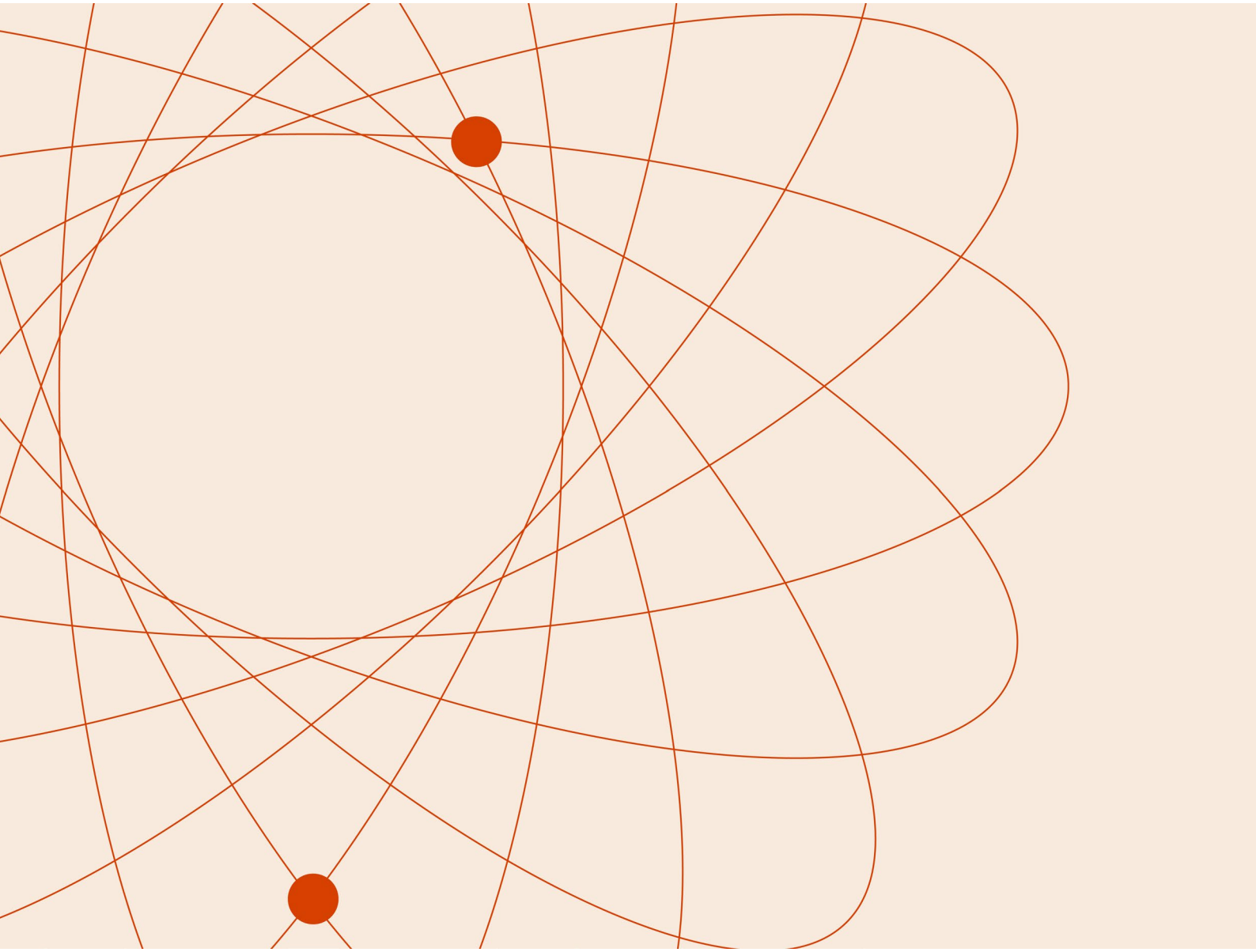
ONESOURCE Indirect Tax Integration for SAP

INSTALLATION AND PROGRAMMERS GUIDE

PRODUCT VERSION 6.9.X.X

JUNE/2025



Document History

Version Number	Version Date	Summary
1.0	June 2025	First release of ONESOURCE Indirect Tax Integration Configuration Guide for ONESOURCE Tables 6.9.x.x

Contents

Introduction

Welcome to ONESOURCE Indirect Tax Integration for SAP

Corporations using SAP as their Enterprise Resource Planning (ERP) system can simplify their worldwide Indirect Tax requirements by implementing ONESOURCE Indirect Tax Suite. The suite includes Determination, Integration for SAP, Reporting, and Compliance. The benefits provided by ONESOURCE Indirect Tax Suite are:

- Fast, accurate sales, use, consumer's use tax, and VAT results.
- Monthly tax rate and rules updates for over 175 countries.
- Integrated tax calculation with SAP minimizing user decisions and tax errors.
- Removal of the need to change SAP tax codes each time a rate/rule changes, eliminating business interruptions, and running out of tax codes in SAP.
- Complete an audit database from which you can generate both standard and custom reports as well as returns.

ONESOURCE Indirect Tax Integration for SAP 6 provides an interface designed, built, and maintained by Thomson Reuters. This global tax integration solution is designed from the ground up, with integration pointing into SAP ECC and S/4HANA application modules as desired. It consists of a data collector, tax interface, and return process of tax results to the calling application with G/L integration in support of downstream SAP processes such as standard VAT reports and returns processing. It makes use of the SOAP (Simple Object Access Protocol) provided by SAP to communicate with ONESOURCE Indirect Tax Determination. This Integration enables worldwide tax calculations, including VAT/ GST, US Sales and Use Tax, Brazil, India and other country-specific taxation.

The interface is entirely built within the SAP Development Workbench, including a user menu for all interface related configurations, setups, and reports. The interface has a field mapping solution allowing a Tax Business Analyst to map SAP data to Determination and vice versa via a customization table, eliminating most of the user-exit coding of the past. Tax calculation logs can be accessed via a transaction with a search function from within SAP greatly simplifying tax setup, analysis, and troubleshooting.

Audience

If you are responsible for overseeing setting up ONESOURCE Indirect Tax Integration for SAP, you will need to coordinate help from the following people:

- SAP Business Systems Analyst
- SAP Configuration Consultant
- SAP Technical Resource (ABAP Programmer)
- SAP Security Contact
- Tax Professional

Make this guide available to each of these contributors to ensure you have a successful installation.

Prerequisites

For a seamless and successful deployment of Integration for SAP we highly recommend that you follow the instructions in the guides listed below in order:

- Configuration Guide for ONESOURCE Tables
- Configuration for SAP Tables
- Installation and Programmers Information
- Installation Instructions (User Guide)
- Special Functions

When working on Integration for SAP you must have knowledge of the SAP tax features, covering all aspects of FI, MM, and SD and have spent time either as an expert configurator or consultant in these areas. Because the setup of tax integration with ONESOURCE Indirect Tax also includes technical work in the ABAP Workbench, such as data dictionary changes and ABAP coding, you must be able to understand and interpret these changes as well. We recommend that you assemble a team to implement this product because it requires both functional and technical input. Your team should include someone who understands business requirements and processes, as well as someone who can implement the required software changes.

Please take the following into account before setting up the Integration for SAP:

- This guide assumes a fresh install of the Integration for SAP. Customers who are upgrading from a prior 5.x version of Integration should contact Thomson Reuters Indirect Tax.
- Minimum SAP system version must be ECC 6.0, EHP 6. Please see tested platforms by Thomson Reuters in Platform Information section.
- Procure a subscription to ONESOURCE Indirect Tax Determination
- It is assumed that the persons who install, configure, and use the tax interface in SAP have some basic understanding of the overall ONESOURCE Indirect Tax Suite of products and how they interact with each other.

Resources

The following table describes which resources are available:

RESOURCE	DESCRIPTION
Customer Support	Look for answers in the Knowledge Base, or open a support ticket.
Installation Instructions (User Guide)	This guide provides an overview of the architecture and business processes as they relate to Sales and Use tax, as well as VAT scenarios in FI, SD, and MM. The target audiences are the Business Systems Analysts, Consultants, and Tax Professionals who set up the tax processes in SAP.
Installation and Programmers Guide	This guide instructs you how to install the Integration for SAP. The target audiences are the Basis personnel who will process the application of the transports to the SAP system and the ABAP programmers who will perform the required INCLUDE statements within the user exits and other coding blocks. There is also discussion in this manual for the ABAP programmer regarding customization logic and how custom additions to the programs should be added to the system if needed in the future. This guide describes the supported combinations of operating systems, databases, and application servers/web containers. Refer to support lifecycle for SAP for the end-of-life dates for ONESOURCE Indirect Tax Integrations for SAP. Consult this guide to see which combinations of software Thomson Reuters tests with Integrations.
Configuration Guide SAP tables	This guide instructs you how to configure and set up SAP tables and processes to enable tax calculations to meet your unique requirements.
Configuration Guide ONESOURCE tables	This guide instructs you how to configure and set up ONESOURCE Indirect Tax tables and processes to enable tax calculations to meet your unique requirements.
Configuration Guide for Special Functions	This guide instructs you how to configure and set up SAP and Integration tables and processes to enable tax calculations to meet your unique requirements for special functions within SAP, including Plants Abroad, Cash Discounts, Deferred Taxes, Service Entry Sheets, Settlement Management, and Tax-only invoices.
Vendor Charged Tax (VCT) Guide	This guide describes functionality for integrating Indirect Tax Enterprise Cloud VCT calculations with SAP Global Next transactions. This functionality is for the United States and Canada only

Support Protocol

The ONESOURCE Indirect Tax Integration for SAP is built, maintained, and owned by Thomson Reuters. We follow SAP best practices, development standards, and strive to minimize the impact this solution will have on your SAP environment. With any 3rd party Add-On in SAP, the vendor providing the solution is responsible for support of that Add-On. In the case of an issue with the ONESOURCE Indirect Tax Integration for SAP please follow these simple steps to open a support ticket with Thomson Reuters:

1. Identify the potential issue and gather all necessary facts (log files, scenarios, configurations, screen prints).
2. Provide system environment information such as your SAP Version (ECC or S/4 HANA), EHP/ SP or FPS level, as well as the Integration version.
3. Another month later, you imported your March source file. The March source file contains both the January and February transactions, in addition to the March transactions. Again, ONESOURCE Tax Information Reporting needs to know that the January and February transactions were included.
4. Open a support ticket with Indirect Tax athttps://tax.thomsonreuters.com/support/ONESOURCE/indirect- tax/

INTEGRATION OVERVIEW

WHAT IS INTEGRATION FOR SAP VERSION 6?

