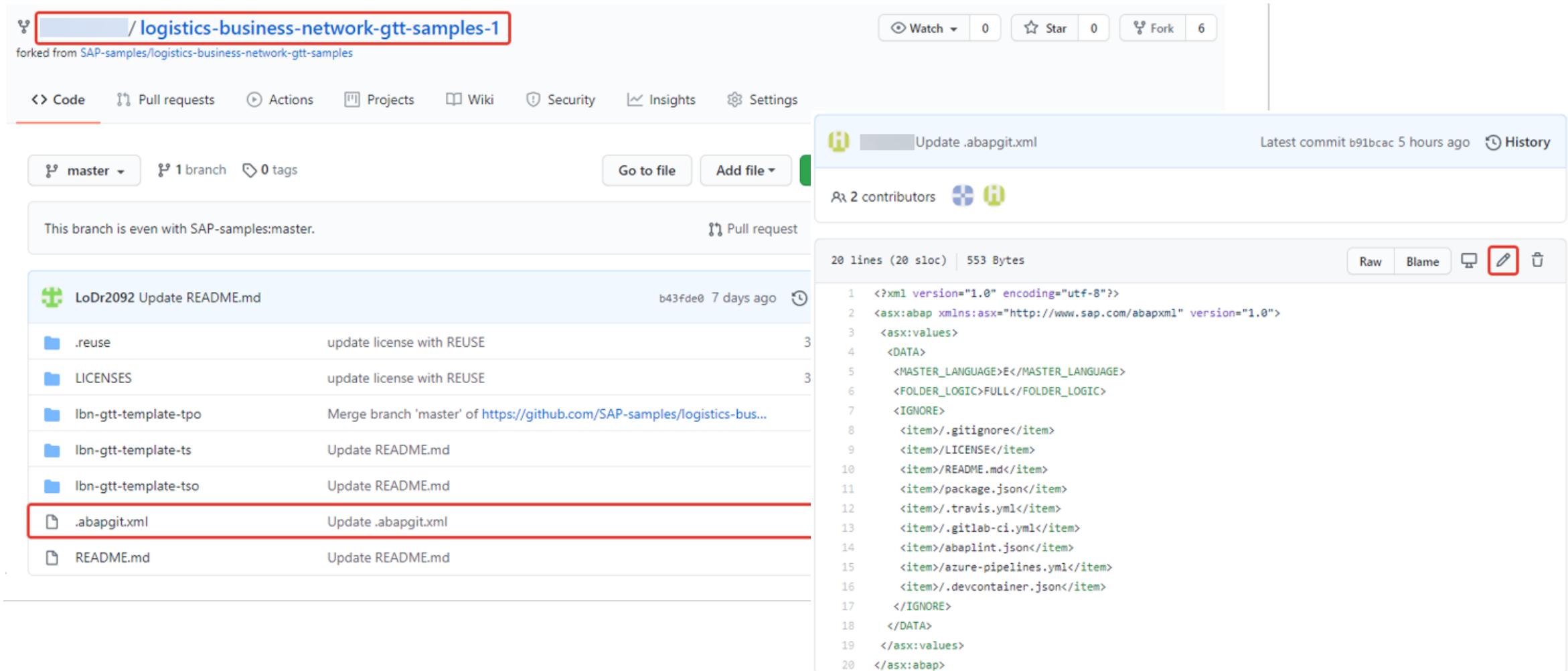
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, there are buttons for 'master' (selected), '1 branch', '0 tags', 'Go to file', 'Add file', and 'Code'. The repository description states: 'This branch is even with SAP-samples:master.' A 'Pull request' and 'Compare' button are also present. The commit history table lists several commits, with the last commit, 'Update .abapgit.xml', highlighted by a red box. The commit details show it was updated 14 days ago. The repository sidebar includes sections for 'About', 'Readme', 'Releases' (no releases published), and 'Packages' (no packages published).

| File | Description | Updated |
|---------------------------|--|--------------------|
| 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 URL in the address bar is highlighted with a red box. The repository has been forked from SAP-samples/logistics-business-network-gtt-samples. The master branch is selected, showing 1 branch and 0 tags. A pull request is visible. The '.abapgit.xml' file is listed in the commit history, and its edit button is highlighted with a red box. The file content is displayed in a code editor, showing XML configuration for ABAP logic and ignore patterns.

/ logistics-business-network-gtt-samples-1

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

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file

This branch is even with SAP-samples:master.

Pull request

| File / Commit | Description | Author | Date |
|---------------------------|--|---------|------------|
| LoDr2092 Update README.md | update license with REUSE | b43fde0 | 7 days ago |
| .reuse | update license with REUSE | | 3 |
| LICENSES | update license with REUSE | | 3 |
| Ibn-gtt-template-tpo | Merge branch 'master' of https://github.com/SAP-samples/logistics-bus... | | |
| Ibn-gtt-template-ts | Update README.md | | |
| Ibn-gtt-template-tso | Update README.md | | |
| .abapgit.xml | Update .abapgit.xml | | |
| README.md | Update README.md | | |

Update .abapgit.xml

Latest commit b91bcac 5 hours ago History

2 contributors

20 lines (20 sloc) | 553 Bytes

Raw Blame  

```
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 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, with the line '

```
<?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...

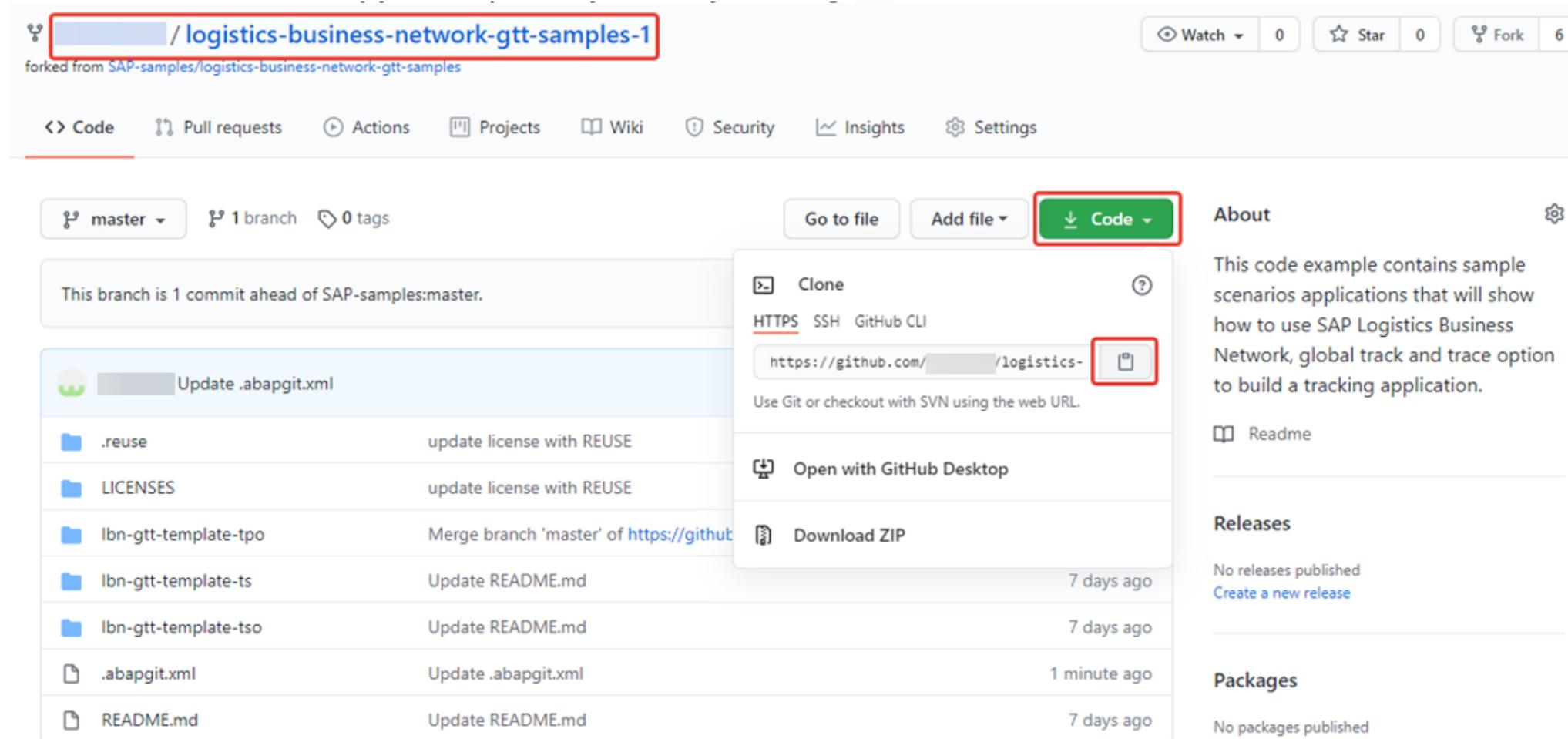
Commit directly to the master branch.

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 repository is forked from 'SAP-samples/logistics-business-network-gtt-samples'. The 'Code' dropdown menu is open, and the 'Clone' option is selected, with a red box highlighting the copy icon next to the URL. The URL is <https://github.com/.../logistics-...>.

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

| File | Description | Last Commit |
|----------------------|--|--------------|
| .reuse | update license with REUSE | 7 days ago |
| LICENSES | update license with REUSE | 7 days ago |
| Ibn-gtt-template-tpo | Merge branch 'master' of https://github.com/ | 7 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 | 1 minute ago |
| README.md | Update README.md | 7 days ago |

About

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

Readme

Releases

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

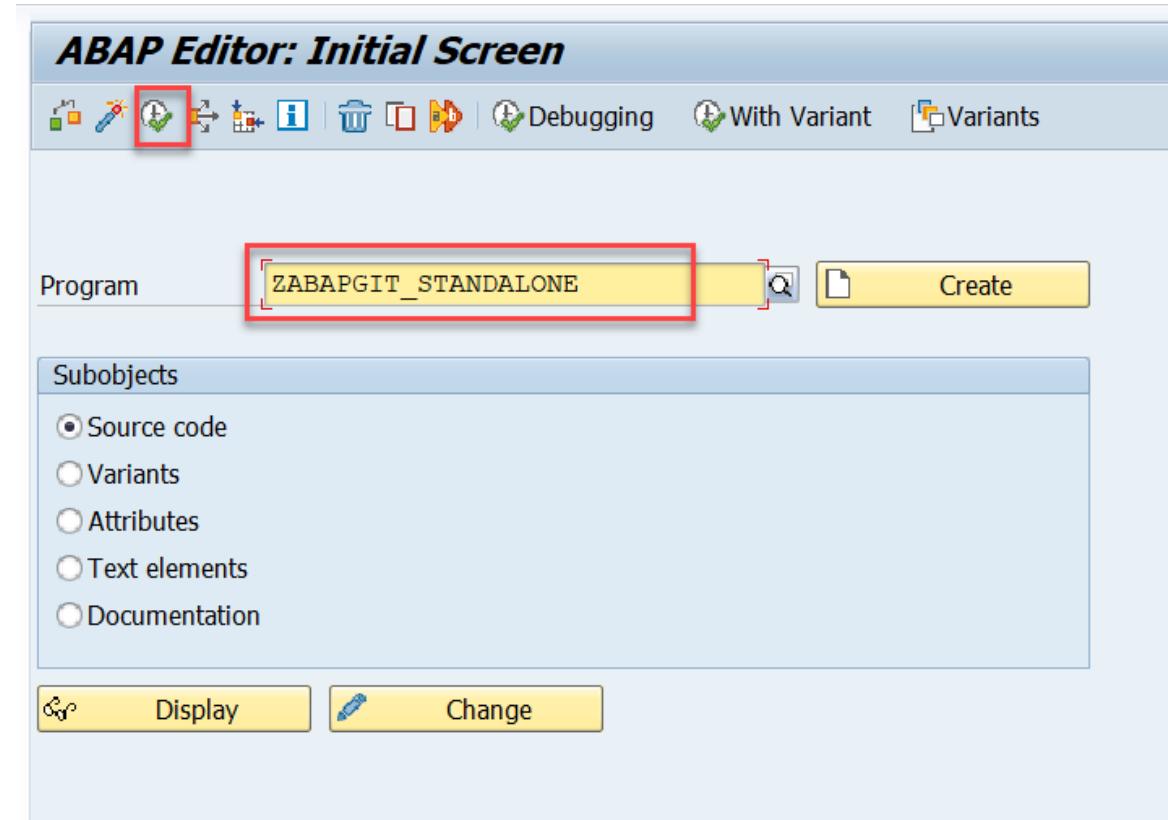
Packages

No packages published

STEP 4: Download ABAP Code from GitHub

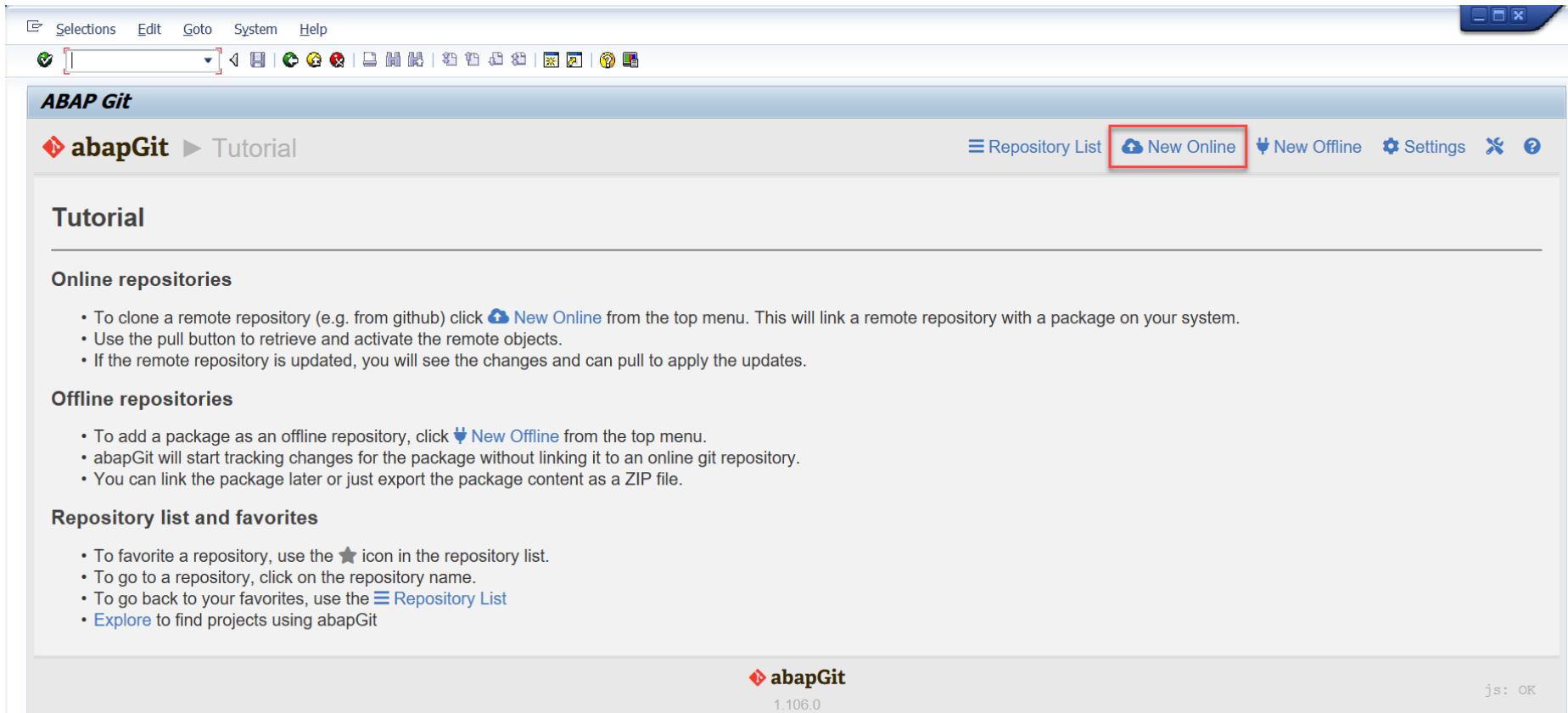
4-1: Enter T-code **SE38** and fill the report name from STEP 1, **ZABAPGIT_STANDALONE**