Integration for SAP version 6 is an integration designed to significantly improve the user experience and dynamically expand current and future system capabilities for indirect tax calculations, reporting, and data mapping. This integration is based on ABAP coding directly within ONESOURCE Indirect Tax's SAP registered namespace which will provide far greater flexibility to meet challenging and ever-changing global tax requirements and simplifies US and Canadian tax.

The integration is designed with the global customer in mind and closely follows SAP's structure and logic for global VAT processing. Users can take advantage of many standard SAP functions that are only made available with a non-jurisdiction-based configuration. As future SAP features and functions become available to customers, this integration is able to easily adapt and support tax calculation calls without the need for costly integration efforts.

Tax code usage within the integration provides static assignment of tax codes based on downstream reporting and compliance needs without the requirement of assigning a tax rate, eliminating setup of different tax codes when tax rates change. This design allows the use of SAP standard reports and functions and avoids running out of possible tax code assignments. Support of a large number of global taxing authorities and historical rate changes are not an issue.

Users can now take advantage of standard VAT reports, plants abroad configuration, and many other standard SAP features that were a challenge or impossible with the prior jurisdiction code based interface. Additional functionalities have been provided to remove limitations on the summarization of taxing authorities, number of fields mapped to Determination requests and responses, and other key elements required for meeting current and future global tax challenges.

Below is a reference table of some of the many features and functions that are available with this Integration design. Throughout the guides we will discuss in more detail each of the new features listed and further explain what opportunities are now available to the system user.

QUICK VIEW OF FEATURES

FUNCTION	DESCRIPTION
All configuration within SAP tables	All configuration is now located in a separate ONESOURCE Indirect Tax SAP partner namespace of /IDT/.
ONESOURCE user menu within SAP	A ONESOURCE menu allows access to Integration configuration tables, functions, logs, and reports directly within SAP.
Authorization objects for users and general administrator	SAP transaction codes now allow for access control to setup and configuration settings. Access to tax calculation logs and reports are secured by IDT authorization objects. This allows access control based on a company's own security protocol.
Flexible field and address mapping	Field mapping is put in place to allow users to map request attributes and response fields from Determination directly with the use of tables. This simplifies the data mapping process and reduces the requirement of extensive ABAP coding. Major performance enhancements have been addressed with the addition of dynamically generated internal code when a line is changed or added.
Tax data storage for all business transactions	A table has been created to store data from Determination in SAP for downstream processes for all business processes calculating tax. Tax details are stored for SD, MM, and FI transactions.
Tax code and account assignment using Determination ERP_CODE_MAPPING	Tax code and account key logic used by SAP to assign the G/L account number are established via Determination's ERP_CODE_MAPPING function and the use of specific condition logic to assign the correct codes and G/L account based on reporting needs.
Log report and configurable logging methodology	Multi-level log configuration is now available within SAP and can be maintained by user, transaction code, and other conditions. This flexible log viewer is provided for searching and displaying tax calculation logs to quickly troubleshoot taxing issues.
Log file delete/archive process	Menu option has been added to allow management of the log entries by deletion or download to zip file.

FUNCTION	DESCRIPTION
Developer log report and configuration table	An additional log for developers has also been provided which gives the developer additional views of the background processing with special features that can be turned on to aid in program debugging and system analysis, such as routes and journeys that were activated in the call and other analysis functions.
Developer log delete function	Ability to delete old developer logs that were created for analysis and debug purposes.
Audit report	Audit report tracks and identifies any tax entries that have been posted to SAP General Ledger but have not updated the tax audit database. Report can also submit missing items for repost to audit.
Flexible use between modules of SAP i.e.: SD, MM, AR, AP, FI	Ability to configure some or all modules as needed as well as combine usage of ONESOURCE Indirect Tax and SAP native tax calculation methods by module.
Removal of limit on number of tax codes	Use of multiple tax codes according to their taxing authority and reporting needs but still retain the rate structure changes within Determination without having to create additional codes for rate changes.
Removal of jurisdiction code method	Taxability now controlled by the address of the taxable entity and not by the assignment of a jurisdictin code.
Ability for attribute mapping	Ability to map custom attributes at both header and again at line-item level from SAP to the Determination request and the response from Determination back to SAP.
Dynamic creation of tax lines	Dynamic process resulting in a possible number of 99 separately returned taxing authorities to the line-item tax calculation. (SAP limited to 99)
Separate tax lines for freight	Ability to have separate tax authority lines on the conditions tab of documents for freight handling separate from the related expense line.
Authority text now displays on conditions tab	The name of the tax authority now displays on the conditions tab of the documents.

FUNCTION	DESCRIPTION
Per document calculation in all modules	Per document calculation in SD, MM, and FI allows for proper calculation of max tax scenarios common on US Sales Tax calculations.
Use of calculation schema in Purchasing	Use of the calculation schema in Purchasing provides document level calculation for max tax and other conditions as required.
Improved cash discount processing	Improved transaction logic for calculation of tax on cash discounts taken at time of payment
Plants Abroad functionality	Ability to use Plants Abroad configuration with SAP standard configuration.
Tax Codes can be marked as non-tax relevant	Ability to mark certain tax codes as not relevant and prevent a call to Determination if desired.
Ability to use ILM to archive Global Next tables	Support for SAP ILM tool to archive Global Next tables /IDT/D_AUDIT_REC, /IDT/D_AUDIT_STA, /IDT/D_TAX_DATA.
SOAP interface	The integration is built on SAP's Internet Communications Framework using SOAP and Determination's WSDL/XSD definition.
Down Payments processing	Ability to utilize the down payment process in SAP for both customer side and vendor side transaction including the request for down payment, down payment application, and transfer of the down payment to the customer/vendor open A/R and A/P accounts, from FI and SD modules
Tax based on entry of gross amount	FB00 entry to set tax based on entry of the gross amount has been reviewed and issues adjusted with calculation and return to audit.
Evaluated Receipt Settlement process	Tax calculation on MRRL and MRRS for creation of invoice in LIV based on the goods receipt transaction is now available using standard SAP processes.
Service Entry Sheets	Using Service Entry Sheets on purchase orders is now supported for tax calculation based on the Service Entry Sheet line-items and transaction ML81N.
Deferred Tax Transfer	Ability to run the RFUMSV25 and RFUMSV50 programs with SAP standard configuration.

FUNCTION	DESCRIPTION
Brazil Enablement	Functionality that allows the SAP Brazil Country version tables and logic to work with our Determination Tax Engine in a non-jurisdictional environment.
Unplanned Delivery Costs in MIRO	Tax call and logic supporting calculations on unplanned delivery charges on MIRO invoices entry.
Multiple Account Assignments to a Line Item in Purchasing and MIRO Invoice Entry	Ability to allocate a line-item to multiple cost object assignment and calculate tax based on ship to address of the various multi- assignment lines including ability to override the address when needed at time of MIRO invoice entry.
Consignment and Pipeline settlements	Use of transaction MRKO for transfer of goods for consignment and pipeline.
India Enablement	Functionality that allows the SAP India Country version tables and logic to work with our Determination Tax Engine in a non- jurisdictional environment.
Parked Document Processes	Functionality for tax calls and log entries for parked document processes and transactions were reviewed and updated.
Tax Data Download Report	Tax data download report has been written to replace the US Tax report in the integration reporting menu. The download report allows data to be extracted for Tax Data Table for improved and flexible use of transaction data for analysis and downstream reporting needs covering US, Canada and Global transactions. In S/4HANA, CDS views are provided for the table for enhanced reporting
BAPI support and custom utility for BAPI	Support for a list of BAPI including utility for use with BAPI_ACC_DOCUMENT_POST
Tax Only Invoices	Ability to post tax adjustments directly to tax General Ledger accounts, using standard SAP transactions
S/4HANA Settlement Management	Functionality to take advantage of SAP S/4 HANA Settlement Management feature for processing customer and supplier rebates.
Vendor Charged Tax Verification (VCT-V)	Functionality for integrating Indirect Tax Enterprise Cloud VCT-V calculations with SAP Global Next transactions. This functionality is released the United States and Canada transactions.

BENEFITS OF USING ONESOURCE INDIRECT TAX

Enabling Determination to calculate tax for your SAP transactions results in the following key benefits:

Seamless Integration

Once integrated, your tax professionals can continue to use SAP functionality without needing to learn new technology or processes. Determination automatically obtains a complete set of data elements necessary to perform the appropriate tax calculations, and then returns the results to SAP all without the need for manual intervention.

Accurate Tax Calculation

- Determination provides these features:
- Integrates worldwide tax calculation
- Enables global visibility and real-time transactions
- Gives control of the data and decision-making to the tax professional
- Eliminates or minimizes IT involvement as tax authorities, rates, and rules change
- Provides a scalable, maintainable enterprise solution

Using ONESOURCE Indirect Tax as your global transaction tax management solution reduces costs, increases accuracy, and provides the flexibility you need to adapt to an ever-changing business taxation environment.

Auditing/ Reporting

Tax calculations that are processed using Determination are stored in Determination audit tables. From this audit data, you can generate standard or customized reports based on any user-defined criteria. For example, you can create tax-by-authority reports or any other desired reports needed to satisfy your company's needs.

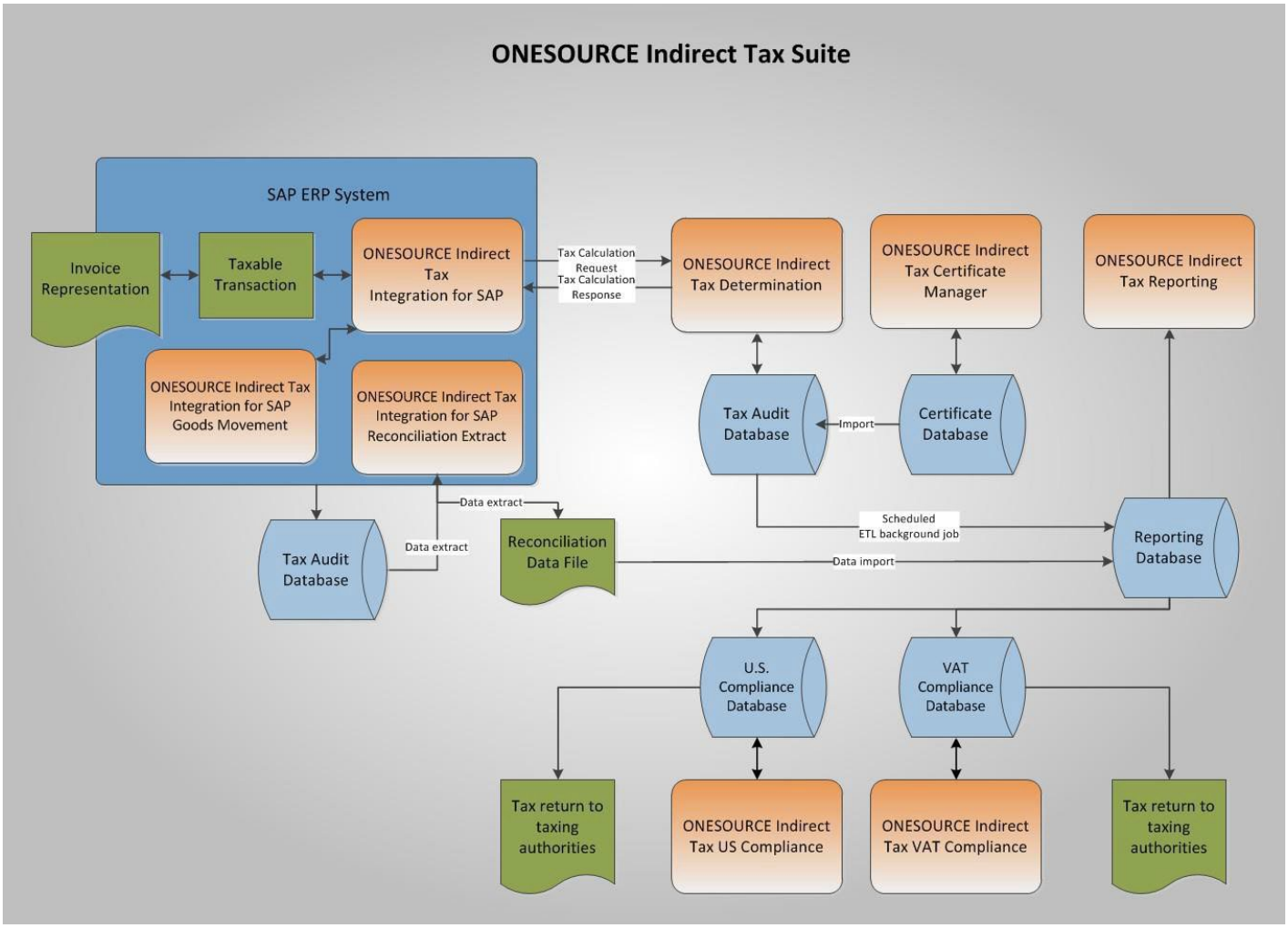
USER PROCESSES THAT TRIGGER DETERMINATION TAX CALCULATIONS

The Installation and Configuration Guides provide procedures needed to install and configure Integration to enable end-to-end tax calculation with Determination.

Once Determination is fully integrated with SAP, you can automatically generate correct tax results during the following actions:

- Procure to Pay; Purchase Orders, Logistic Invoice Verification with posting to AP
- Order to Cash; Sales Order to Billing with posting to AR
- Processing FI module adjusting entries

ARCHITECTURE AND DESIGN OVERVIEW



THE ONESOURCE INDIRECT TAX SUITE

ONESOURCE Indirect Tax Suite of products is made up of the following components:

Integration:

ONESOURCE Indirect Tax Integration seamlessly connects your ERP system to Determination for tax calculations and appropriate return of tax results to the ERP for invoice printing and posting to the General Ledger. Integrations are developed and maintained in-house by a team of Thomson Reuters Business Systems Analysts, Developers, and Quality Assurance employees providing the most advanced tax engine determination capability and compliance returns processing globally. Our solution can be fully assimilated into any of your existing businesses, e-commerce, or financial systems using our open integration architecture. Tax calculation calls can be easily inserted into existing system workflows and processes to deliver real-time or batch solutions with accurate tax results.

Determination:

ONESOURCE Indirect Tax Determination enables companies to consolidate their global tax policy in one central location. All enterprise-wide applications can use a single scalable instance of Determination and still deliver business-specific tax policy across multiple-business systems. Fully integrated to all your financial applications, Determination enables the passing of transaction data from the financial system to the tax engine, and returns transaction taxes in real time for fast, reliable, and accurate indirect tax determination. We offer fully supported standard Oracle and SAP integrations, as well as custom integrations via our tax calculation web service.

Tax Certificate Manager:

ONESOURCE Indirect Tax Certificate Manager is a solution for the precise tracking, validating, and governing of exemption certificates. As part of ONESOURCE, it provides integration to our ONESOURCE Indirect Tax Determination software that allows for the export of customers and exemption certificates. ONESOURCE Indirect Tax Certificate Manager improves efficiency in all aspects of the burdensome exemption certificate lifecycle by reducing operating costs, mitigating risk, and increasing accuracy. ONESOURCE Indirect Tax Certificate Manager reduces audit exposure and assessments while empowering you with full control of the exemption certificate process to maintain Sarbanes-Oxley compliance.

Reporting:

ONESOURCE Indirect Tax Reporting software provides fast, accurate, and flexible reporting that's fully integrated with our ONESOURCE Indirect Tax global software suite to support your global compliance, reconciliation, and data analysis processes. An easy-to-use interface provides a library of over 40 production- ready reports that can deliver the most relevant data in a few simple clicks. Drill-down capabilities provide a way for you to quickly explore the underlying data details, all the way down to the lowest level individual authority taxes. Our summary-level or detail-level reports allow you to choose the type of report data that best meets your immediate tax data needs in the most efficient way possible.

Compliance for US:

Regardless of location or industry, Sales & Use Tax Compliance has the forms required to meet your needs. It provides over 600 signature-ready state and local returns that are facsimiles of the official forms. Returns and schedules include sales, seller's use, consumer's use, and rental tax forms for all applicable states, as well as the District of Columbia. Industry-specific food and beverage returns are also included. In addition, more than 70 electronic returns are available and accepted in over 25 states. Sales & Use Tax Compliance is one of the market leaders in e-filing support. Thomson Reuters continues to work directly with state taxing authorities to ensure full compliance for each state's unique electronic filing requirements. The software also goes beyond borders to include the returns required for tax compliance in both Canada and Puerto Rico.

Compliance for VAT:

ONESOURCE Indirect Tax's flexibility accommodates your distinct VAT compliance requirements, while maintaining a robust risk management framework. It enables automated data collection and entry in a number of ways to ensure data integrity from numerous data sources. We maintain and update the latest tax rules, which enable you to focus on your indirect tax compliance rather than the implications of changing regulations using our solution maps and your company's unique in-house knowledge into the compliance process. We can reduce risk and assist with succession planning. Our VAT compliance solution has in-built and maintained VAT logic, automated VAT returns from data taken directly from financial systems and has detailed exception reporting embedded in the ONESOURCE software. It has a full audit trail of data from the return back to the source, and HMRC-approved XML e-filing capability.

Goods Movement:

A separate Goods Movement product is now written within the Integration as an add-on for all of your US sales tax material and goods movement use tax accrual needs. The integration uses all the table tables and SOAP interface technology that was provided as part of Integration Version 6. This menu driven version takes advantage of our entire table mapping and menu features for you to batch process your goods movement entries as part of your month end accrual process. Tables allow you to configure all transaction movement types you desire to accrue as well as the use of the field mapping logic for adding additional data elements to the response and request data for the Goods Movement routes and journey paths.

GLOSSARY

The following terms may be helpful when implementing Integration:

TERM	MEANING
SOAP (Simple Object Access Protocol)	SOAP is a way to build connections between software applications across networks including the internet. It works very much like a Yahoo search, where you pass in a request and get a response from a server. SOAP requests and responses are in the form of XML messages.
WSDL (Web Services Description Language)	A WSDL describes a way to send messages to a software application and how to read the corresponding response.
XML (Extensible Markup Language):	XML is a messaging language that is relatively easy to read for both people and software.

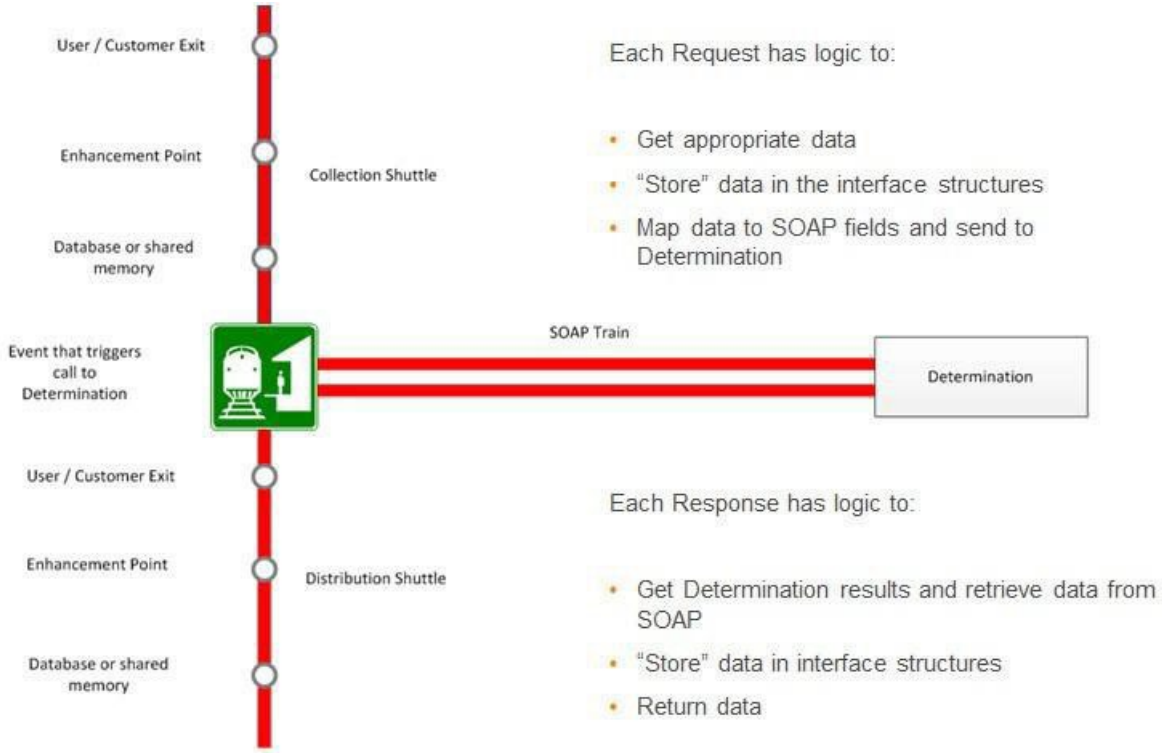
TERM	MEANING
Proxy	Within an SAP environment, a proxy is a representation of an outside application. Proxies can be generated from a WSDL. The proxy acts as if it is the outside application to the rest of the SAP environment. Any messages sent to the proxy are forwarded on to the outside application and responses from the outside application are returned through the proxy.
Pricing Procedure / Calculation Schema / Tax Procedure	A pricing procedure, calculation schemas, and tax procedures contain a list of conditions that form the spine of the pricing process. It must be correctly configured for the tax calculation process to work correctly.
Pricing Conditions / Tax Conditions	Pricing conditions are the vertebrae of the pricing procedure in SD and the calculation schema in Purchasing. Likewise, tax conditions are the vertebrae of the tax procedure as used by Logistics Invoice Verification and FI-AP/AR. Each pricing condition represents a step in the pricing process. Four pricing conditions must be correctly configured for the tax calculation process to work correctly in SD and Purchasing; a line-item data collector condition, a Determination calling condition, and two conditions to place calculated tax data into the prices. For the tax procedure only the two conditions to place the calculated tax data into the system is required.
Group Conditions	Group conditions are pricing conditions that work at a whole document level instead of at a line-item level as the other pricing conditions do. For this reason, they are very useful in calculating whole document taxes as required by many tax laws.
Condition Value Formulas / Scale Base Formulas	These are a type of user-exit that is part of a condition of a pricing procedure, tax procedure, or calculation schema.