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`

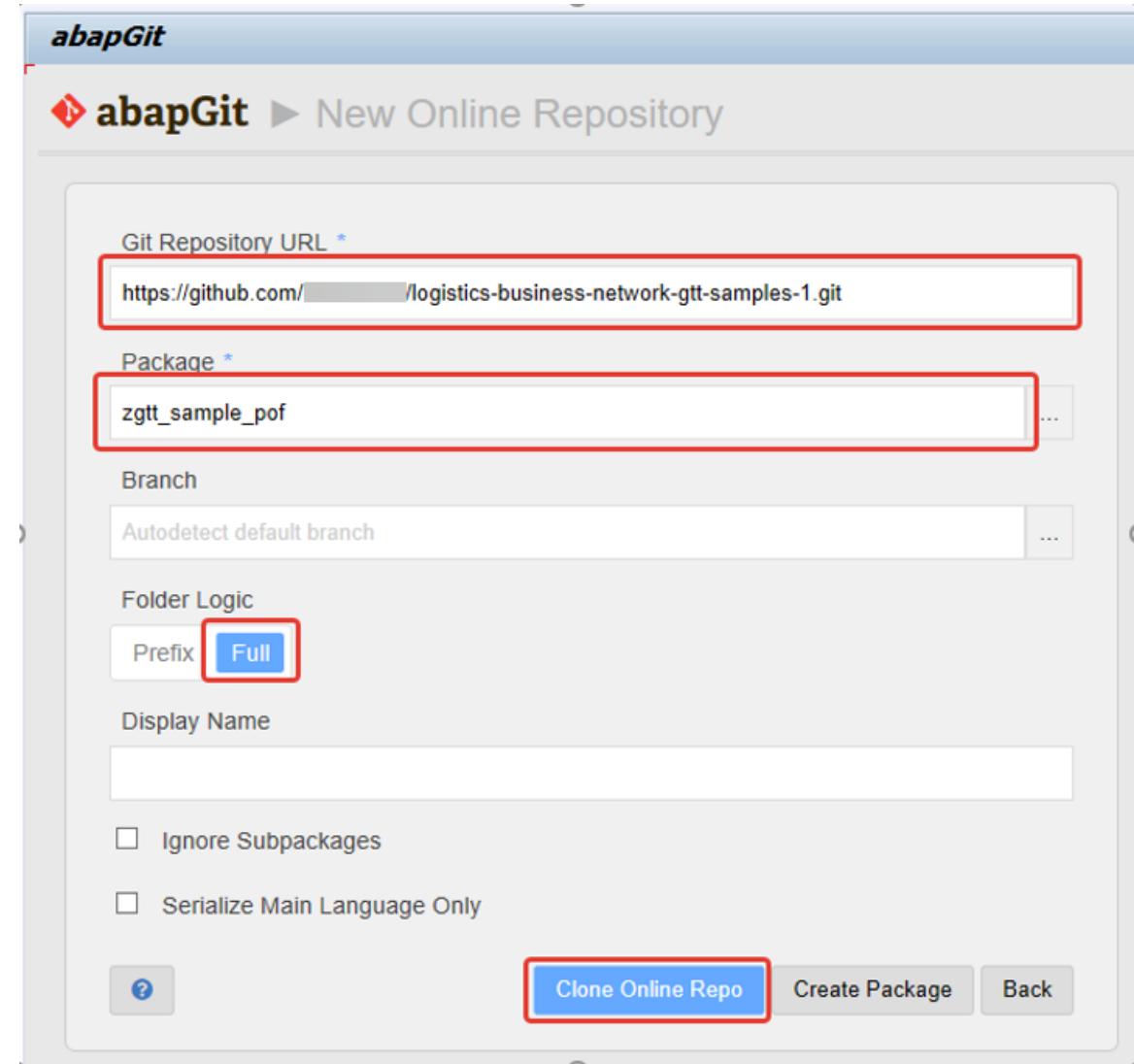
Caution:

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.

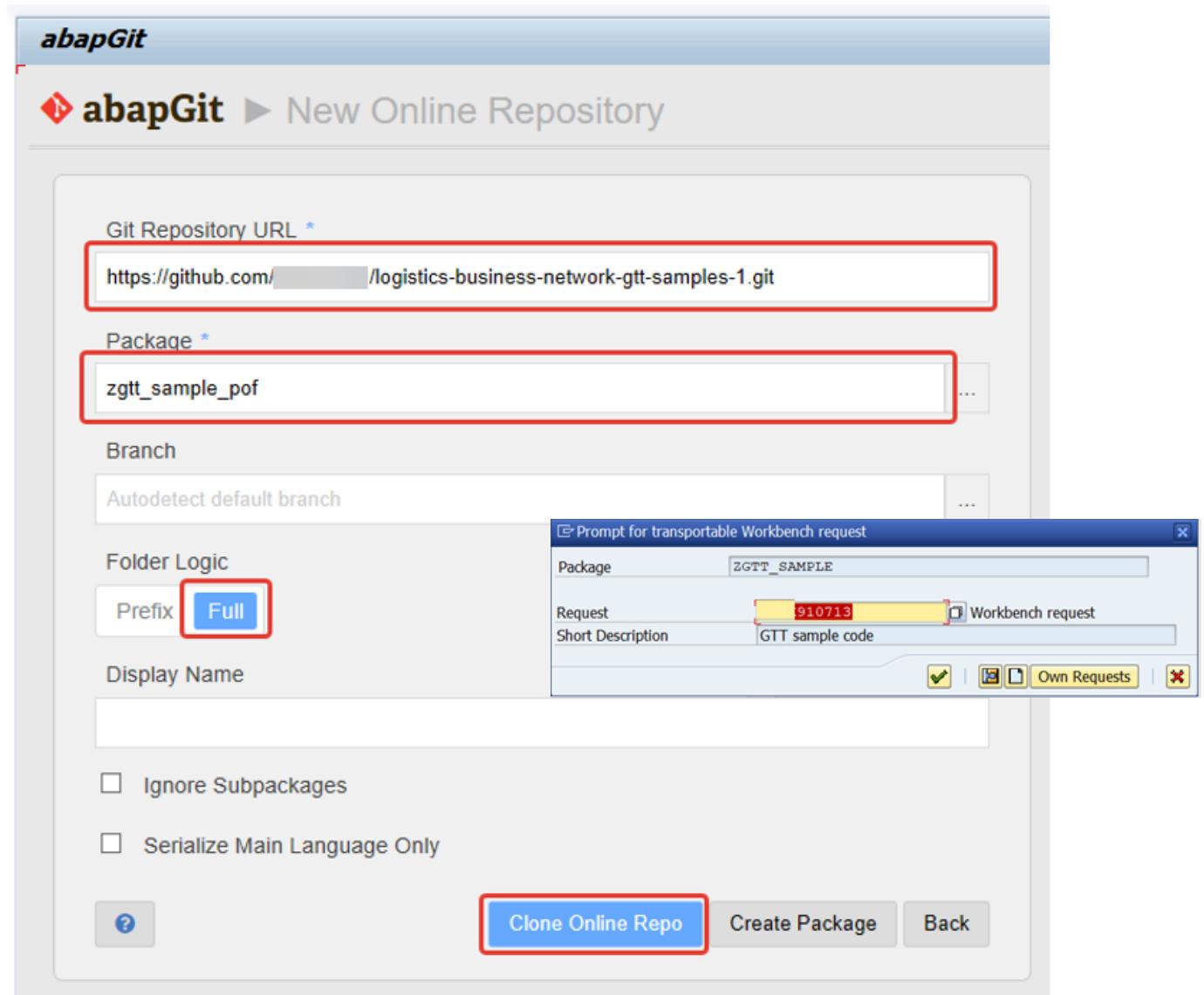
4-6: Set *Full* for **Folder Logic**

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

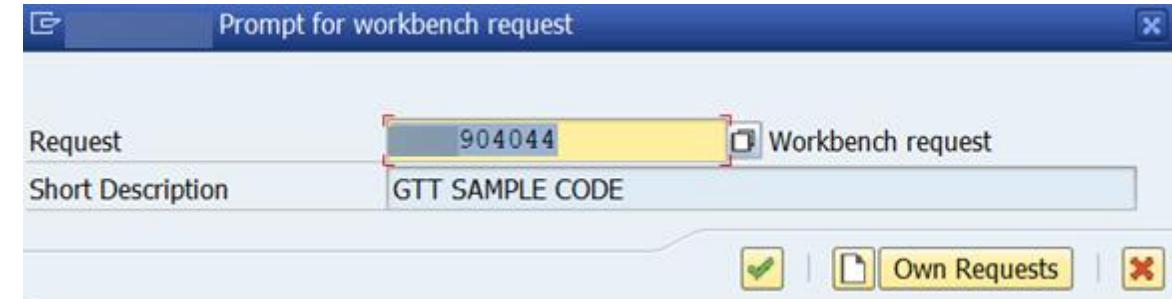
4-9: Click **Pull** to pull down the latest version of sample code.

The screenshot shows the abapGit interface. At the top, there's a navigation bar with 'abapGit' and a 'Repository' link. Below it, a repository card for 'logistics-business-network-gtt-samples-1' is displayed, showing its URL, last commit hash ('baaf604'), and current branch ('master'). A 'zgtt_sample_pof' folder icon is also present. A toolbar below the card includes buttons for 'Pull' (which is highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'View', 'Refresh', and a settings gear icon.

| Type | Name | Path | diff |
|-------------------------|----------------------------|---|--|
| 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] diff [A] |
| DEVC | ZGTT_SAMPLE_POF | /lbn-gtt-template-tpo/abap/zsrc/package.devcl.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt00.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt01.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt10.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt11.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt20.xml /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.abap /lbn-gtt-template-tpo/abap/zsrc/zpof_gtt.fugr.lzpof_gtt30.xml | diff [D] diff [A] 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 GTT SOF - Delivery Order's Planned Event Relevance Indicator |
| Data Elements | ZPOF_KOSTA ZPOF_PDSTK ZPOF_PKSTA ZPOF_WBSTA GTT POF - Planned Event Relevance for Put Away of Delivery GTT SOF - Planned Event Relevance for POD of delivery GTT POF - Planned Event Relevance for Packing of Delivery GTT POF - Planned Event Relevance for Goods Mvt of Delivery |
| Class Library | ZCL_IM_POF_GTT_LE_SHIPMENT Implementation Class for BAdI Implementation ZPOF_GTT_L... |
| Functions Groups | ZPOF_GTT POF GTT |
| Message Classes | ZPOF_GTT Purchasing Order Fulfillment Messages |
| Enhancements | ZPOF_GTT_LE_SHIPMENT Shipment Cross TP Update |
| Classic BAdIs (Impl.) | |

The screenshot shows the SAP Object Navigator interface. The top navigation bar includes 'Object Navigator', 'Edit Object' (with icons for back, forward, search, and refresh), and a toolbar with various icons. The main area is titled 'Repository Browser' and shows a tree view of objects under the package 'ZGTT_SAMPLE_POF'. A red box highlights the package name in the search bar and the entire tree view of objects. The table below lists the object names and their descriptions. The 'ZPOF_GTT' table is specifically highlighted with a red box in the tree view.

C) Download ABAP Code from GitHub

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

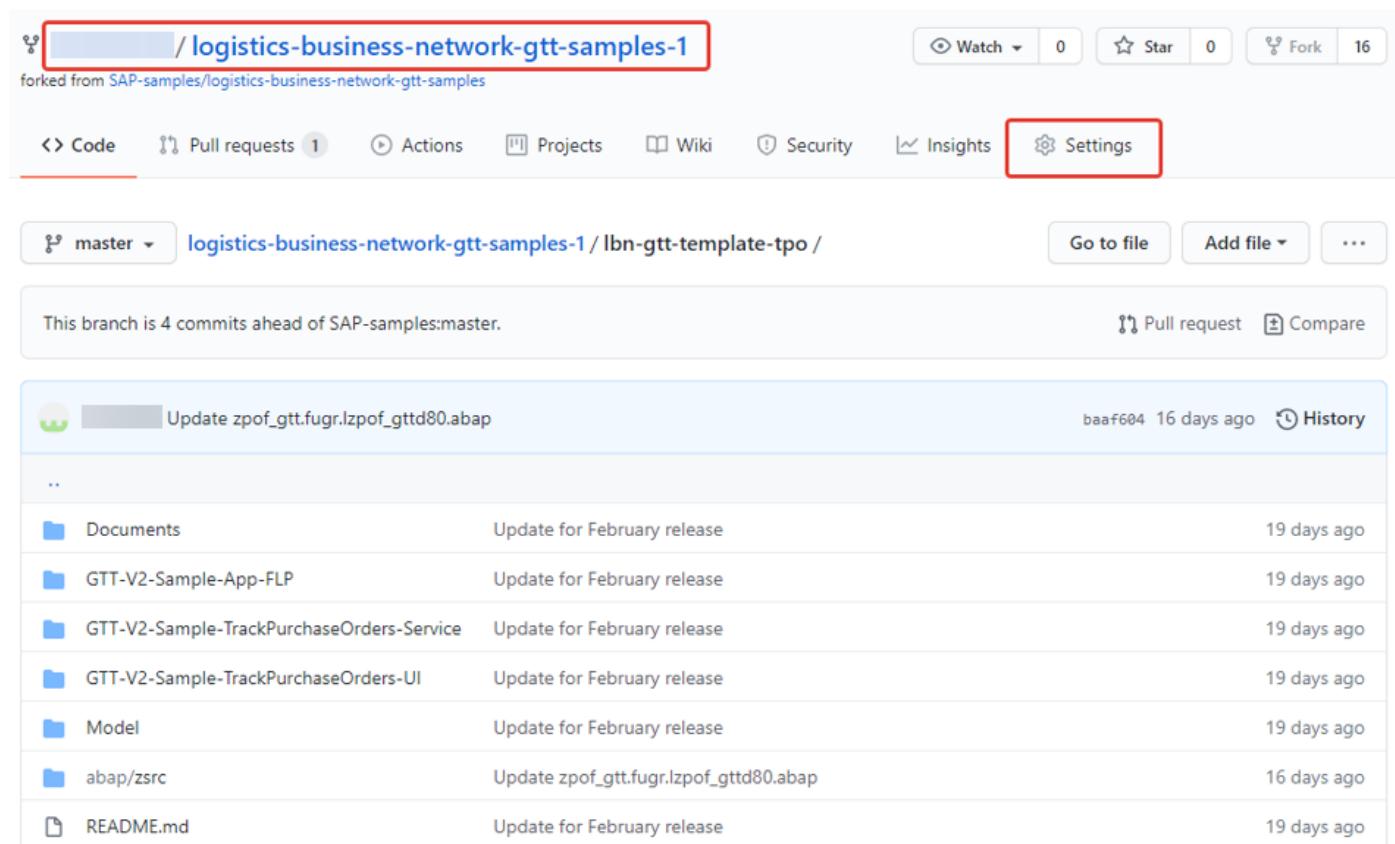


STEP 1: Delete the User's Account Repository

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

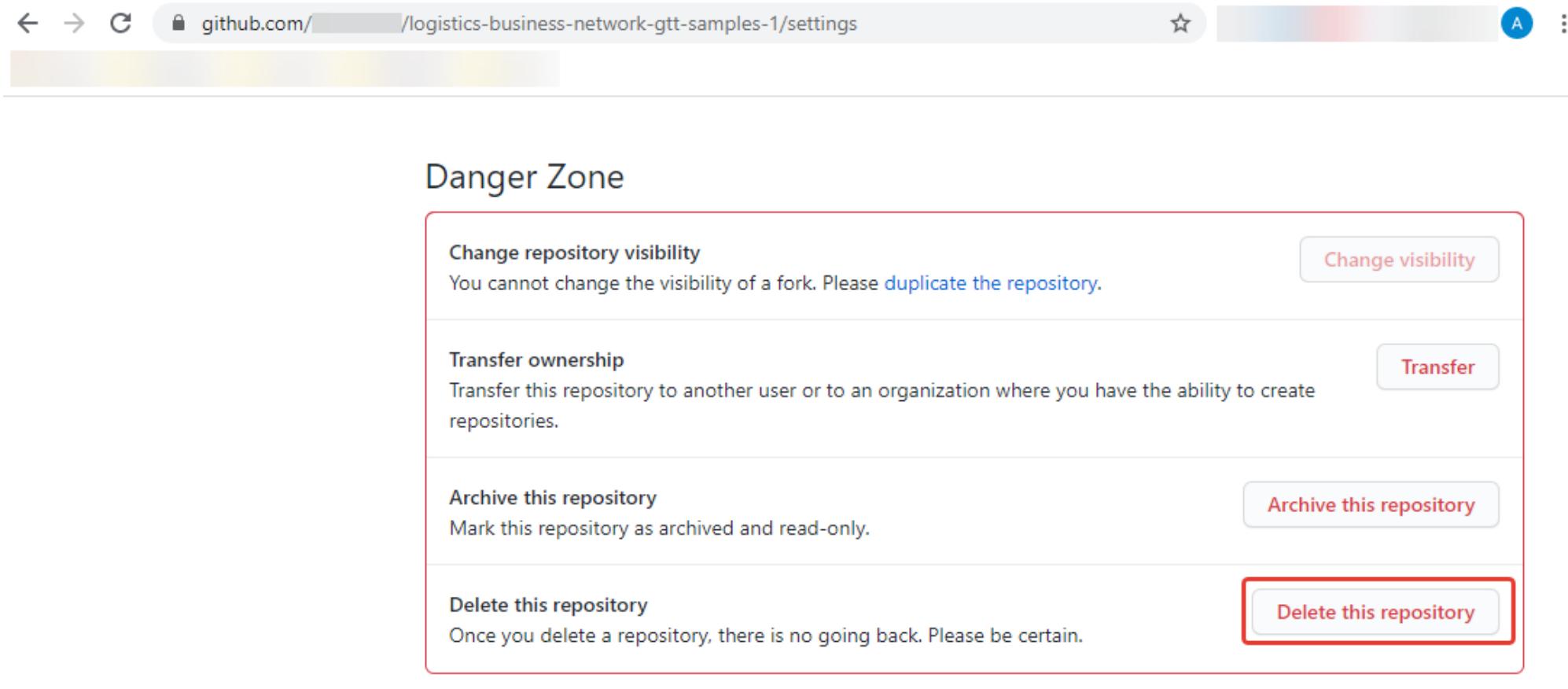
In the latest release, there is some code changes in public sample code, you need to update the local code according to the latest public sample code.

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



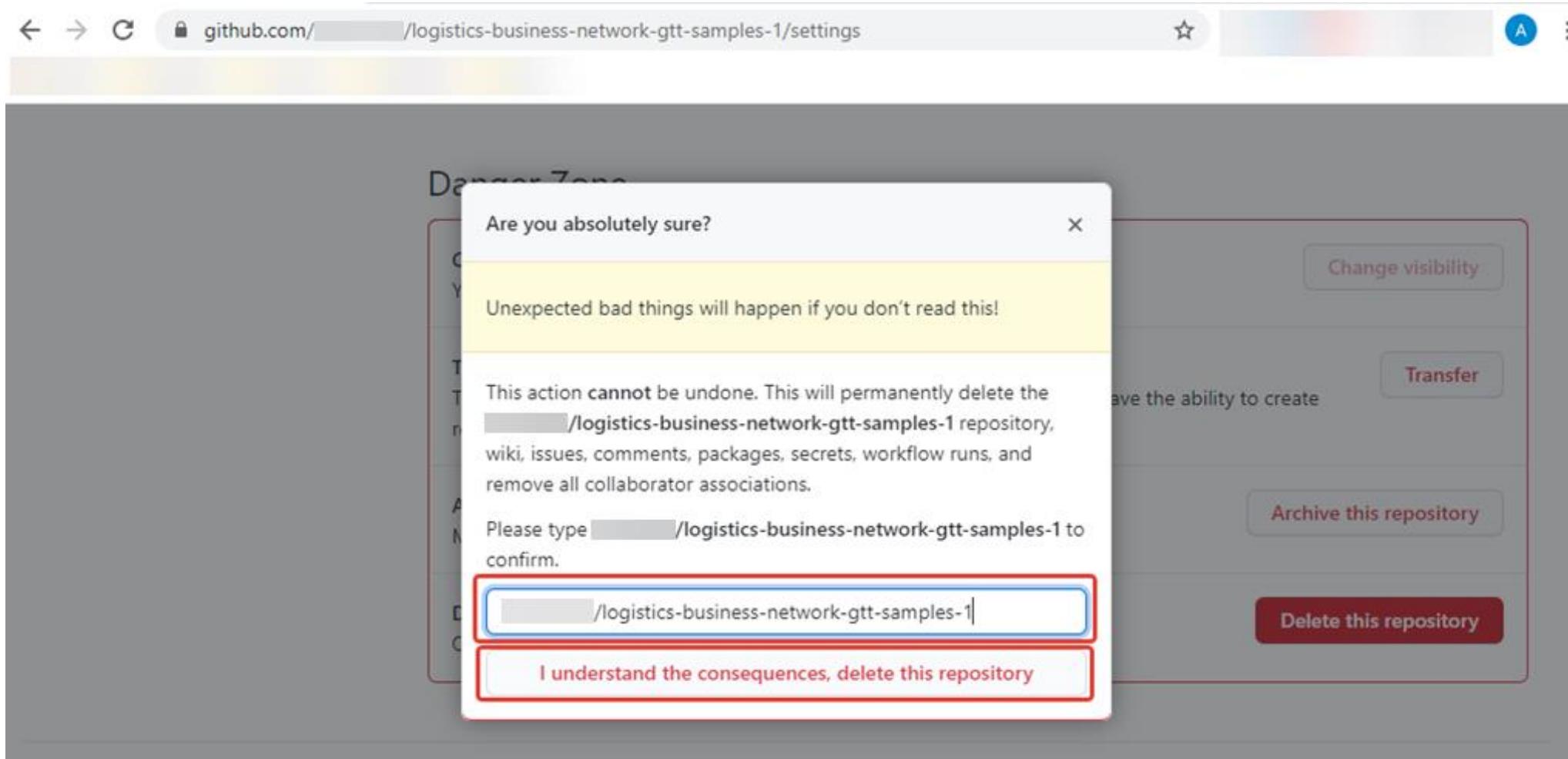
STEP 1: Delete the User's Account Repository

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

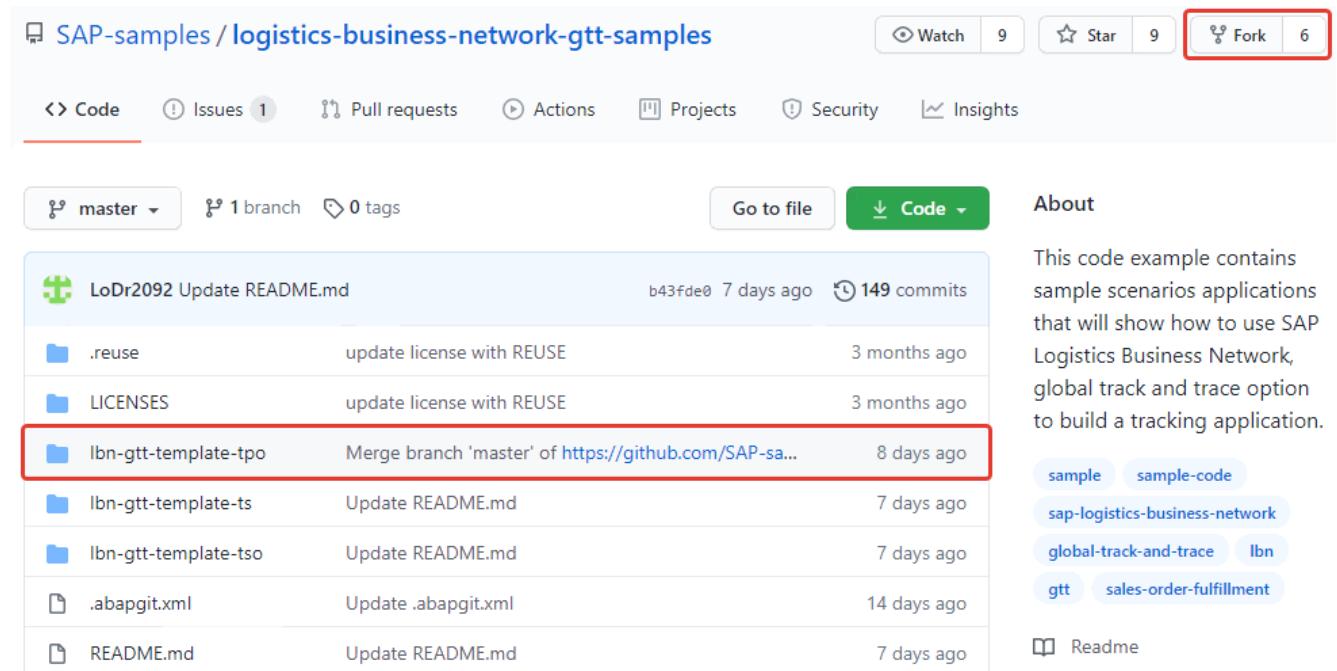
1-5: The user's account repository is deleted.

The screenshot shows a web browser window with the URL "github.com" in the address bar. The GitHub homepage is displayed, featuring the navigation bar with "Pull requests", "Issues", "Marketplace", and "Explore" links. A prominent red box highlights a success message in the center of the page: "Your repository "/logistics-business-network-gtt-samples-1" was successfully deleted." Below this message, there is a modal window titled "Introduce yourself" with instructions on how to create a README file. The modal also contains a sample README.md content with numbered items 1 through 5, each preceded by an emoji icon. At the bottom of the modal are "Dismiss this" and "Continue" buttons. On the left side of the page, there are sidebar sections for "Create your first project" (with "Create repository" and "Import repository" buttons), "Working with a team?", and a general GitHub introduction.

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 right corner features a red box around the 'Fork' button, which has a value of '6'. Below the header, there are tabs for Code, Issues (1), Pull requests, Actions, Projects, Security, and Insights. Under the Code tab, it shows 'master' branch, '1 branch', and '0 tags'. There are buttons for 'Go to file' and 'Code'. On the right, there is an 'About' section with a detailed description of the code example. Below the description is a list of recent commits, with the commit 'Ibn-gtt-template-tpo' highlighted by a red box. The commit details show it merged the 'master' branch from 'https://github.com/SAP-sa...' 8 days ago. Other commits listed include 'LoDr2092 Update README.md', '.reuse', 'LICENSES', 'Ibn-gtt-template-ts', 'Ibn-gtt-template-tso', '.abapgit.xml', and 'README.md'.

| Commit | Description | 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 is a fork from 'SAP-samples/logistics-business-network-gtt-samples'. The page includes a navigation bar with 'Code' (selected), 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, there are buttons for 'master' branch, '1 branch', '0 tags', 'Go to file', 'Add file', and 'Code'. A message states 'This branch is even with SAP-samples:master.' with 'Pull request' and 'Compare' buttons. The main content area displays a list of commits:

| Commit | Message | 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 |

On the right side, there are sections for 'About', 'Readme', 'Releases', and 'Packages'. The 'About' section describes the repository as containing sample scenario applications for SAP Logistics Business Network. The 'Readme' section has a link to 'Readme'. The 'Releases' section indicates 'No releases published' with a 'Create a new release' button. The 'Packages' section indicates 'No packages published' with a 'Publish your first package' button.

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

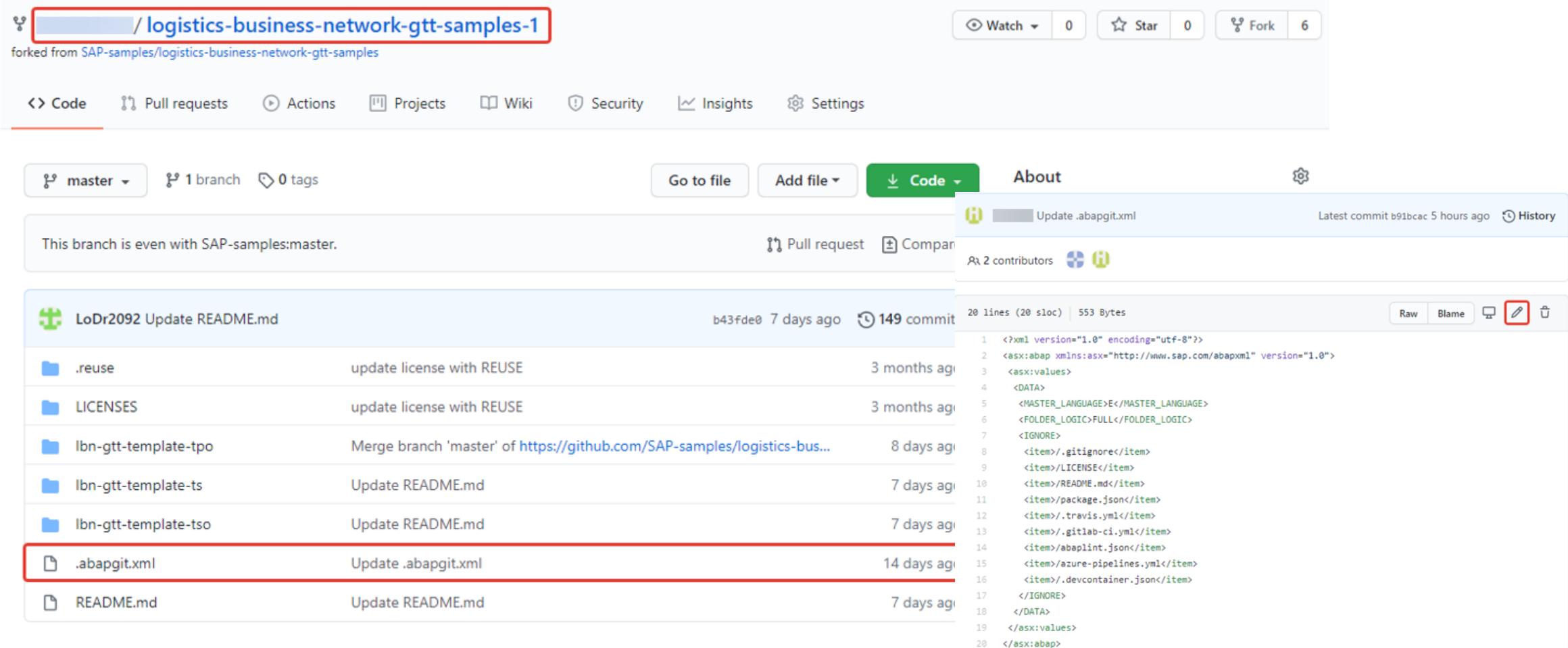
3-1: In the user’s account repository, click the file ‘.abapgit.xml’.

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 'Code' tab is selected. The 'master' branch is current, with 1 branch and 0 tags. A message states 'This branch is even with SAP-samples:master.' There is one pull request and a compare link. The commit history lists several commits, with the last commit, 'Update .abapgit.xml', highlighted by a red box. This commit was made 14 days ago by LoDr2092. Other commits include 'Update README.md' (7 days ago), 'Merge branch 'master' of https://github.com/SAP-samples/logistics-bus...' (8 days ago), and updates to LICENSES and REUSE files. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages', each with their respective descriptions and links.

| 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. In the commit list, the file '.abapgit.xml' is highlighted with a red border. The commit message is 'Update .abapgit.xml'. The code editor on the right displays the XML configuration 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 repository 'logistics-business-network-gtt-samples-1'. The file content is displayed in a code editor with line numbers. A red box highlights the line '<STARTING_FOLDER>/</STARTING_FOLDER>'. The commit message field contains 'Update .abapgit.xml'. The 'Commit changes' button is highlighted with a red border.

```
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>
```

Commit changes

Update .abapgit.xml

Add an optional extended description...