THE DATA INTEGRATION MODEL: THE TRAIN STATION ANALOGY

To visualize the movement of data between SAP and Determination we have come up with the below noted diagram that uses the analogy of a train station and shuttle bus transportation system. Some of the terminology that you hear us use within this analogy will become familiar to you as we use them again in describing various steps within the configuration processes and field mapping of the ONESOURCE Indirect Tax Integration for SAP.

The Train Station is the point at which the request data has been gathered and is ready to be sent to Determination on a Train, as well as the point where response data from Determination is being sent back and is ready for distributing to SAP transactions and tables. The Train Station is the event that triggers a SOAP call to and from Determination. The double red line represents the flow of the data to and from Determination (the Train).

The single red line on the top represents the path or Collection Shuttle that is used to gather data from the SAP system's various modules into the Determination request. Along the shuttle's route there are various points or shuttle stops that are executed to pick up data to go to Determination on the Train. These can include user/customer exits within ABAP programs, program enhancement points, or areas within the database or shared memory within SAP.



Likewise, on the Distribution Shuttle line, data is being returned to various points or shuttle stops that are executed to return data from the Determination Train. These can include user/customer exits within ABAP programs, program enhancement points, or areas within the database or SAP tables. The data points are "picked up" or "dropped off" via the shuttle. How the data points are mapped (or assigned a seat on the shuttle) relates to the logic and procedures within the various Journeys and Routes.

A Journey is an object that handles the complexity inherent to a specific set of data that is sent to or received from Determination. A Journey includes logic to pass data to/from SAP transactions, store the data, and move data from/to specific fields or seats on the Train. For a list of all the journeys and their use see the Installation Guide section on additional information on Journeys.

A Route is an object that handles the complexity inherent to a group of transactions. Think of it as the bus route that a shuttle takes from a given “side of town” to/from the Train Station. There can be many different routes to get to the Train Station. Sales, Group Billing, Group Purchasing are routes that handle the complexity specific to the SD Sales, SD Billing, and MM-Purchasing transactions, and user-exits. Other routes include Non-Group-Doc-AP, and Non-Group-Doc-AR, and Non-Group-Doc-LIV which handle the complexity inherent to a group of transactions in Accounts Payable, Accounts Receivable, and Logistics Invoice Verification. For a list of all the routes and their use see the Installation Guide section on additional information on routes.

PRE- INSTALLATION STEPS

DOWNLOADING THE SOFTWARE

To download and install Integration, retrieve the ONESOURCEIDTIntegrationSAP_xxxx.zip file from the

Customer Center:

1. Open Microsoft Internet Explorer and navigate to Customer Center.
2. Log on using the Username and Password assigned by Customer Support.
3. Find Integration for SAP (Enhanced Global) in the Available Products list and verify that Installed Version is set to “None”.
4. Click Download for Integration for SAP (Enhanced Global)
5. Save the file.
6. Unzip the ONESOURCEIDTIntegrationSAP_xxxx.zip file.

DOWNLOADING THE DOCUMENTATION

The latest documentation files are available in the Thomson Reuters Knowledge Base.

1. Open Microsoft Internet Explorer and navigate to the Customer Center.
2. Log on using the Username and Password assigned by Customer Support.
3. Select the document file(s) relevant to your task and save them locally to your system.

Note: It might be best to create a new directory ONESOURCEIDTIntegrationSAP_<version>, where version is the release number of the product documentation downloaded. Then save all documents into that one folder.

SYSTEM PREPARATIONS

Before you import the provided SAP transports, make sure you have an SAP system dedicated for installation of Integration. It is highly recommended to do the first import into a Sandbox or Development system. Never import directly into a Production system.

To successfully install Integration and test communication with a tax engine, make sure you have Determination configured as well.

INSTALLATION INTEGRATION

IMPORTANTING TRANSPORT OBJECTS

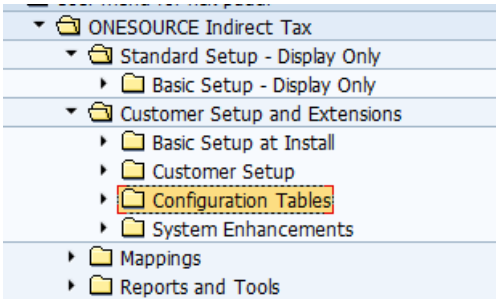
Thomson Reuters provided transports will need to be imported into the clients Development system to setup ONESOURCE Indirect Tax Integration for SAP. All our objects are in the Thomson Reuters Indirect Tax name space /IDT/. We don't change any customer owned objects.

For transports details refer to Product updates guide documentation.

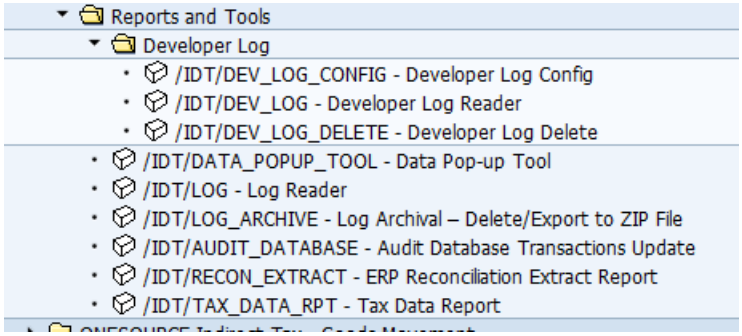
ADDING USER MENU TO A ROLE

For users to be able to access the ONESOURCE Indirect Tax for SAP setup, configurations, and reports we delivered an SAP User Menu. For users to be able to access the User Menu they would need to be given access to the /IDT/GENERAL or /IDT/USER role.

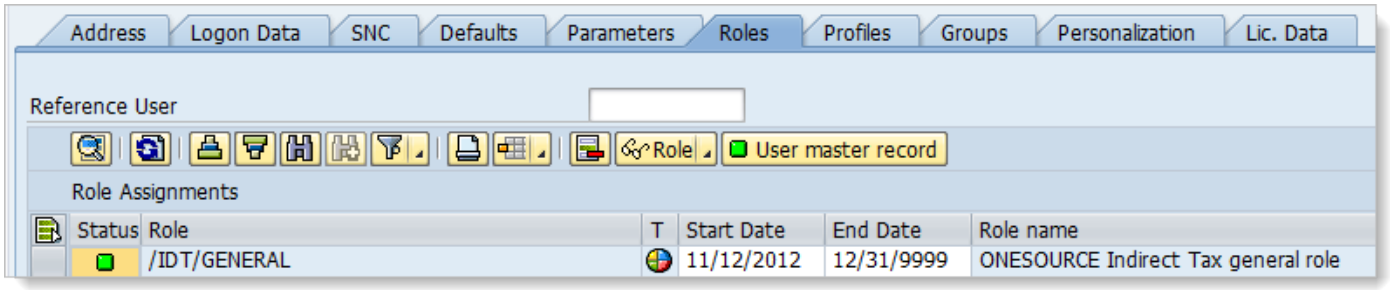
/IDT/GENERAL contains all of the IDT transactions codes and the full ONESOURCE User Menu. This would normally be given to users that oversee management of the system, configuration, and mapping.



/IDT/USER contains three menu options within the Reports and Tools menu and would normally be used by a system user that needs to review log reports, extract data for the reconciliation report, and update or review the audit database at month end for any transactions that posted to the G/L but did not go to the audit database.

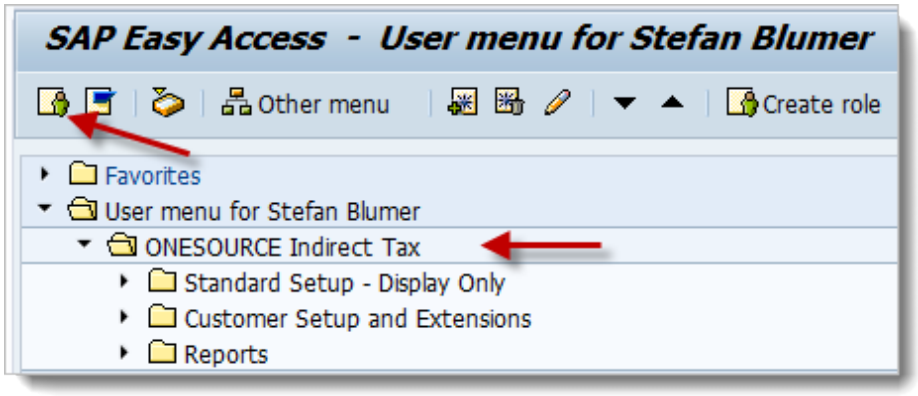


To assign the role to a user go to Transaction Code SU01. On the Roles Tab, insert the role name, beginning and ending dates as shown below.



Note: Only one or the other role needs to be assigned, not both.

Once the role has been assigned a user can access the User Menu from the main SAP start screen as follows by selecting the User Menu icon or Ctrl+F10.



ADDING INCLUDE STATEMENTS

To connect our tax interface with SAP business processes, code must be added in several areas of the SAP system, mainly in formulas, user-exits, BADI, and other methods further described in this section. In each case you will add a statement of code that then calls the Thomson Reuters delivered interface code.