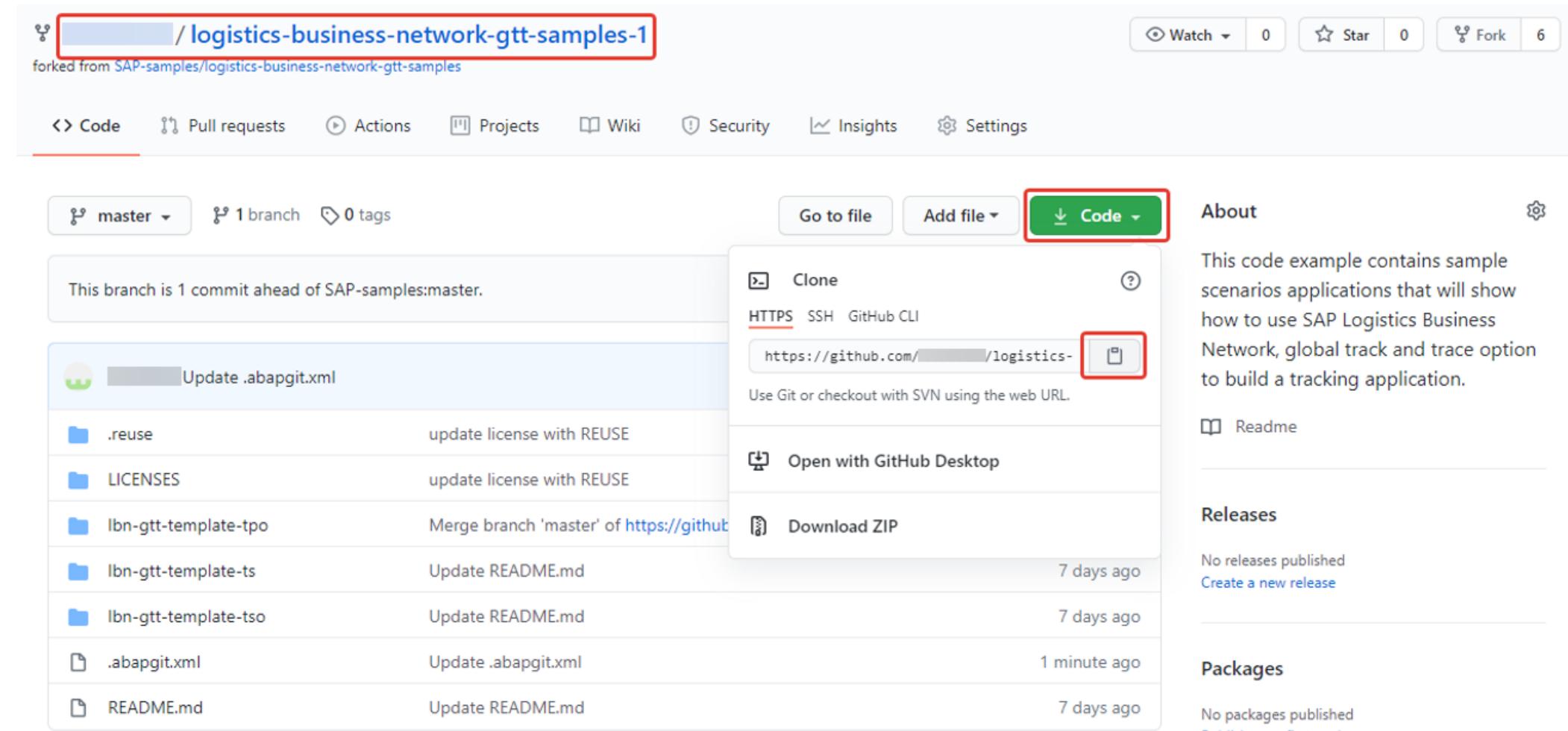
-o- Commit directly to the master branch.

↗ 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.

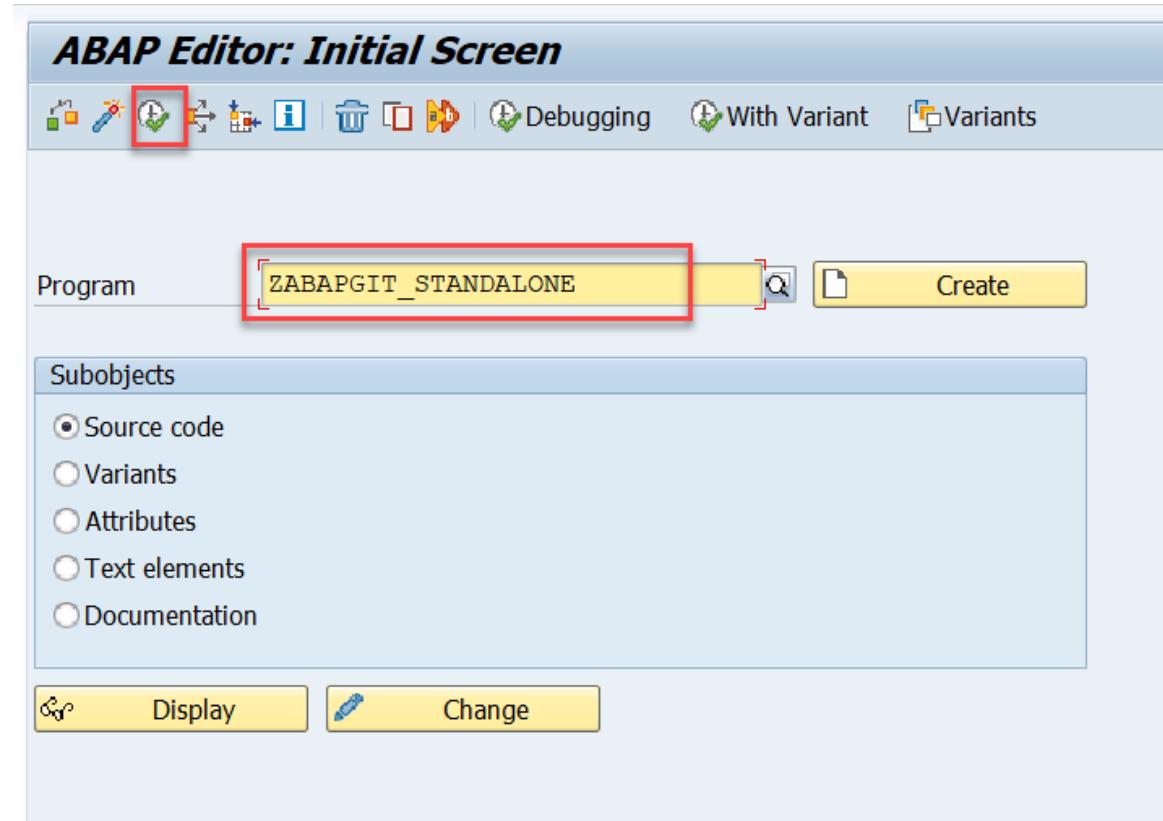


The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The URL in the address bar is highlighted with a red box. The 'Code' dropdown menu is open, and the 'Clone' section is selected, showing the HTTPS URL [https://github.com/\[REDACTED\]/logistics-](https://github.com/[REDACTED]/logistics-). A red box highlights the copy icon next to the URL. 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 'Readme' and 'Releases' sections are also visible.

STEP 4: Update ABAP Code from GitHub

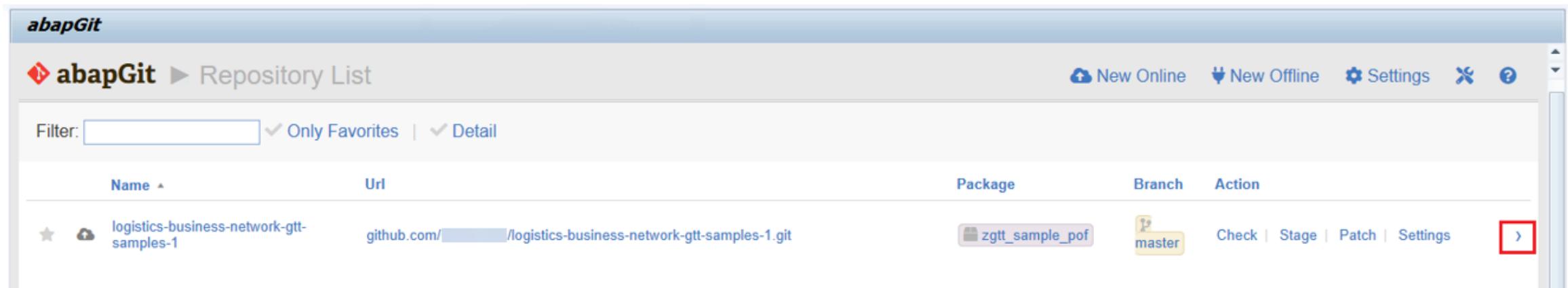
4-1: Enter T-code *SE38* and fill in the report name *ZABAPGIT_STANDALONE*.

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.

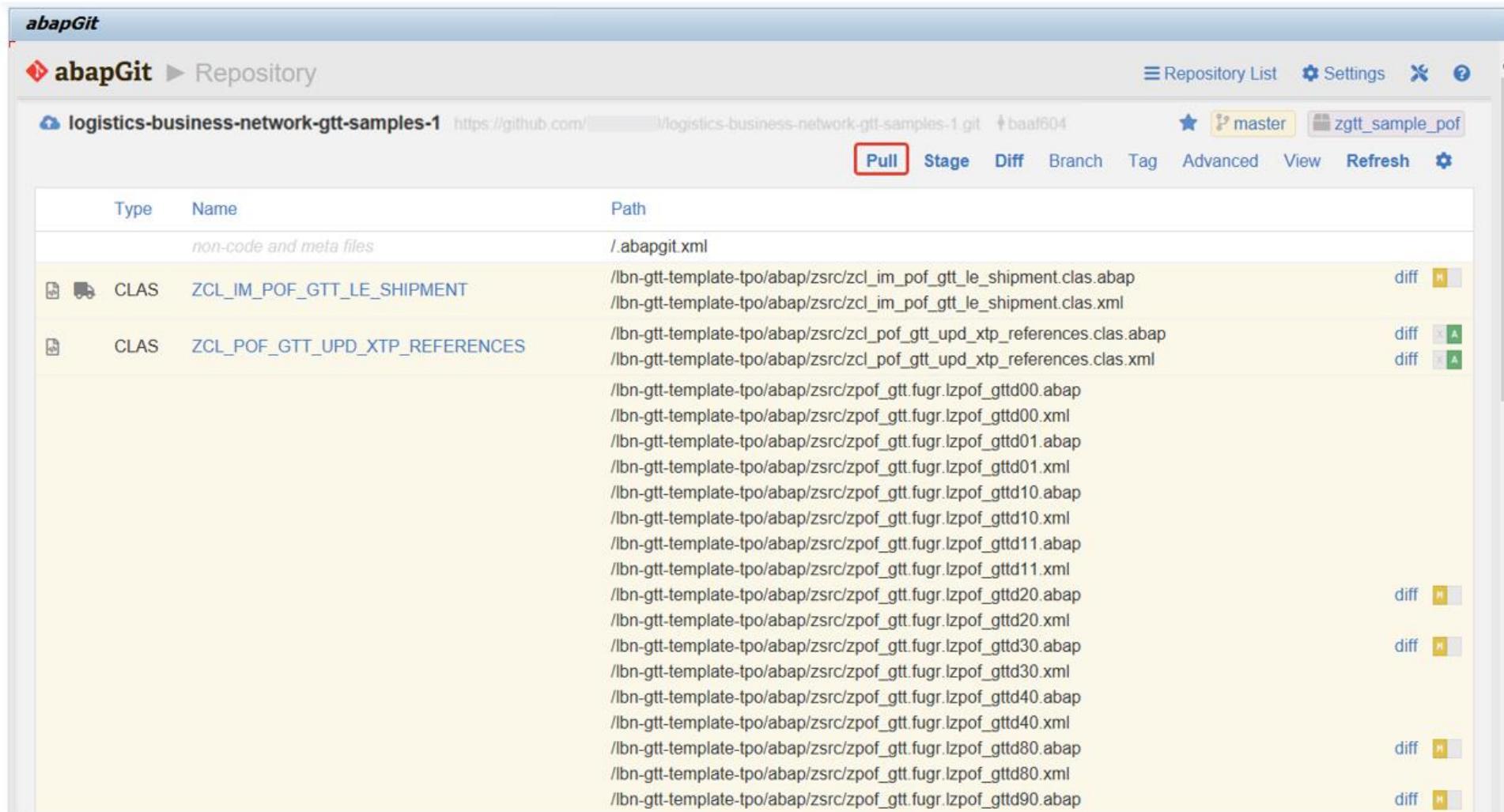


The screenshot shows the abapGit application interface. The title bar says "abapGit". The main area is titled "Repository List". There is a "Filter:" input field and two checkboxes: "Only Favorites" and "Detail". The repository list table has columns: Name, Url, Package, Branch, and Action. One repository row is shown:

| Name | Url | Package | Branch | Action |
|--|--|---|--|--|
| logistics-business-network-gtt-samples-1 | github.com/ ████████ /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 with the following details:

- Title Bar:** abapGit
- Breadcrumbs:** abapGit > Repository
- Repository Information:** logistics-business-network-gtt-samples-1, URL: https://github.com/[REDACTED]/logistics-business-network-gtt-samples-1.git, Commit: baaf604, Branch: master, Tag: zgtt_sample_pof
- Toolbar Buttons:** Pull (highlighted with a red box), Stage, Diff, Branch, Tag, Advanced, View, Refresh, Settings
- Table Headers:** Type, Name, Path
- Table Data:** The table lists several files under two categories:
 - non-code and meta files:** .abapgit.xml
 - CLAS (Class) files:**
 - 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
 - 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 Buttons:** diff (with yellow icon)

C) Download ABAP Code from GitHub

C3. Download Another ABAP Code from GitHub (TSOF)



STEP 1: Fork Sample Code Repository

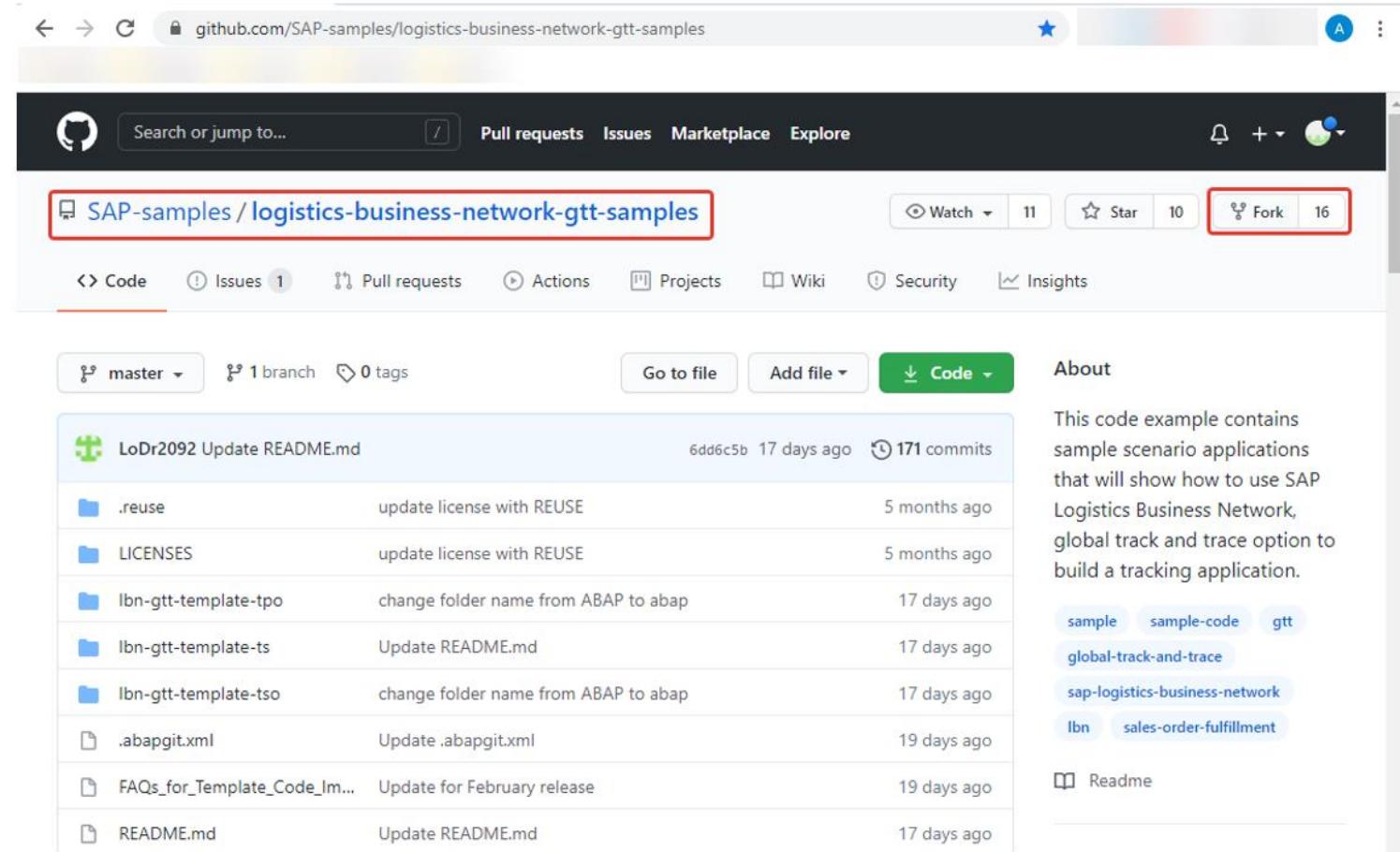
Prerequisite:

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.