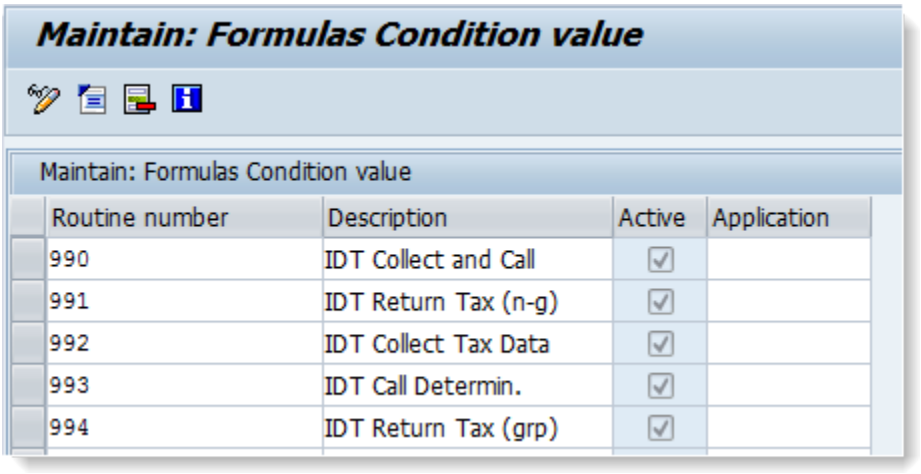
Note: Each customer's environment is different. When adding code to user-exits we don't know what other code has been implemented in the same area of the user-exit. It is the customers responsibility to review their code and determine the best place within the other custom code to add our include statement. In most cases it is probably best to add our include at the end of other code, unless prior code will skip our code, or we have specified otherwise in these instructions.

Creating Condition Value Formulas

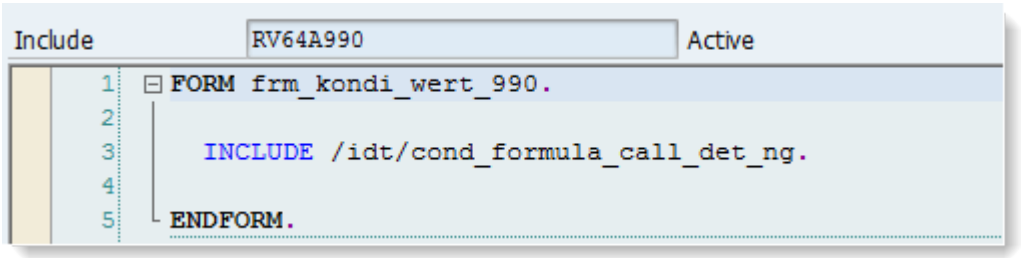
Use the next available formula in your system. The examples shown here are examples only. Once the formulas are created and generated you will be able to assign them to our Integration via configuration. See the Configuration Guide -> ONESOURCE Route Configuration for more details on this follow-up step.

Transaction Code VOFM -> Formulas -> Condition value

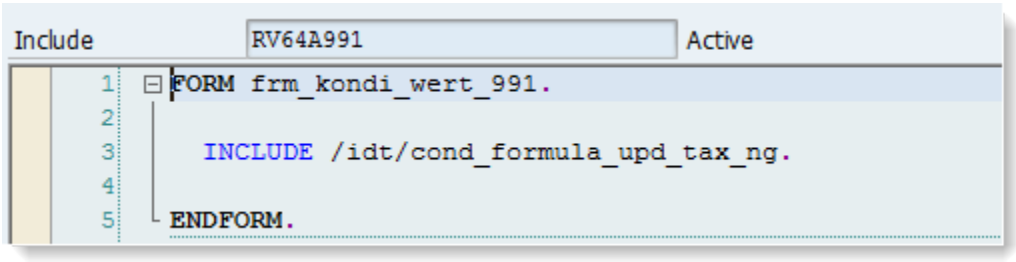
Create the formulas as outlined in the samples below:



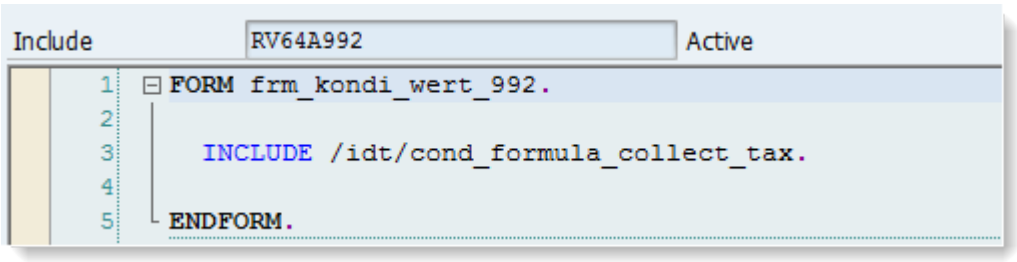
1. Collecting data and calling Determination (non-group process) [RV64A990] INCLUDE /IDT/COND_FORMULA_CALL_DET_NG.



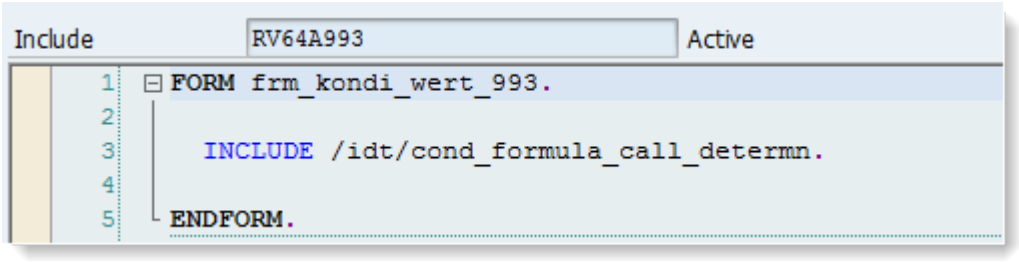
2. Update tax condition with tax data (non-group process) [RV64A991]
   INCLUDE /IDT/COND_FORMULA_UPD_TAX_NG.



3. Collecting line tax data (group process) [RV64A992]
   INCLUDE /IDT/COND_FORMULA_COLLECT_TAX

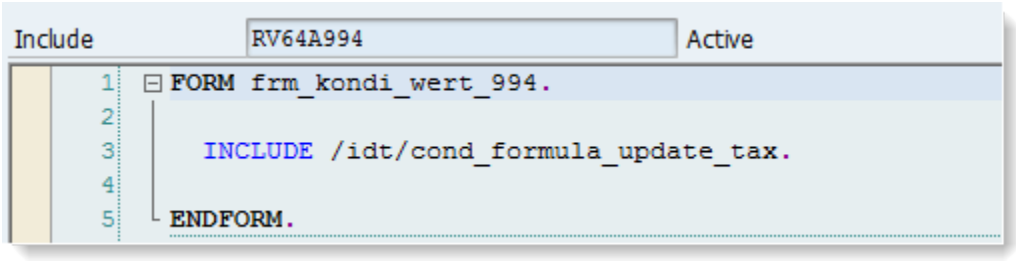


4. Calling Determination (group process) [RV64A993]
   INCLUDE /IDT/COND_FORMULA_CALL_DETERMN.



5. Update tax condition with tax data (group process) [RV64A994]

INCLUDE /IDT/COND_FORMULA_UPDATE_TAX.

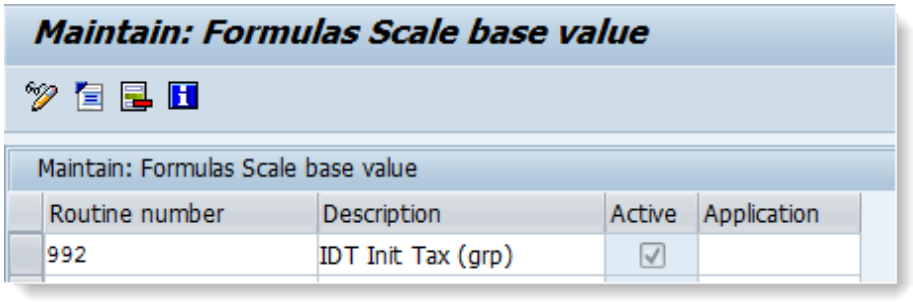


Creating Scale Based Formula

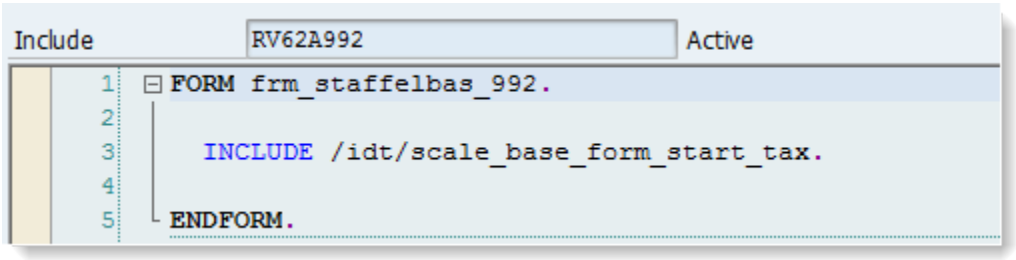
Use the next available formula in your system. The examples shown here are examples only. Once the formulas are created and generated you will be able to assign them to our Integration via configuration. See the Configuration Guide -> ONESOURCE Route Configuration for more details on this follow-up step.

Transaction Code VOFM -> Formulas -> Scale base

Create the formula as outlined in the samples below:



1. Triggering the start of pricing (group process) [RV62A992]
   INCLUDE /IDT/SCALE_BASE_FORM_START_TAX.

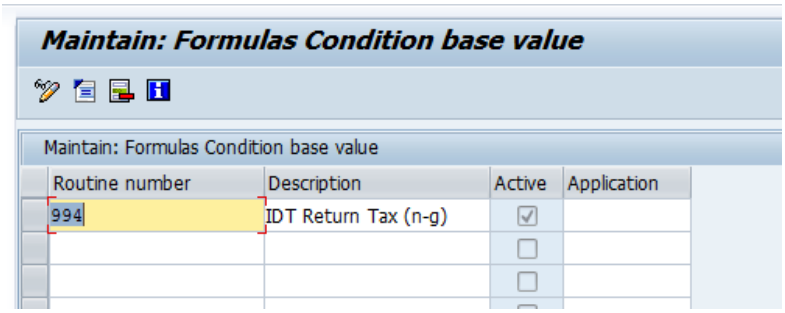


Condition Base Value

Use the next available formula in your system. The examples shown here are examples only. Once the formulas are created and generated you will be able to assign them to our Integration via configuration. See the Configuration Guide -> ONESOURCE Route Configuration for more details on this follow-up step.

Transaction Code VOFM -> Formulas -> Condition Base Value

Create the formula as outlined in the samples below:



1. Updating the tax base value (non-group process)
   INCLUDE /IDT/COND_FORMULA_UPD_BASE_NG.

SD User Exit Code

Implement the following user-exit code in support of address data collection for the tax interface, collecting transaction data for use in the field mapper, as well as updating the log with the sales document number once a document is saved in SAP.