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

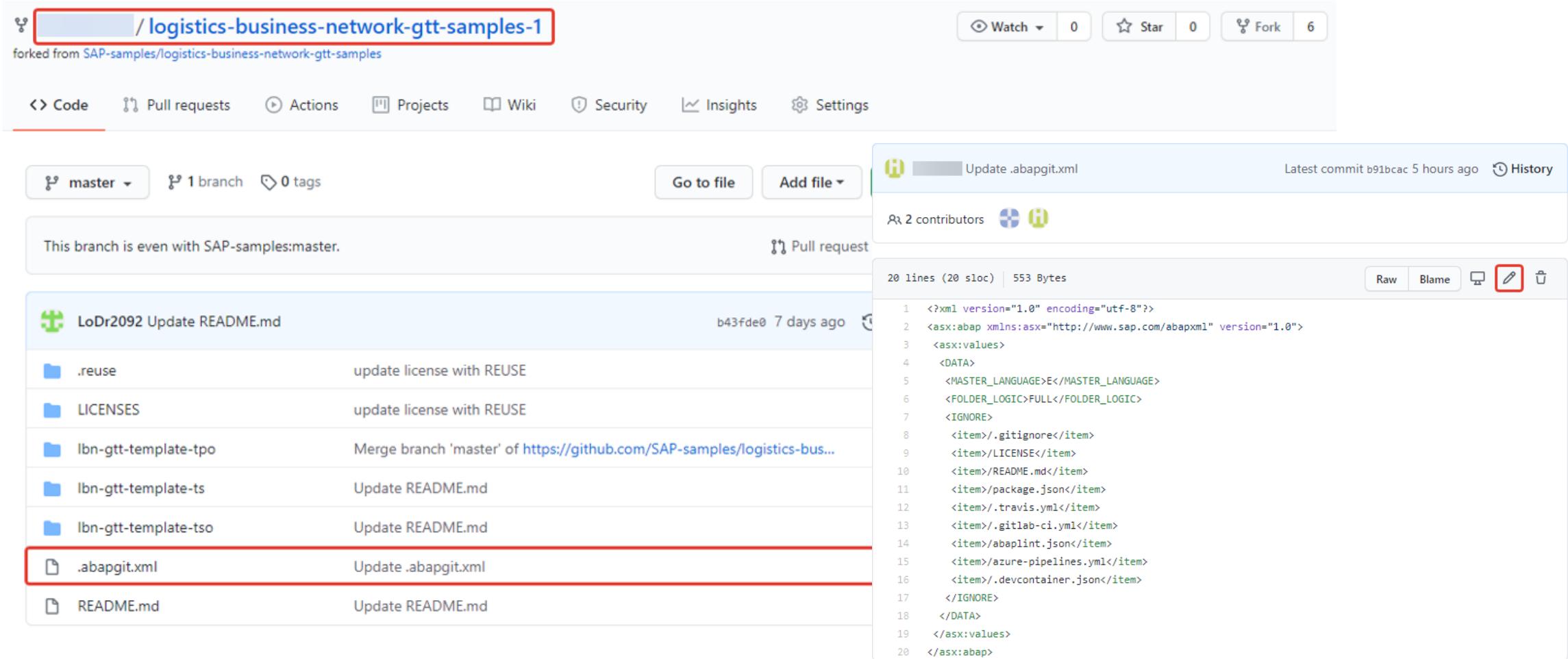
2-1: In the user’s account repository, click the file ‘.abapgit.xml’.

The screenshot shows a GitHub repository page for 'logistics-business-network-gtt-samples-1'. The repository is a fork 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 .reuse files, as well as merges and updates to .abapgit.xml and README.md. The '.abapgit.xml' file is highlighted with a red border. The right sidebar contains sections for 'About', 'Readme', 'Releases', and 'Packages', all of which are currently empty.