MV45AFZZ

1. USEREXIT_PRICING_PREPARE_TKOMK
   NCLUDE /IDT/PRICE_PREP_TKOMK_SALES.
2. USEREXIT_PRICING_PREPARE_TKOMP
   INCLUDE /IDT/PRICE_PREP_TKOMP_SALES.
3. USEREXIT_SAVE_DOCUMENT
   INCLUDE /IDT/UE_SAVE_DOCUMENT.

RV60AFZZ

1. USEREXIT_PRICING_PREPARE_TKOMK
   INCLUDE /IDT/PRICE_PREP_TKOMK_BILLING.
2. USEREXIT_PRICING_PREPARE_TKOMP
   INCLUDE /IDT/PRICE_PREP_TKOMP_BILLING.
3. USEREXIT_SAVE_DOCUMENT_PREPARE
   INCLUDE /IDT/BILLING_SAVE_DOC_PREPARE.

the user-exit /IDT/BILLING_SAVE_DOC_PREPARE can't be found in RV60AFZZ please review SAP Note 1449861, 1831486 and follow instructions on how to implement this exit first.

Purchasing User Exit Code

The following code insertions need to be done in the MM area to enable tax calls on Purchase and Logistics Invoicing documents. They collect header and line information for the tax interface and field mapper, as well as update the logs with the document number during document save.

BADI ME_PO_PRICING_CUST

1. Method PROCESS_KOMK.
   INCLUDE /IDT/PURCHASING_BADI_KOMK.
2. Method PROCESS_KOMP.
   INCLUDE /IDT/PURCHASING_BADI_KOMP.

BADI ME_PROCESS_PO_CUST

1. Method POST.
   INCLUDE /IDT/PURCHASING_BADI_POST.

BADI EXTENSION_US_TAXES

1. Method MM_ITEM_TAX_MODIFY of BADI EXTENSION_US_TAXES should have the following include. This code is used to collect LIV transaction data.
   INCLUDE /IDT/BADI_LIV_EX_US_TAX_ITEM.
2. Method MS_TAX_DATA_SERVICES of BADI EXTENSION_US_TAXES should have the following include. This will make standard SAP call taxes every time when something is changed at line-item level for SES.
   INCLUDE /IDT/BADI_SES_EX_US_TAX.

BADI LE_SHP_PRICING

1. Method CHANGE_INPUT_HEADER_AND_ITEMS
   INCLUDE /IDT/PRICE_PREP_DELIVERY.

Internal Procedure Call

No coding is needed for this, but for documentation we list this internal call made within our Integration:

1. Method /IDT/ROUTE_GROUP_PURCHASING -> HANDLE_ERROR_MESSAGE calls macro MMPUR_MESSAGE_FORCED from include MM_MESSAGES_MAC. This code is to issue error messages within the purchasing transactions.

Price Condition User Exit Code

To enable the tax details to be available in the pricing screens of SAP the following insertions need to be made in the price condition user-exit areas:

RV61AFZB

These user-exits make sure that the XKOMV /IDT/* fields are not overwritten in various condition screens, as well as are used to make sure that the condition lines have the correct header condition counter in the condition screens.

1. USEREXIT_XKOMV_ERGAENZEN
   INCLUDE /IDT/UE_XKOMV_ERGAENZEN.
2. USEREXIT_XKOMV_ERGAENZEN_MANU
   INCLUDE /IDT/UE_XKOMV_ERGAENZEN_MANU.
3. USEREXIT_XKOMV_FUELLEN_O_KONP
   INCLUDE /IDT/UE_XKOMV_FUELLEN_O_KONP.

LV69AFZZ

These user-exits are used to populate the authority names in the condition types by overriding the default name of the condition with the authority name returned from Determination.

1. USEREXIT_FIELD_MODIFIC_KOPF
   INCLUDE /IDT/UE_FIELD_MODIFIC_KOPF.
2. USEREXIT_FIELD_MODIFICATION
   INCLUDE /IDT/UE_FIELD_MODIFICATION.

EXIT_SAPLMLSP_030/EXIT_SAPLMLSP_010

The below mentioned includes in the /IDT/ namespace are to be included in the user exit as shown below. These exits are part of the SAP enhancement – SRVESLL – Item level and SRVESSR – Header level. In ML81N these hooks are intended to pass transactions header and line-item data to custom structures and tables which are used to populate data while calling Determination. A project needs to be created and activated in CMOD and it should include the two enhancements SRVESLL and SRVESSR.

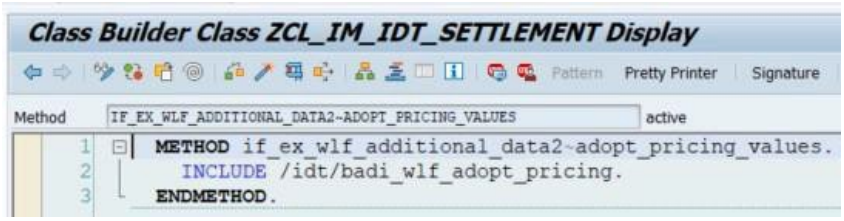
1. EXIT_SAPLMLSP_030
   INCLUDE /IDT/SERVICE_ENTRY_SHEET_ITEM.
2. EXIT_SAPLMLSR_010
   INCLUDE /IDT/SERVICE_ENTRY_SHEET_HDR.

Settlement Management BADI Includes

Note: The BADI list mentioned for Settlement Management are valid from S/4 HANA version 1809 onwards. The following BADI Implementations are needed for Settlement Management solution to work properly.

BADI WLF_ADDITIONAL_DATA2

1. Method ADOPT_PRICING_VALUES - Sets Pricing flag to move Tax code from pricing to Settlement Management Pricing Structure
   INCLUDE /IDT/BADI_WLF_ADOPT_PRICING.

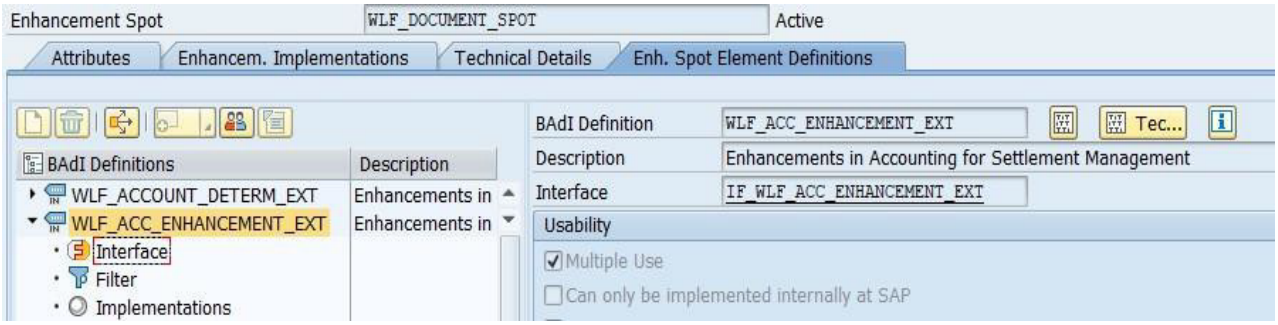


1. Method CHANGE_AT_UPDATE - Logic to pass Settlement Management Document Number and update Tax data table
   INCLUDE /IDT/BADI_WLF_CHANGE_AT_UPDATE.
2. Method CHANGE_PRICING_HEADER - Pass Header data to Collect Address Data
   INCLUDE /IDT/BADI_WLF_PRICING_HEADER.
3. Method CHANGE_PRICING_ITEM - Pass Item data to Collect Address Data
   INCLUDE /IDT/BADI_WLF_PRICING_ITEM.

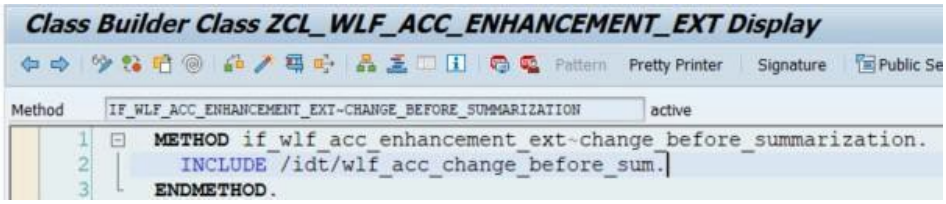
Note: Please also activate rest of the methods in this BADI to avoid adjustment messages from SAP.

BADI WLF_ACC_ENHANCEMENT_EXIT

This BADI is needed to move Tax data to Accounting. It is part of Enhancement Spot WLF_DOCUMENT_SPOT.

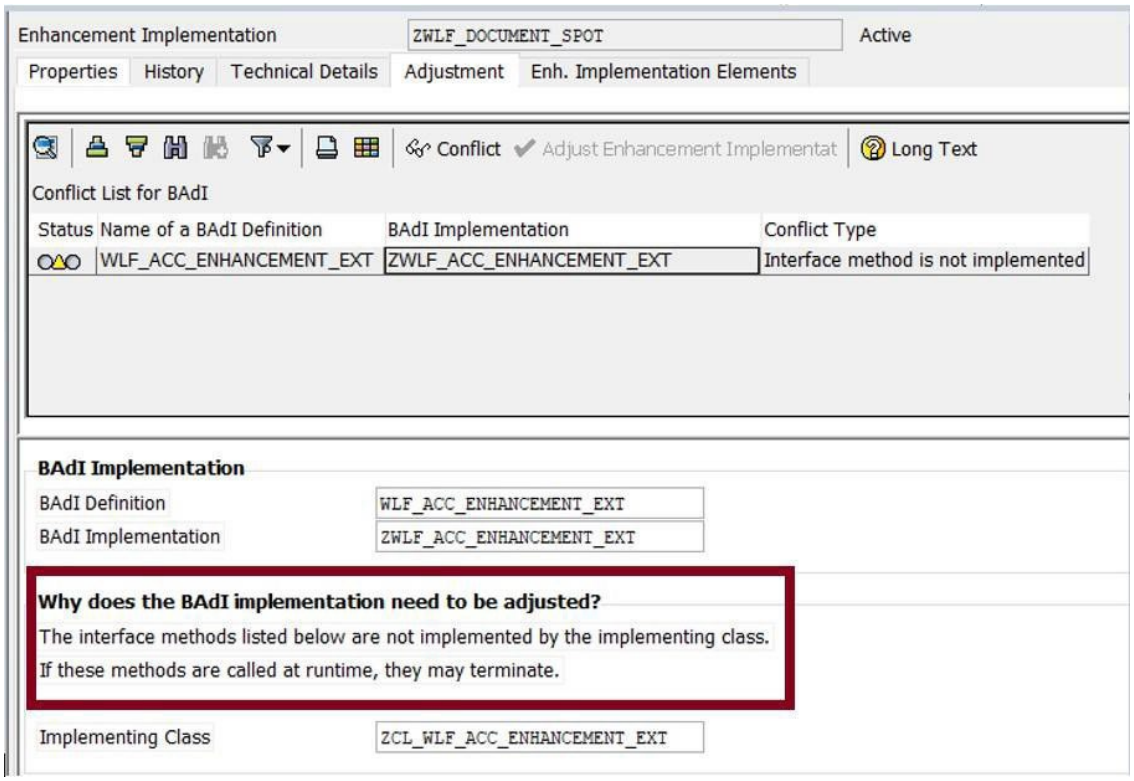


1. Method CHANGE_TAX_LINE - Tax lines are passed to SAP. Counter (TAXPS) keeps tax lines separate.
   INCLUDE /IDT/WLF_ACC_CHANGE_TAX_LINE.
2. Method CHANGE_BEFORE_SUMMARIZATION - Pass Accounting key (KVSL1) and Tax rate percentage (KBETR) to Tax lines.
   INCLUDE /IDT/WLF_ACC_CHANGE_BEFORE_SUM.



Note: Please also activate rest of the methods in this BADI to avoid adjustment messages from SAP.

screen example.



Implicit Enhancements

The following implicit enhancements are required to enable updating the tax code in the SAP transaction with the one returned by the Determination Tax Code Qualifier.

CALCULATE_TAX_DOCUMENT

At the end of the function module, right before the “ENDFUNCTION.” insert the include statement:

INCLUDE /IDT/CALCULATE_TAX_DOCUMENT.

MR_CALCULATE_TAX_DOCUMENT

At the end of the function module, right before the “ENDFUNCTION.” insert the include statement:

INCLUDE /IDT/MR_CALCULATE_TAX_DOCUMENT.

NDVAT_ADJUSTMENT

At the end of the function module, right before the “ENDFUNCTION.” insert the include statement:

INCLUDE /IDT/ADJUST_REV_EXP_TAX_DATA.

Include Program: LKONTF01

An implicit enhancement is to be created and the include /IDT/YBSEG_CREATE is to be placed at the end of form YBSEG_CREATE in include LKONTF01.This is used to link the DRSEG and BSEG tables during the LIV processing.

INCLUDE /IDT/YBSEG_CREATE.

FI_TAX_SV_COMPRESS

Add an implicit enhancement to the start of function FI_TAX_SV_COMPRESS. Insert the include statement:

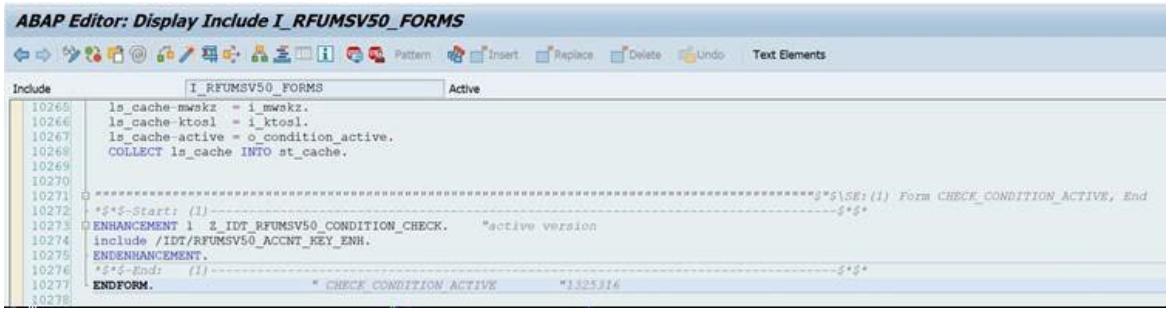
INCLUDE /IDT/FI_TAX_SV_COMPRESS.

Implicit Enchancement for Deferred Tax Program RFUMSV50

Note: From 6.6.0.0 release, Deferred tax solution for RFUMSV50 has been developed to work with S/4 HANA version 1809 and upwards.

Add an implicit enhancement at the end of the form ‘CHECK_CONDITION_ACTIVE’ in the include

I_RFUMSV50_FORMS and insert IDT include /IDT/RFUMSV50_ACCNT_KEY_ENH.



HANA System Manual Adjustments Required Post Install

There is a manual step involved after our product transports from 6.4.3.x are installed in the SAP system to modify the CDS views V_KONV and V_KONV_DRAFT. There are two S4 HANA versions out there. One is S4 simple finance and another is S4 Enterprise management which has both simple finance and simple logistics. This manual step is required only for the latter version not for S4 simple finance.

Refer to OSS note 2220005 and see attached document.

"If you have added an append structure to database table KONV in the source release, you must also add an append structure with the same fields (same field names, same data types) to the structure PRCS_ELEMENTS_DATA which is included in database tables PRCD_ELEMENTS and PRCD_ELEM_DRAFT and structure PRCS_ELEMENTS."

For S/4HANA implementation, a standalone structure /IDT/PRCS_ELEMENTS needs to be included in an append structure to PRCS_ELEMENTS_DATA DDIC structure. Please note this step is only applicable to HANA system and NOT ECC.

The structure /IDT/PRCS_ELEMENTS contains the below fields. These same fields should also be attached to CDS views V_KONV and V_KONV_DRAFT. Please refer to attached image.

/idt/auna,

/idt/etc,

/idt/uuid,

/idt/not,

/idt/item,

/idt/hdct,

/idt/dot,

/idt/ttyp,

/idt/gross

/idt/tax

Additional note per SAP documentation:

2.2 Impact on Customer DDIC enhancements

If you have added an append structure to database table KONV in the source release, you must also add an append structure with the same fields (same field names, same data types) to the new structure PRCS_ELEMENTS_DATA which is included in database tables PRCD_ELEMENTS and PRCD_ELEM_DRAFT and structure PRCS_ELEMENTS. Do not append your fields to PRCD_ELEMENTS directly. You have to add the append structure to PRCS_ELEMENTS_DATA in conversion phase SPDD.
This is necessary because only then will the automatic data conversion from KONV to PRCD_ELEMENTS convert the content of the append fields to the new database table PRCD_ELEMENTS.
The pre-check delivered via SAP Note 2188735 provides a check for such append fields, and issues a warning if these append fields are recognized in the source release. The warning refers to the necessary action in conversion phase SPDD.

Note

In S/4HANA, tables PRCD_ELEM_DRAFT and MMQTNPRICING_D are used to temporarily store pricing results as a draft during the creation or modification of documents prior to saving or 'activating' these documents.

When reading from PRCD_ELEMENTS into the internal KONV format, in many applications the CDS view V_KONV is used. This view reads data from database table PRCD_ELEMENTS and exposes them in the format of database table KONV. It is therefore necessary to enhance this CDS view, too. Otherwise, customers' own fields will not be automatically transferred when selecting pricing result records. The same is true for table PRCD_ELEM_DRAFT and CDS view V_KONV_DRAFT.

Caution

Customers' -own fields that are assigned to table KONV need to be added to structure PRCS_ELEMENTS_DATA which is included in database table PRCD_ELEMENTS and PRCD_ELEM_DRAFT and structure PRCS_ELEMENTS in the SAP S/4HANA code line. In addition the fields need to be appended to table MMQTNPRICING_D.

By extending the CDS view V_KONV and V_KONV_DRAFT the fields are considered in the CDS views and hence are read in automatically when the pricing result is accessed by the applications. Therefore for customers with own fields in KONV, there is also the need to extend the CDS views.

SAP provides centrally reports to migrate data from the old database table KONV to the new database table PRCD_ELEMENTS. The content of customer specific fields is automatically migrated if the fields exist in the source and in the target table.

Portion from Cookbook_Pricing_ConditionTechnique_20160725.pdf

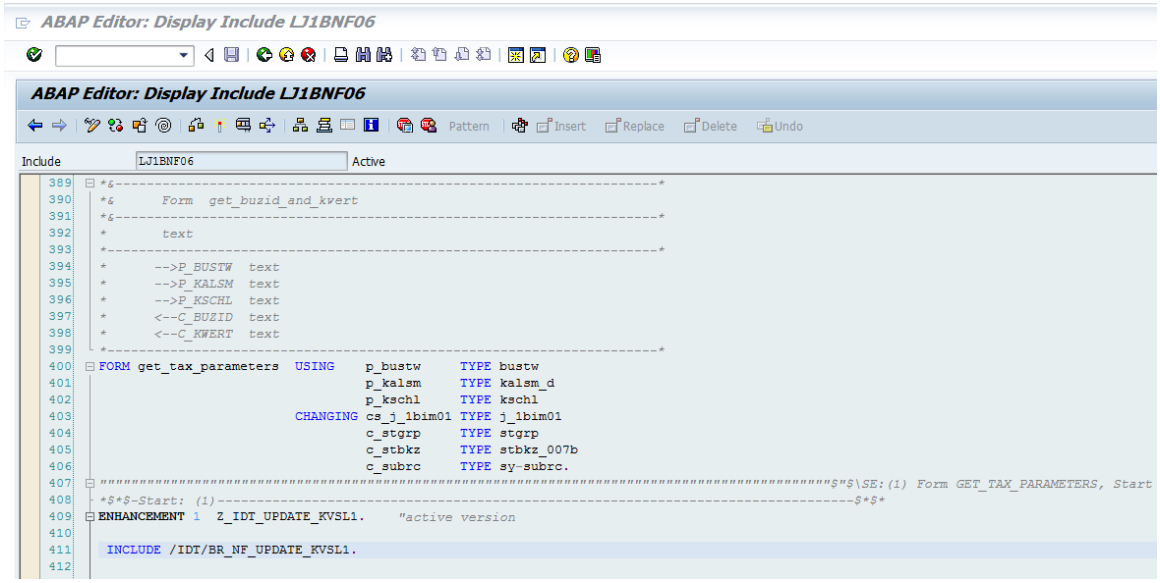
Brazil Implicit Enhancements

Note: Important: The following Brazil Enhancements include hooks which should only be added to the system if you are planning to use the Integration 6.4 Brazil enablement logic. If you have elected to leave Brazil on the Integration 5.x old interface, then by adding these hooks the program will create short dumps in your system. You can either elect to not add these includes into the programs or comment them out to not invoke them with the Brazil configuration if you are continuing to use the old Integration for Brazil. If you have multiple company codes for Brazil, you must keep them all on the same version to avoid this error.