| File / Action | Description | 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 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 was 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. On the left, a list of files shows changes: 'LoDr2092 Update README.md' (commit b43fde0, 7 days ago), '.reuse' (update license with REUSE), 'LICENSES' (update license with REUSE), 'lbn-gtt-template-tpo' (Merge branch 'master' of https://github.com/SAP-samples/logistics-bus...), 'lbn-gtt-template-ts' (Update README.md), 'lbn-gtt-template-tso' (Update README.md), '.abapgit.xml' (Update .abapgit.xml), and 'README.md' (Update README.md). The '.abapgit.xml' file is highlighted with a red box. On the right, the file content is displayed with syntax highlighting for XML. The code starts with `<?xml version="1.0" encoding="utf-8"?>` and defines an ASX namespace. It contains sections for DATA and IGNORE, listing various files like .gitignore, LICENSE, README.md, package.json, travis.yml, gitlab-ci.yml, abaplint.json, azure-pipelines.yml, devcontainer.json, and .gitattributes.

```
<?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 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 a repository named 'logistics-business-network-gtt-samples-1'. The file being edited is '.abapgit.xml' in the 'master' branch. The code editor shows the following XML snippet:

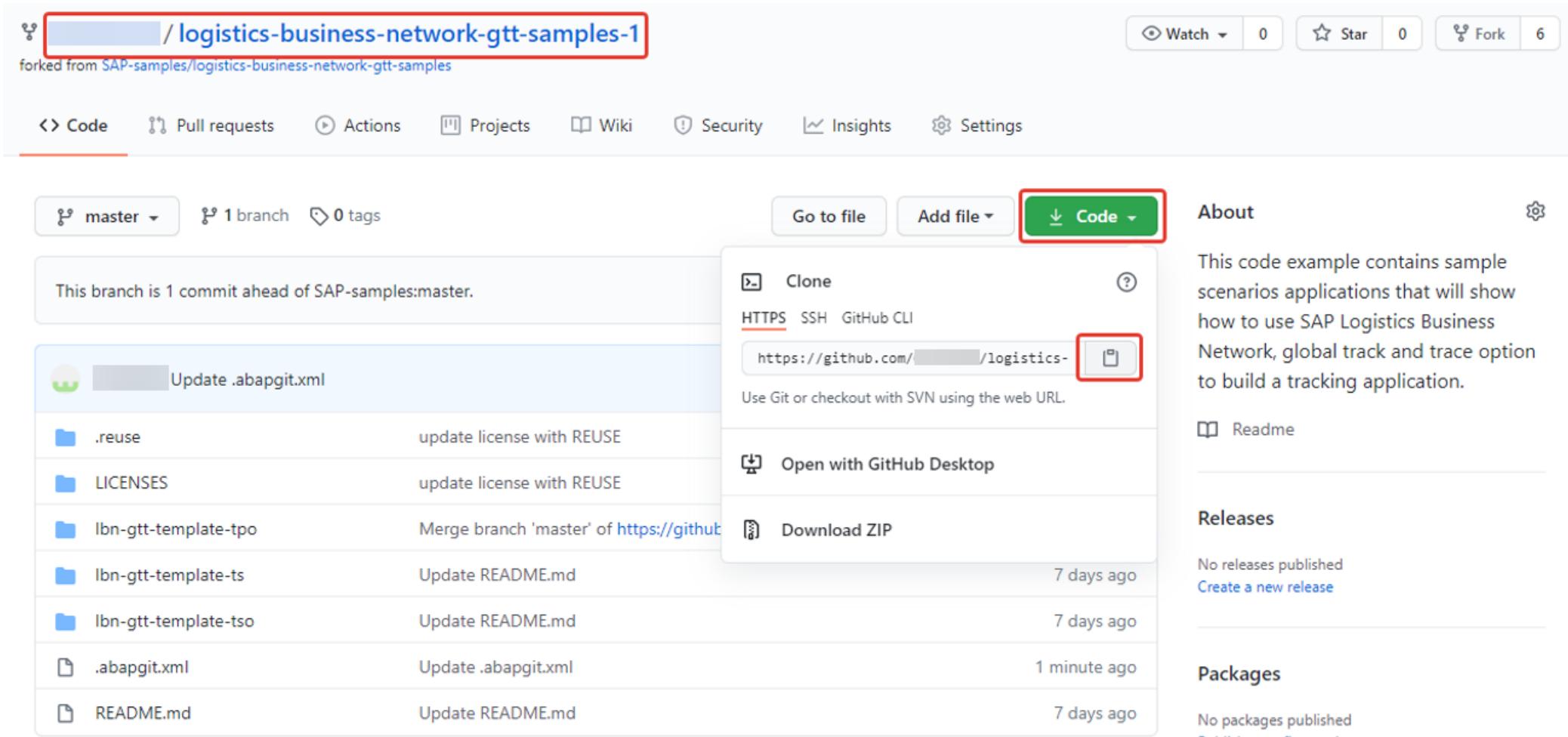
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3 <asx:values>
4 <DATA>
5 <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6 <STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>
7 <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8 <IGNORE>
9   <item>.abapgit.xml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

A red box highlights the line '<STARTING_FOLDER>/lbn-gtt-template-tso/abap/zsrc/</STARTING_FOLDER>'. To the right, a 'Commit changes' dialog is open, containing the following fields:

- Commit message: 'Update .abapgit.xml'
- Description placeholder: 'Add an optional extended description...'
- Branch selection:
 - 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 button (highlighted with a red border)
- Cancel button

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 repository was forked from SAP-samples. 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 (HTTPS, SSH, GitHub CLI), Open with GitHub Desktop, and Download ZIP. The 'Clone' option has a red box around its copy icon. The 'About' section describes the repository as containing sample scenarios for SAP Logistics Business Network, global track and trace options, and building a tracking application. It also links to Readme, Releases (none published), and Packages (none published).

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

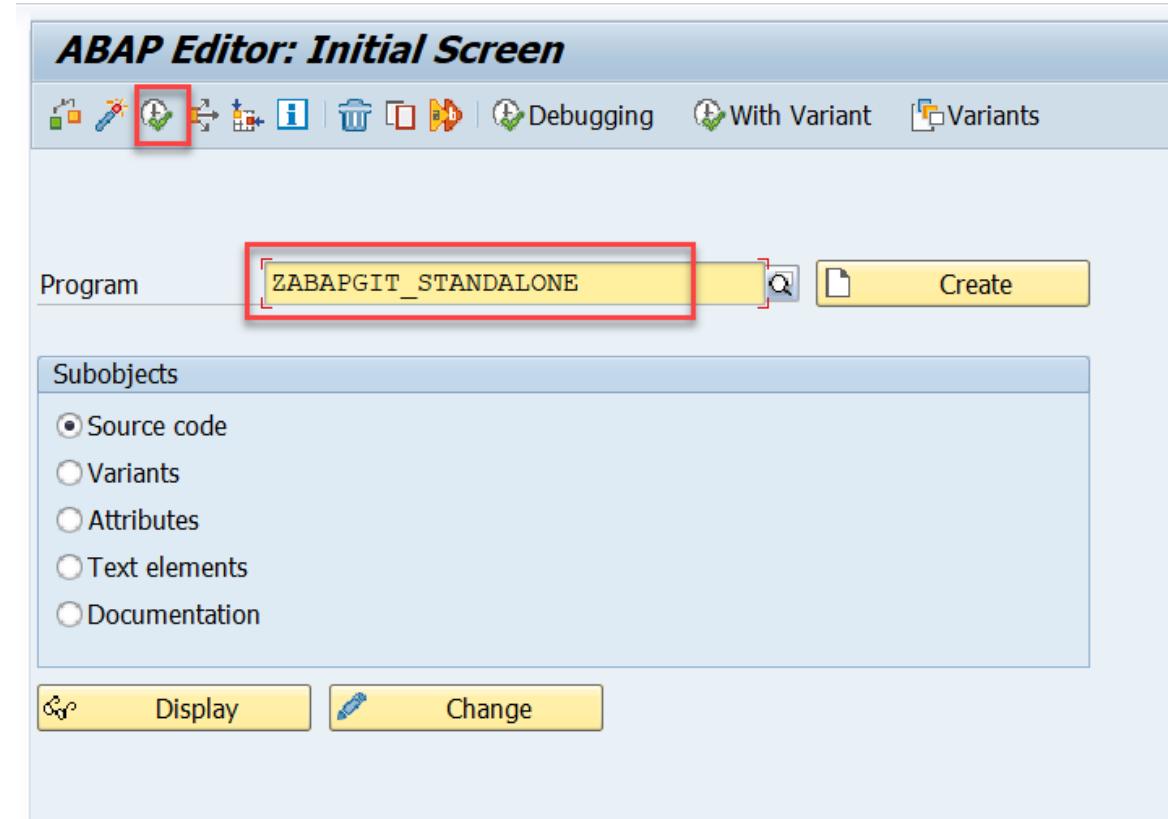
Releases

Packages

STEP 3: Remove TPOF Repository in ABAPGit

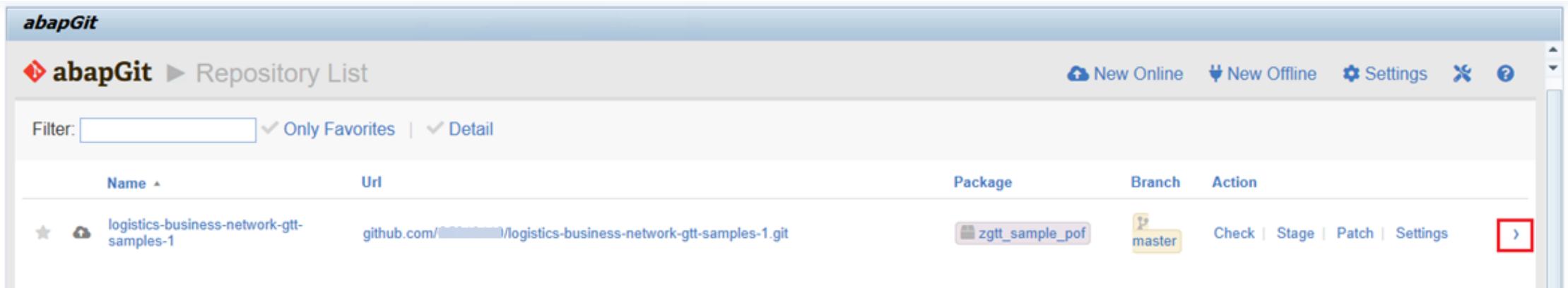
3-1: Enter T-code *SE38* and fill in the report name *ZABAPGIT_STANDALONE*.

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



STEP 3: Remove TPOF Repository in ABAPGit

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

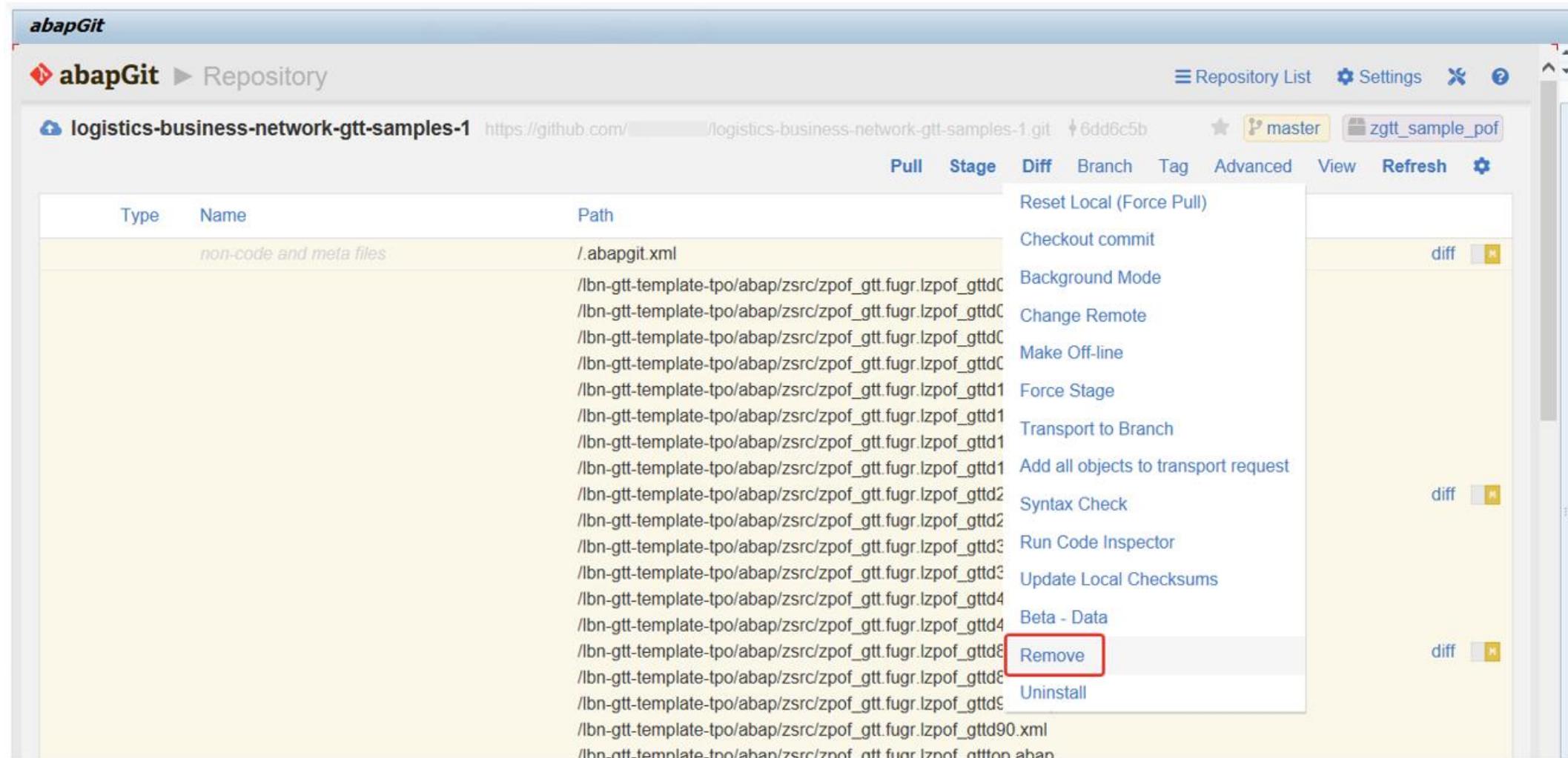


The screenshot shows the ABAPGit application interface. The title bar says "abapGit". The main area is titled "abapGit ► Repository List". There is a "New Online" button, a "New Offline" button, a "Settings" button, and a close button. Below the title, there is a "Filter:" input field, a "Only Favorites" 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 blue arrow icon. The blue arrow icon 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  |

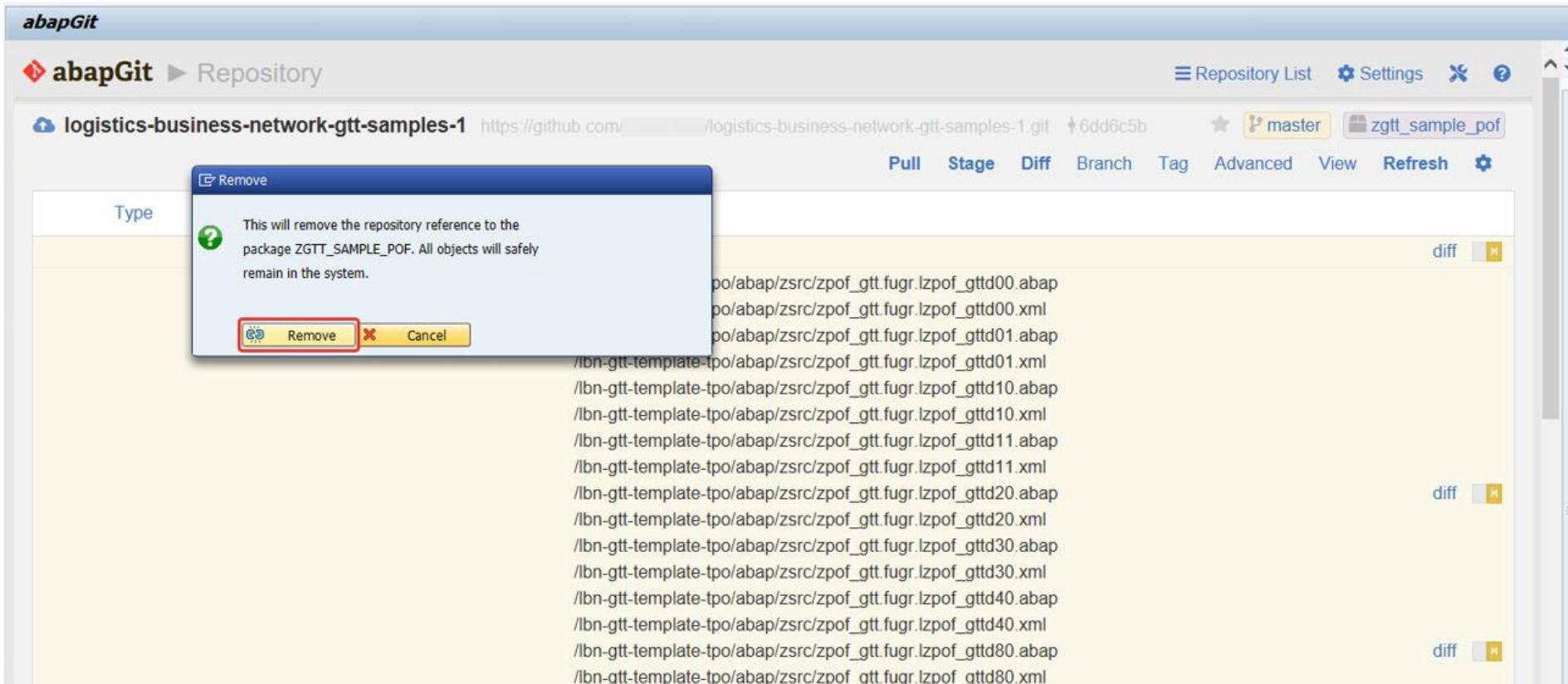
STEP 3: Remove TPOF Repository in ABAPGit

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.

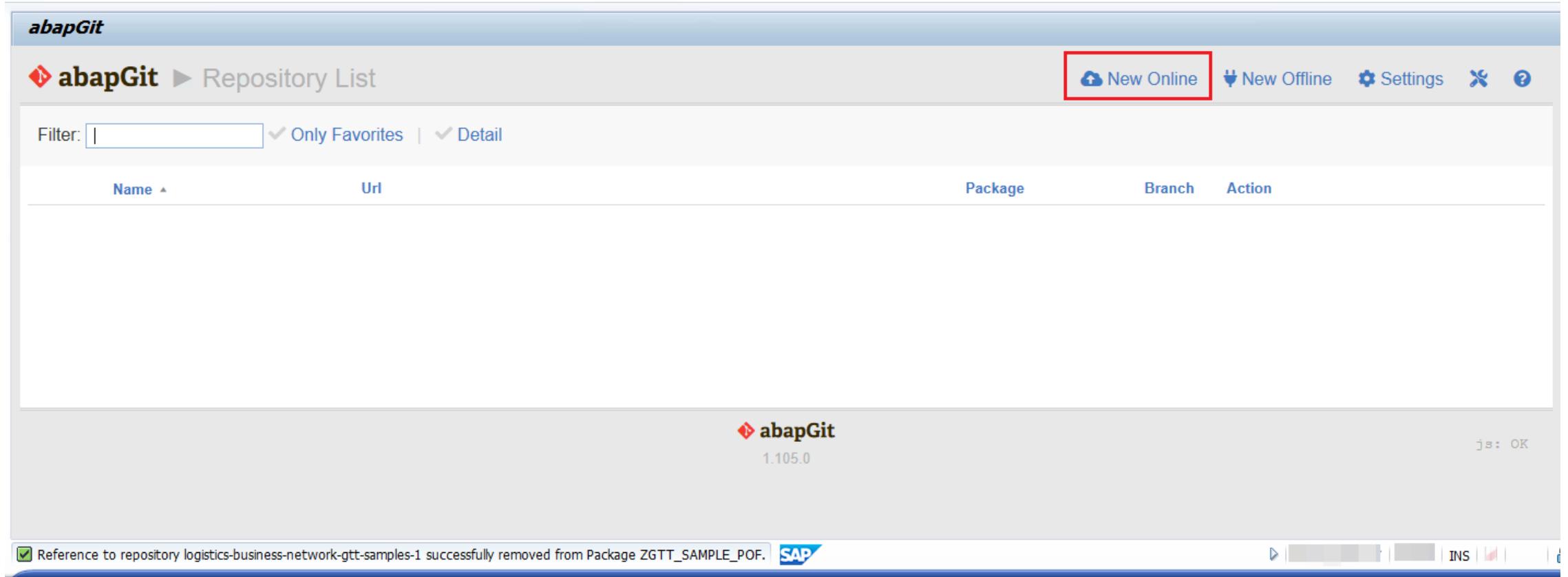


3-6: After repository removal you will see the following message:

Reference to repository logistics-business-network-gtt-samples-1 successfully removed from Package ZGTT_SAMPLE_POF. 

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". Below the header, the main title "abapGit ► Repository List" is displayed. On the right side of the header, there are several buttons: "New Online" (highlighted with a red box), "New Offline", "Settings", and others. Below the header, there is a search bar labeled "Filter:" and two checkboxes: "Only Favorites" and "Detail". The main area is a table with columns: "Name", "Url", "Package", "Branch", and "Action". At the bottom of the screen, there is a footer bar with the abapGit logo, version "1.105.0", and a message "js: OK". The status bar at the very bottom shows a success message: "Reference to repository logistics-business-network-gtt-samples-1 successfully removed from Package ZGTT_SAMPLE_POF." and the SAP logo.

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>

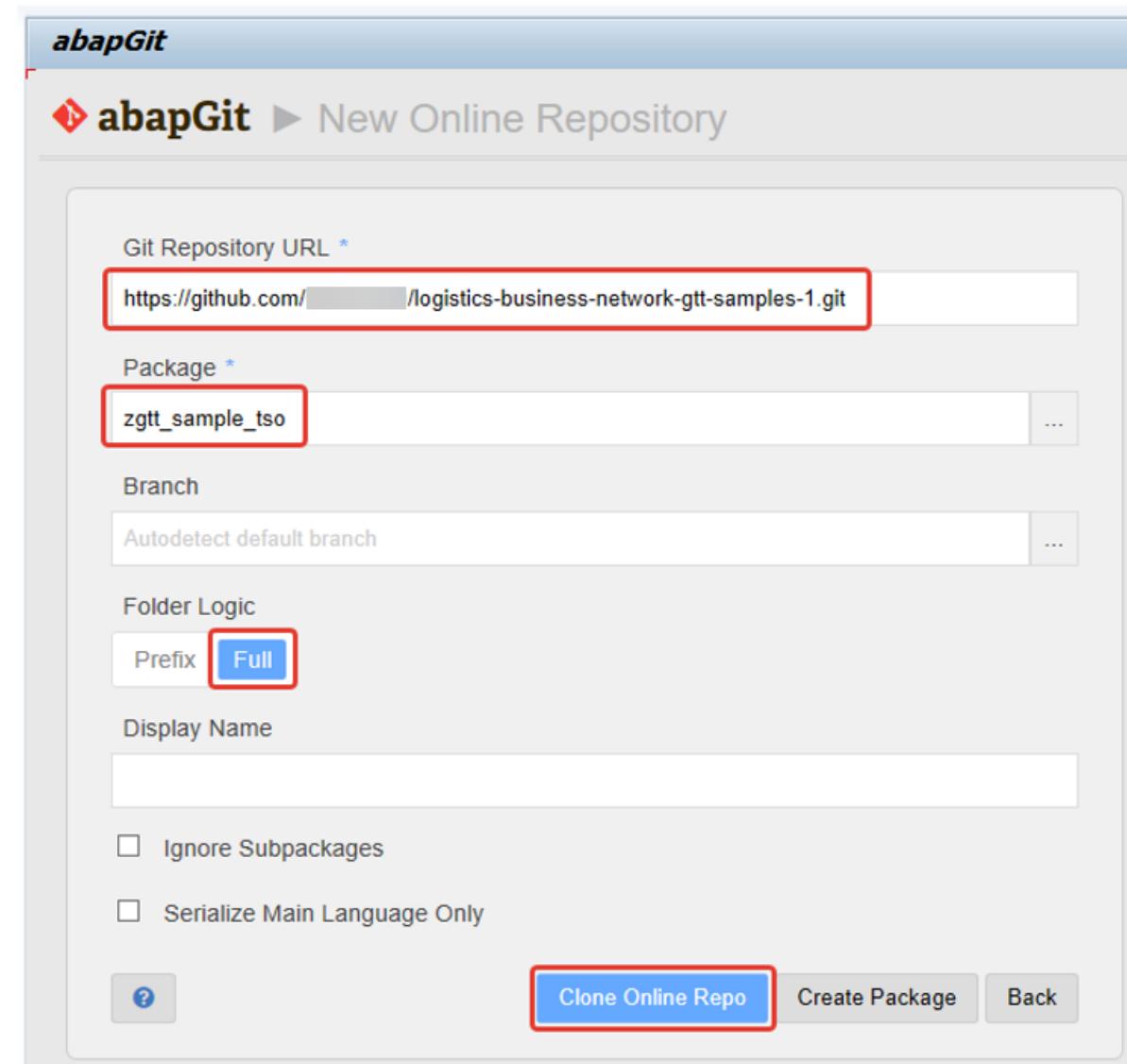
Caution:

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 a GitHub repository named "logistics-business-network-gtt-samples-1". The repository URL is <https://github.com/> /logistics-business-network-gtt-samples-1.git. The master branch is selected. A red box highlights the "Pull" button in the top navigation bar. The main area displays a table of files with columns for Type, Name, and Path. The "Type" column includes icons for AVAS, CLAS, TABL, TTYP, DTEL, DOMA, and DEV. The "Path" column lists various XML files related to ABAP GTT templates. Each file entry includes a "diff" link and a status indicator icon (green, yellow, or red).

| Type | Name | Path | diff |
|------|-------------------------------------|---|--------------------|
| | non-code and meta files | /abapgit.xml | [yellow] |
| AVAS | 0894EF4577391EEAAB910BD805B24F18 | /lbn-gtt-template-tso/abap/zsrc/0894ef4577391eeaab910bd805b24f18.avas.xml | [green] |
| CLAS | ZCL_GTT_SOFTWARE_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 | [green] [green] |
| CLAS | ZCL_IM_GTT_SOFTWARE_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 | [green] [green] |
| TABL | ZGTT_DLW_WATCH_STOP | /lbn-gtt-template-tso/abap/zsrc/zggt_dlw_watch_stop.tabl.xml | [green] |
| TTYP | ZGTT_DLW_WATCH_STOPS | /lbn-gtt-template-tso/abap/zsrc/zggt_dlw_watch_stops.ttyp.xml | [green] |
| DTEL | ZGTT_KUNABLZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_kunablz_txt.dtel.xml | [green] |
| DTEL | ZGTT_LGNUMAZ | /lbn-gtt-template-tso/abap/zsrc/zggt_lgnumaz.dtel.xml | [green] |
| DTEL | ZGTT_LGORTAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lgortaz_txt.dtel.xml | [green] |
| DTEL | ZGTT_LGTRAZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lgtraz_txt.dtel.xml | [green] |
| DTEL | ZGTT_LOCCAT | /lbn-gtt-template-tso/abap/zsrc/zggt_loccat.dtel.xml | [green] |
| DOMA | ZGTT_LOCCAT_DM | /lbn-gtt-template-tso/abap/zsrc/zggt_loccat_dm.doma.xml | [green] |
| DTEL | ZGTT_LOCID | /lbn-gtt-template-tso/abap/zsrc/zggt_locid.dtel.xml | [green] |
| DTEL | ZGTT_LOCTYPE | /lbn-gtt-template-tso/abap/zsrc/zggt_loctype.dtel.xml | [green] |
| DTEL | ZGTT_LSTELZ_TXT | /lbn-gtt-template-tso/abap/zsrc/zggt_lstelz_txt.dtel.xml | [green] |
| DTEL | ZGTT_PLN_EVT_DATETIME | /lbn-gtt-template-tso/abap/zsrc/zggt_pln_evt_datetime.dtel.xml | [green] |
| DEV | ZGTT_SAMPLE_TSO | /lbn-gtt-template-tso/abap/zsrc/package.devcl.xml /lbn-gtt-template-tso/abap/zsrc/zggt_sample_tso.abap | [green] [green] |

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.

 abapGit › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

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

Offline Projects

- Import zip
- Export zip

Reference

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

Installation

 [Improve this page](#)

Summary #

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

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

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

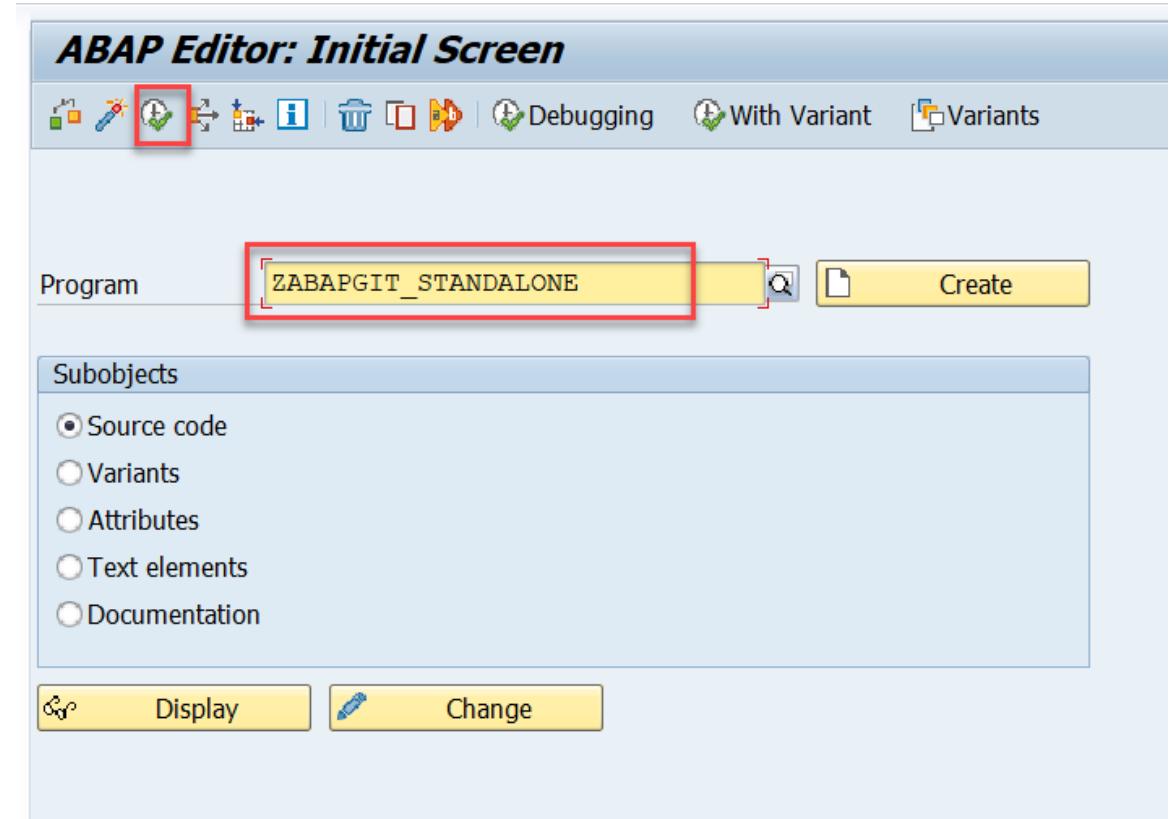
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

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

STEP 2: Download ABAP Code

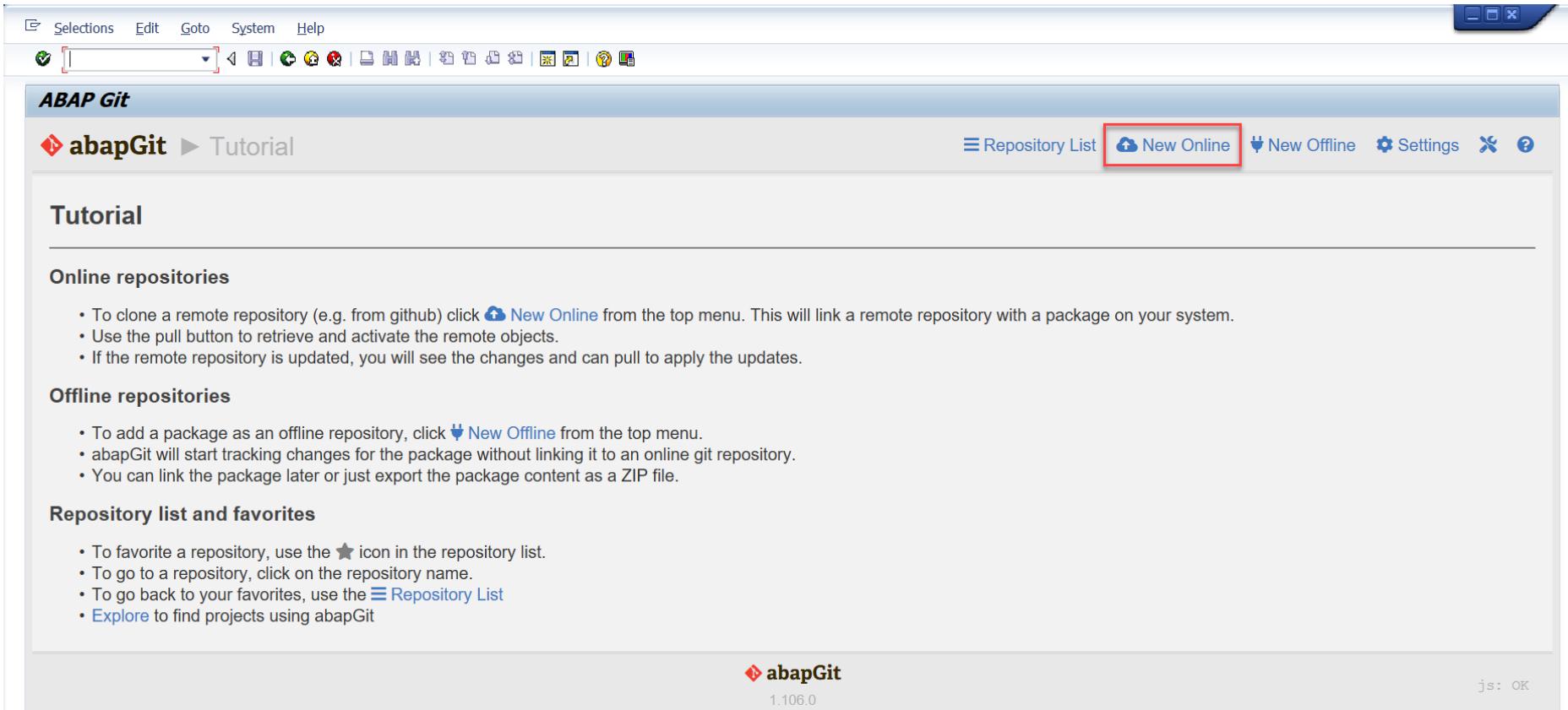
2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE.

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



STEP 2: Download ABAP Code

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



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

Online repositories

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

Offline repositories

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

Repository list and favorites

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

STEP 2: Download ABAP Code

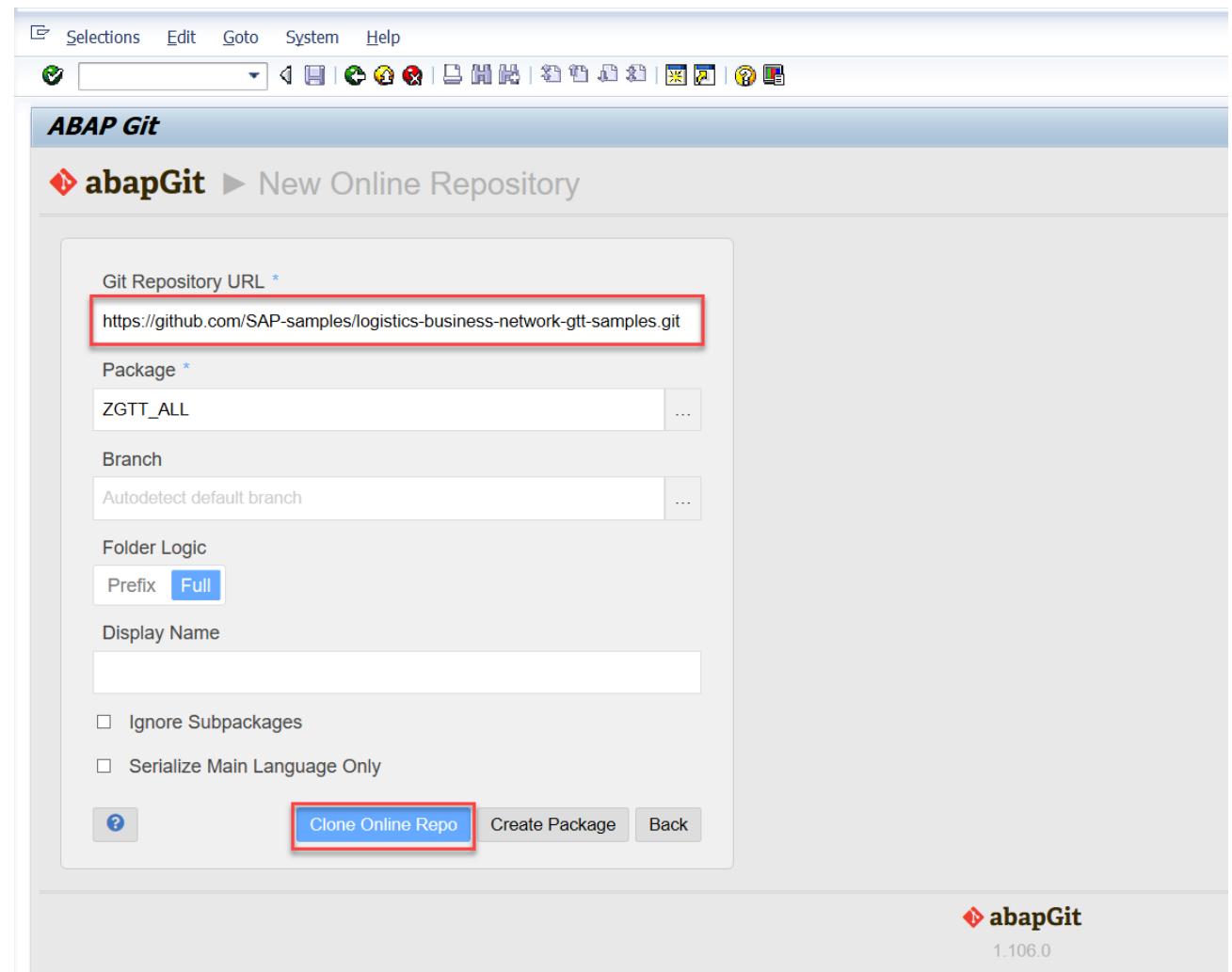
2-4: Fill in the **Git Repository URL**:

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

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

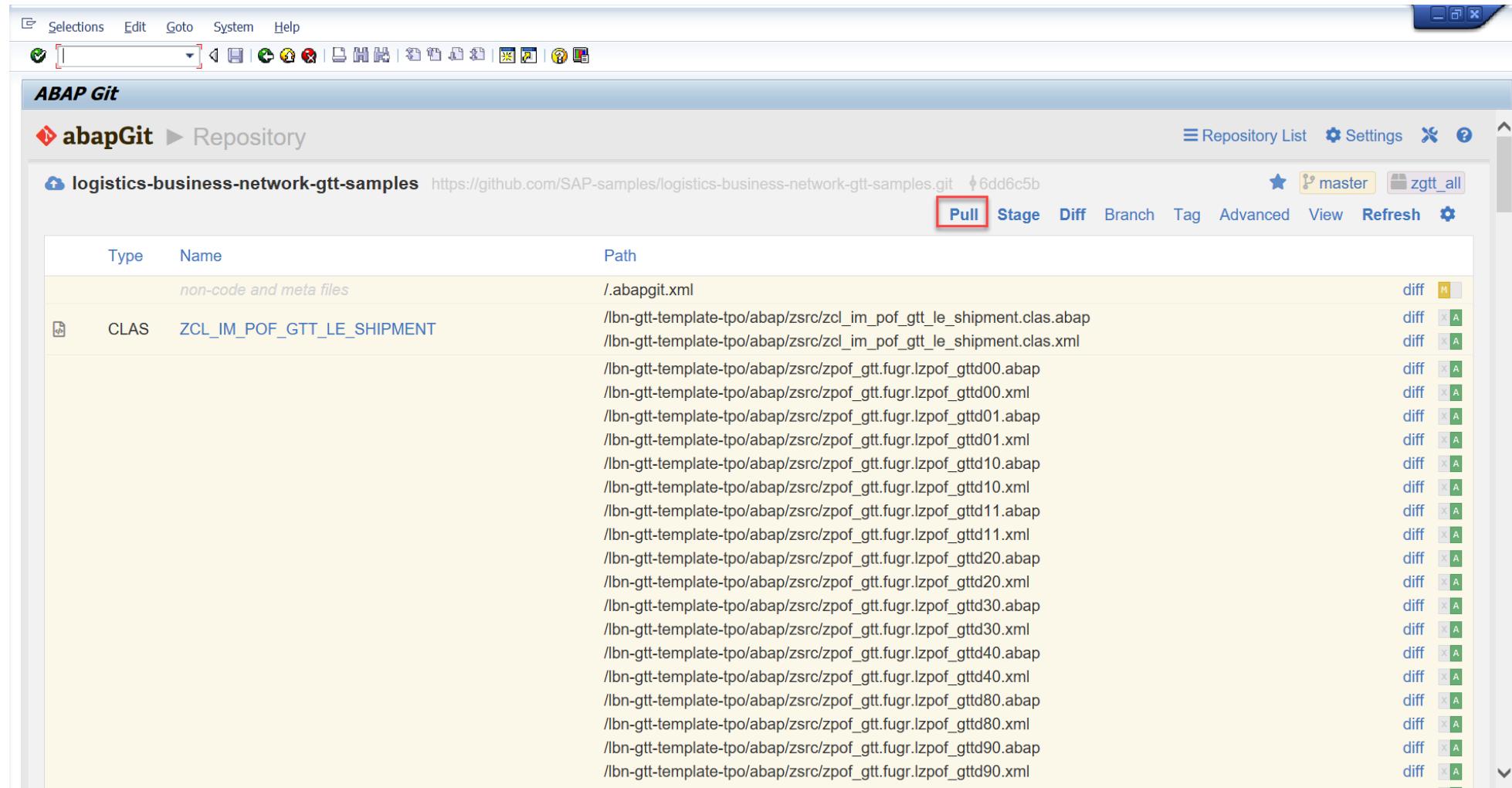
2-6: Set *Full* for **Folder Logic**

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



STEP 2: Download ABAP Code

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

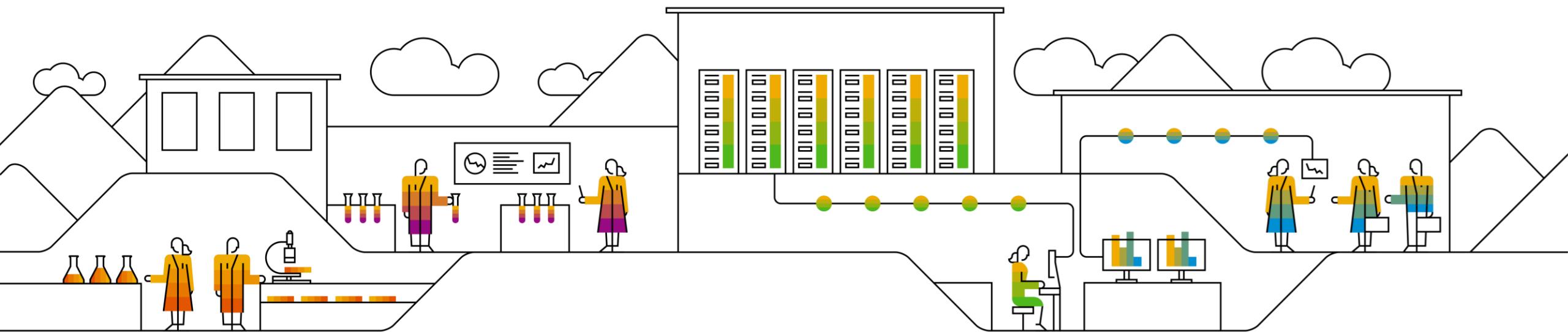


The screenshot shows the ABAP Git interface. At the top, there's a toolbar with various icons. Below it is a header bar with the title "ABAP Git" and a breadcrumb navigation: "abapGit > Repository". Underneath is a sub-header for the repository "logistics-business-network-gtt-samples" with its URL and a commit hash. To the right of the sub-header are buttons for "master" branch and "zgtt_all" tag. A navigation bar below the sub-header includes "Pull", "Stage", "Diff", "Branch", "Tag", "Advanced", "View", "Refresh", and a gear icon. The main area is a table with columns "Type", "Name", and "Path". The "Type" column shows mostly "non-code and meta files" and one entry for "CLAS". The "Name" column lists file names like ".abapgit.xml", "ZCL_IM_POF_GTT_LE_SHIPMENT", and various ABAP and XML files under the path "/bn-gtt-template-tpo/abap/zsrc/". The "Path" column shows the full file paths. To the right of each row are "diff" buttons and small colored boxes indicating changes (yellow for M, green for A, blue for D).

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

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 shows two SAP screens side-by-side. On the left is the 'Change View "Define Application Object Types": Details' screen. It has fields for 'Bus. Proc. Type' (ESC_PURORD) and 'Appl. Obj. Type' (ZPOF_GTT_AC_HD). The 'Appl. Obj. Type' field is highlighted with a red box. Below these are tabs for 'General Data', 'Control Tables', 'Object Identification', 'Global Track & Trace Relevance', and 'Parameter Setup'. Under 'General Data', there are sections for 'Sequencing / Destination' (Seq. No. 10, CI for GTT ZGTTPOFAC) and 'Business Object Reference' (Object Type BUS2012 PurchaseOrder, BO Setup Fnct.). On the right is the 'Model Details' screen for 'pof' (Active). It shows 'Purchase Order Fulfillment' with 'Namespace: com.lbngtsamples.gtt.app.pof' and 'Correlation Level: 4'. The 'IDOC Integration' tab is selected and highlighted with a red box. It displays 'Tracked Process: PurchaseOrder' and 'Integration Switch: ON'. The 'Tracked Process Mapping' section shows 'ERP Object Type: Others' and 'Application Object Type: ZPOF_GTT_AC_HD' (highlighted with a red box). The 'Tracked Process / Events (1)' table lists 'Name' (Tracked Process, PurchaseOrderEvent), 'IDOC' (E1EHPCT, E1EHPAO), and 'Event Code' (E1EHPAO). The 'Fields' table maps fields from the tracked process to IDOC segments:

| Field | IDOC Segment | IDOC Field |
|---------------------|--------------|-------------------------|
| purchaseOrderNo | E1EHPCT | YN_PO_NUMBER |
| supplierId | E1EHPCT | YN_PO_SUPPLIER_ID |
| plannedDeliveryDate | E1EHPCT | YN_PO_DELIVERY_DATE |
| netValue | E1EHPCT | YN_PO_NET_VALUE |
| currency | E1EHPCT | YN_PO_CURRENCY |
| incotermsVersion | E1EHPCT | YN_PO_INCOTERMS_VERSION |
| incoterms | E1EHPCT | 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 "Appl. Obj. Type" field, the value "ZPOF_GTT_AC_HD" is selected, with the description "Purchase Order Head for Procurement Visibility (GTT) - Acceptance" highlighted. In the "Tracking ID Setup" section, 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". The "OK" button is visible at the bottom of the modal.

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

You can assign customization function models to the following extractor 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 Procces |
| 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 navigation path is: Implementation Guide > Existing BC Sets > Define Application Interface > Define Business Process Types. The 'Define Business Process Types' node is highlighted with a yellow background.

Display View "Define Available Application Tables": Overview

This table displays the structure and DDIC definitions for various application tables. The columns include:

| Structure/Table | DDIC Definition | DB Struc. Name | Bus. ... | Updt Fld Name | No C... | Inser... | Upda... | Delet... | Key St... | Ke |
|-----------------------------|-----------------|----------------|-------------------------------------|---------------|---------|----------|---------|----------|-----------|----|
| CONDITIONS_NEW | KOMV | | <input type="checkbox"/> | UPDKZ | I | U | D | 0 | 0 | |
| CONDITIONS_OLD | 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 | UEKKN | EKKN | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| PO_ACCOUNT_ASSIGNMENT_OLD | UEKKN | EKKN | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| PO_ITEM_NUMBER | EKBES | EKES | <input type="checkbox"/> | | | | | 0 | 0 | |
| PO_SCHED_LINE_ITEM_NEW | UEKET | EKET | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| PO_SCHED_LINE_ITEM_OLD | UEKET | EKET | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| PURCHASE_ITEM_NEW | UEKPO | EKPO | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| PURCHASE_ITEM_OLD | UEKPO | 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 | UEKPV | EKPV | <input type="checkbox"/> | | | | | 0 | 0 | |
| VENDOR_CONFIRMATION_NEW | UEKES | EKES | <input type="checkbox"/> | KZ | I | U | D | 0 | 0 | |
| VENDOR_CONFIRMATION_OLD | UEKES | 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.

Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_REL

This window shows the source code for the function module *ZPOF_GTT_OTE_PO_ITEM_REL*. The code handles various input parameters like *lt_app_objects*, *ls_bapiret*, and *is_definition*, and processes them through TRY-CATCH blocks to handle exceptions like *cx_udm_message*. It also includes EXPORTING and IMPORTING sections for *io_udm_message*, *iv_appsystypes*, and *es_bapiret*.

```
DATA: lt_app_objects TYPE trxas_apobj_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_tabledesc-po_item_new
      masterstab = lif_pof_constants->cs_tabledesc-po_header_new
      io_bo_factory = NEW lcl_factory_po_item( )
      iv_appsystypes = i_appsystypes
      is_app_obj_types = i_app_obj_types
      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

This window shows the source code for the include *LZPOF_GTTD20*. It contains several methods, with the *lif_bo_reader-check_relevance* method being the primary focus. This method performs basic checks on the main table, checks if only one PO type is relevant for GTT, and handles various update modes (insert, update, undefined) by comparing new and old values.

```
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_pof_constants->cs_condition-false.

  " is_app_object-maintabdef = lif_pof_constants->cs_tabledesc-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_type( ir_expo = is_app_object-maintabref ) = abap_true AND
  is_object_changed( is_app_object = is_app_object ) = 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_GRANTED.
```

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 image shows two screenshots of the SAP ABAP development environment. The left screenshot is the 'Function Builder: Display ZPOF_GTT_OTE_PO_ITEM_TID' window, showing the source code for the function module. The right screenshot is the 'ABAP Editor: Display Include LZPOF_GTTD20' window, showing the source code for the include module. Both windows show the code in a standard ABAP syntax editor with syntax highlighting and code completion.

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

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

  TRY.
    lcl_ef_performer->get_track_id_data(
      EXPORTING
        is_definition = VALUE #( maintab = lif_pof_constants->cs_tableddef-po_item_new
                                mastertab = lif_pof_constants->cs_tableddef-po_header_new )
      IMPORTING
        io_bo_factory = NEW lcl_factory_po_item()
        i_appsyst
        is_app_obj_types = i_app_obj_types
        it_all_appl_tables = i_all_appl_tables
        it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
        it_app_objects = i_app_objects
      CATCH cx_udm_message INTO lo_udm_message.
      lcl_tools->get_errors_log(
        EXPORTING
          ic_udm_message = lo_udm_message
          iv_appsyst = i_appsyst
        IMPORTING
          es_bapiret = ls_bapiret ).

      " add error message
      APPEND ls_bapiret TO e_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 table_determination_error.
      ENDCASE.
    ENDTRY.
```

ABAP Editor: Display Include LZPOF_GTTD20

```
Include LZPOF_GTTD20 Active
  METHOD lif_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_tableddef-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
        ) ).

      ENDIF.
```

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 for a model named "pof" (Active). The top navigation bar includes tabs for 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. The status bar indicates the namespace is com.lbngttsamples.gtt.app.pof and the correlation level is 4.

Below the tabs, there are filters for Tracked Process (set to PurchaseOrder) and an Integration Switch (set to ON). The main content area is divided into sections: Tracked Process Mapping, Tracked Process / Events (1), and Fields.

Tracked Process Mapping: Shows the mapping between an ERP Object Type (Others) and an Application Object Type (ZPOF_GTT_AC_HD).

Tracked Process / Events (1): A table showing the mapping of a Tracked Process (PurchaseOrderEvent) to an IDOC (E1EHPAO) and an Event Code.

Fields: A table showing the detailed field mappings from the Tracked Process to the IDOC fields.

| 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

The screenshot shows the SAP ABAP Editor interface with the function module ZPOF_GTT_OTE_PO_ITEM selected. The code editor displays the following logic:

```
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_ctrl_tabs = i_app_type_ctrl_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.
```

The screenshot shows the SAP ABAP Editor interface with the include module LZPOF_GTTD20 selected. The code editor displays the following logic:

```
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_DATEETIME* 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 *LOC_ID2* 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 image 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 function module ZPOF_GTT_EE_PO_ITEM. The code implements planned events for purchase order items, utilizing the LCL framework. It includes sections for DATA declarations, TRY/ENDTRY blocks, and various EXPORTING, IMPORTING, and CHANGING parameters. On the right, the 'ABAP Editor: Display Include LZPOF_GTTD30' window shows the source code for the include LZPOF_GTTD30. This include defines methods for the LCL class LCL_PE_FILLER_PO_ITEM, specifically the lif_pe_filler~get_planned_events method, which handles planned events like confirmation and goods receipt.

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

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

CLEAR e_logtable[].

TRY.
  lcl_ef_performer->get_planned_events(
    EXPORTING
      is_definition      = VALUE #(
        maintab   = lif_pof_constants->cs_tabledef-po_item_new
        mastertab = lif_pof_constants->cs_tabledef-po_header_new )
    IMPORTING
      io_factory       = NEW lcl_factory_po_item()
      iv_appsyst      = i_appsyst
      is_app_obj_types = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_type_cntl_tabs = i_app_type_cntl_tabs
      it_app_objects   = i_app_objects
    CHANGING
      ct_expeventdata  = e_expeventdata[]
      ct_measrmntdata = e_measrmntdata[]
      ct_infodata     = e_infodata[])
  CATCH cx_udm_message INTO lo_udm_message.
    lcl_tools->get_errors_log(
      EXPORTING
        io_udm_message = lo_udm_message
        iv_appsyst    = i_appsyst
      IMPORTING
        es_bapiret    = ls_bapiret).
    " add error message
    APPEND ls_bapiret TO e_logtable.

    " throw corresponding exception
    CASE lo_udm_message->txid.
      WHEN lif_ef_constants->cs_errors-stop_processing.
        RAISE stop_processing.
      WHEN lif_ef_constants->cs_errors-table_determination.
        RAISE table_determination_error.
    ENDCASE.
  ENDTRY.
ENDFUNCTION.

ABAP Editor: Display Include LZPOF_GTTD30
Include LZPOF_GTTD30 Active

METHOD lif_pe_filler~get_planned_events.
  add_confirmation_event(
    EXPORTING
      is_app_objects = is_app_objects
    CHANGING
      ct_expeventdata = ct_expeventdata ).

  add_goods_receipt_event(
    EXPORTING
      is_app_objects = is_app_objects
    CHANGING
      ct_expeventdata = ct_expeventdata ).

ENDMETHOD.

ENDCLASS.
```

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: ENDTRY.
100: 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 Definitions
- 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_CONTRO
 - IS_APPROPRIATE_CONF_TYPE
- Attributes

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:     txrid = 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:   " QUANTITY
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:   " CONFIRMATION TYPE
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.
```

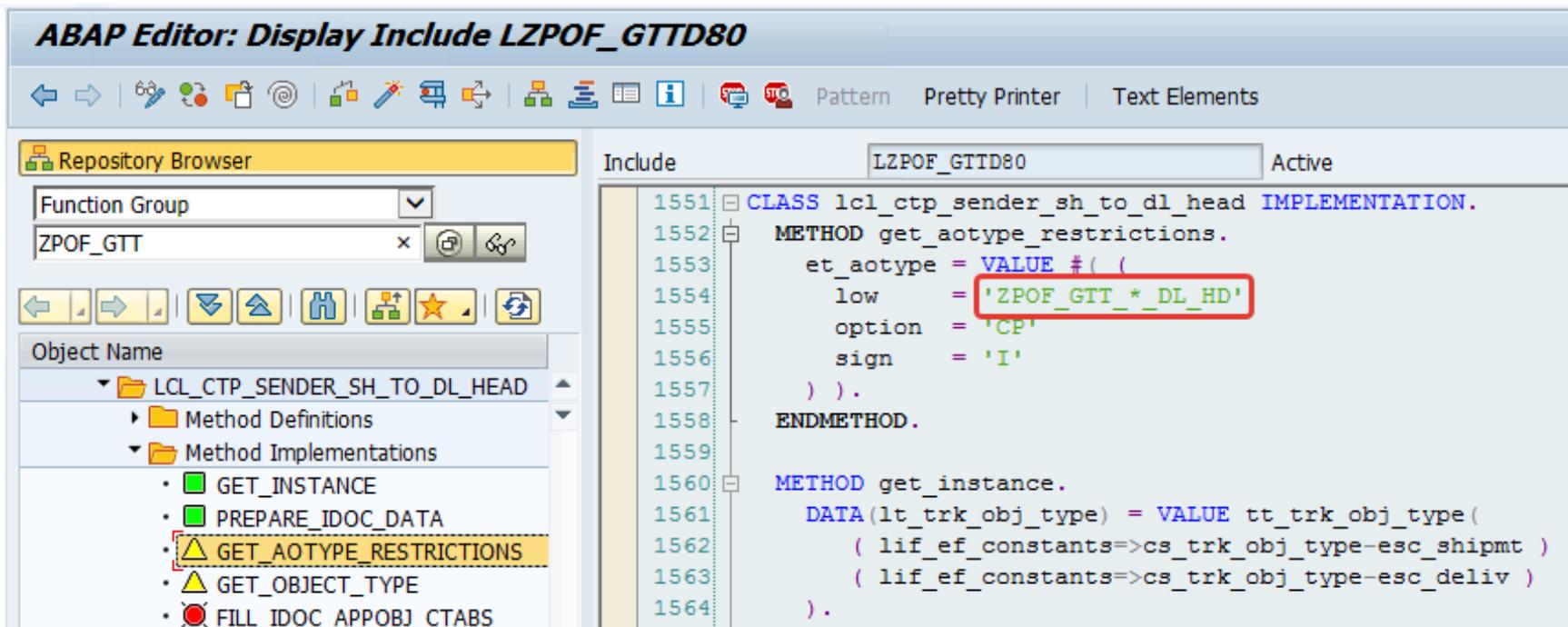
11: Enhancement Codes for Cross-process Tracking

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



The screenshot shows the ABAP Editor with the title "ABAP Editor: Display Include LZPOF_GTTD80". The left pane is the Repository Browser, showing a Function Group named "ZPOF_GTT" and a list of objects under "LCL_CTP_SENDER_SH_TO_DL_HEAD" including "GET_INSTANCE", "PREPARE_IDOC_DATA", "GET_AOTYPE_RESTRICTIONS" (highlighted in yellow), "GET_OBJECT_TYPE", and "FILL_IDOC_APPOBJ_CTABS". The right pane displays the source code for the LZPOF_GTTD80 include. The code defines a class implementation for "lcl_ctp_sender_sh_to_dl_head" with two methods: "get_aotype_restrictions" and "get_instance". The "get_aotype_restrictions" method contains a line of code where the value of "et_aotype" is set to "ZPOF_GTT_*_DL_HD", which is highlighted with a red rectangle.

```
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.