Update Account Key when Brazil NF taxes are getting created

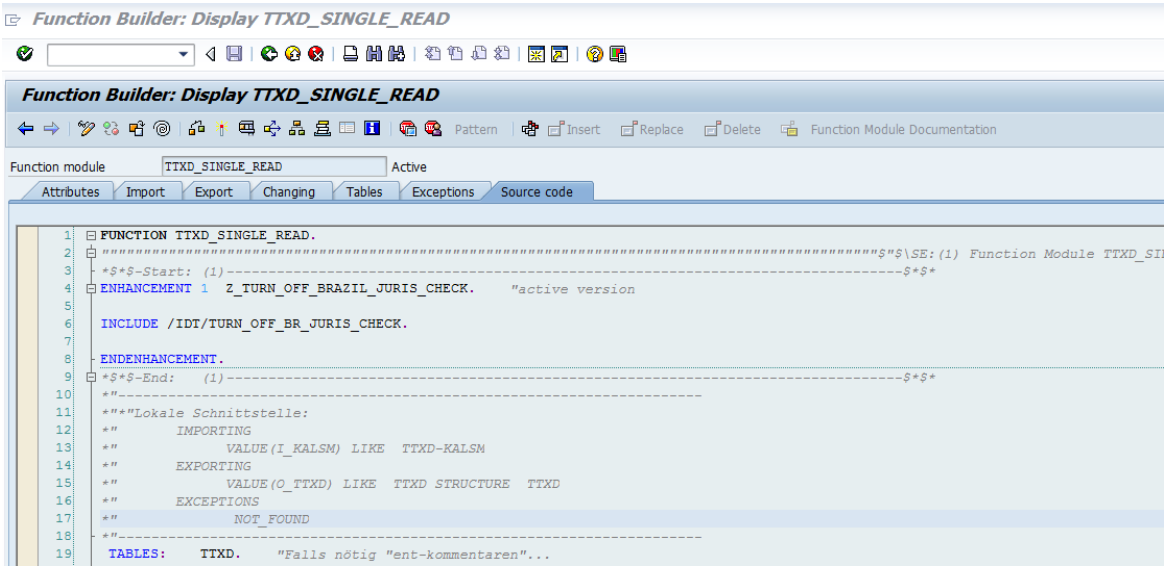
Include program /IDT/BR_NF_UPDATE_KVSL1 is expected to update the account key when NF taxes are getting created in Brazil.

This include gets called as part of implicit enhancement in include LJ1BNF06 as part of function module J_1B_IM_TX_CALCULATE_TAX_NEW at the start of the form get_tax_parameters. Please refer to the screen shot below for this hook.

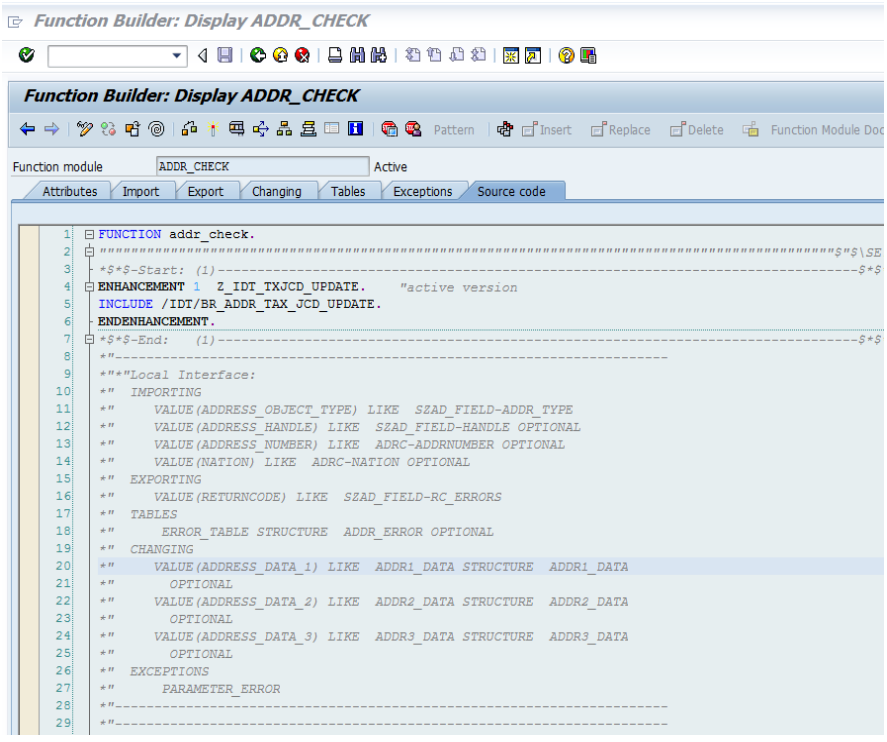


Tax Jurisdiction Check for Brazil

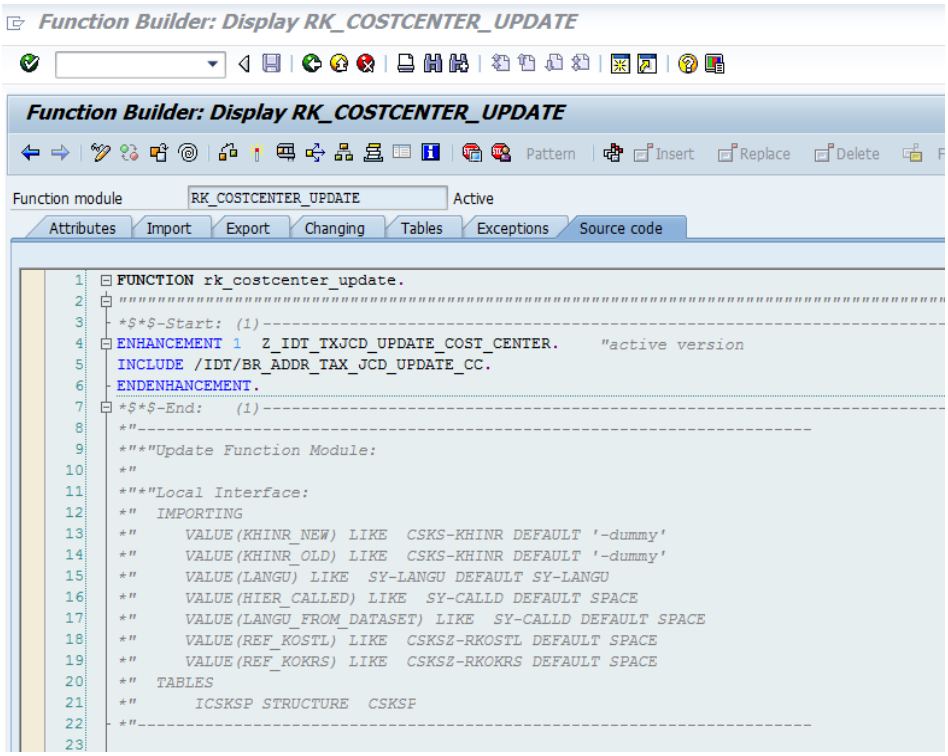
1. Create a new Implicit Enhancement at the start of the function module TTXD_SINGLE_READ. Include program /IDT/TURN_OFF_BR_JURIS_CHECK needs to be added.
   INCLUDE /IDT/TURN_OFF_BR_JURIS_CHECK.



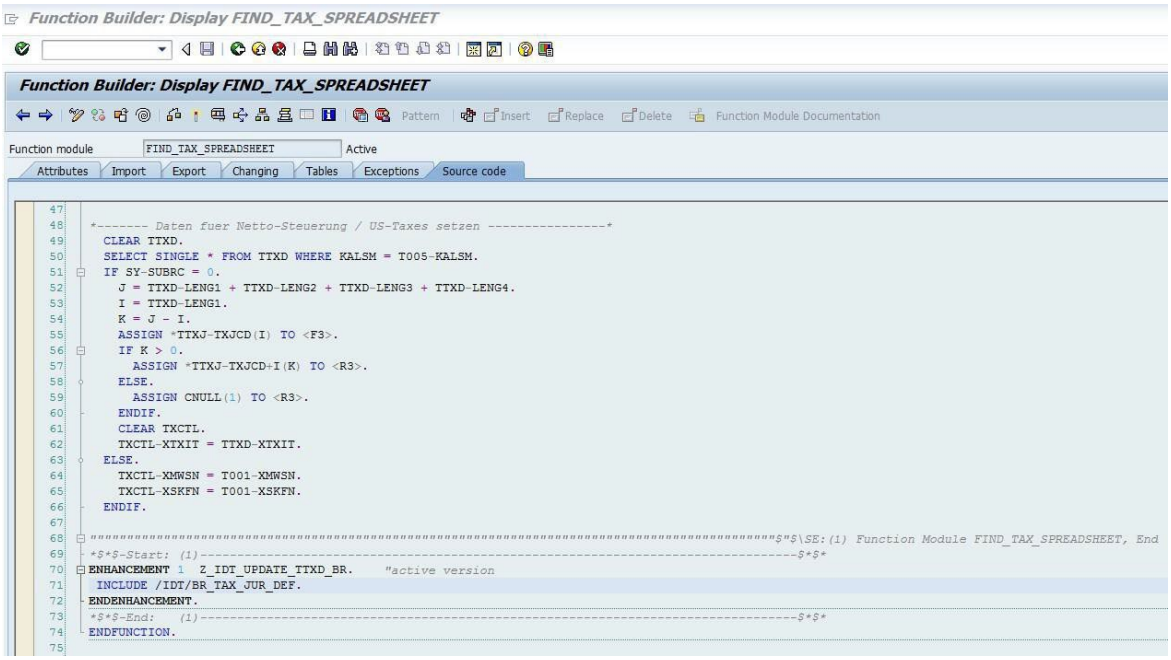
2. INCLUDE "/IDT/BR_ADDR_TAX_JCD_UPDATE" at the start of the function module ADDR_CHECK.
   INCLUDE /IDT/BR_ADDR_TAX_JCD_UPDATE.



3. INCLUDE “/IDT/BR_ADDR_TAX_JCD_UPDATE_CC” at the start of the function module RK_ COSTCENTER_UPDATE. This is required only if the TXJCD is to be determined for Cost Center.
   INCLUDE /IDT/BR_ADDR_TAX_JCD_UPDATE_CC.



4. INCLUDE “/IDT/BR_TAX_JUR_DEF” at the end of the function module FIND_TAX_SPREADSHEET. This is required to by-pass the Brazil TTXD table level data checks at some instances in SAP transactions (E.g.: Service Entry Sheet, STO etc.).
   INCLUDE /IDT/BR_TAX_JUR_DEF.

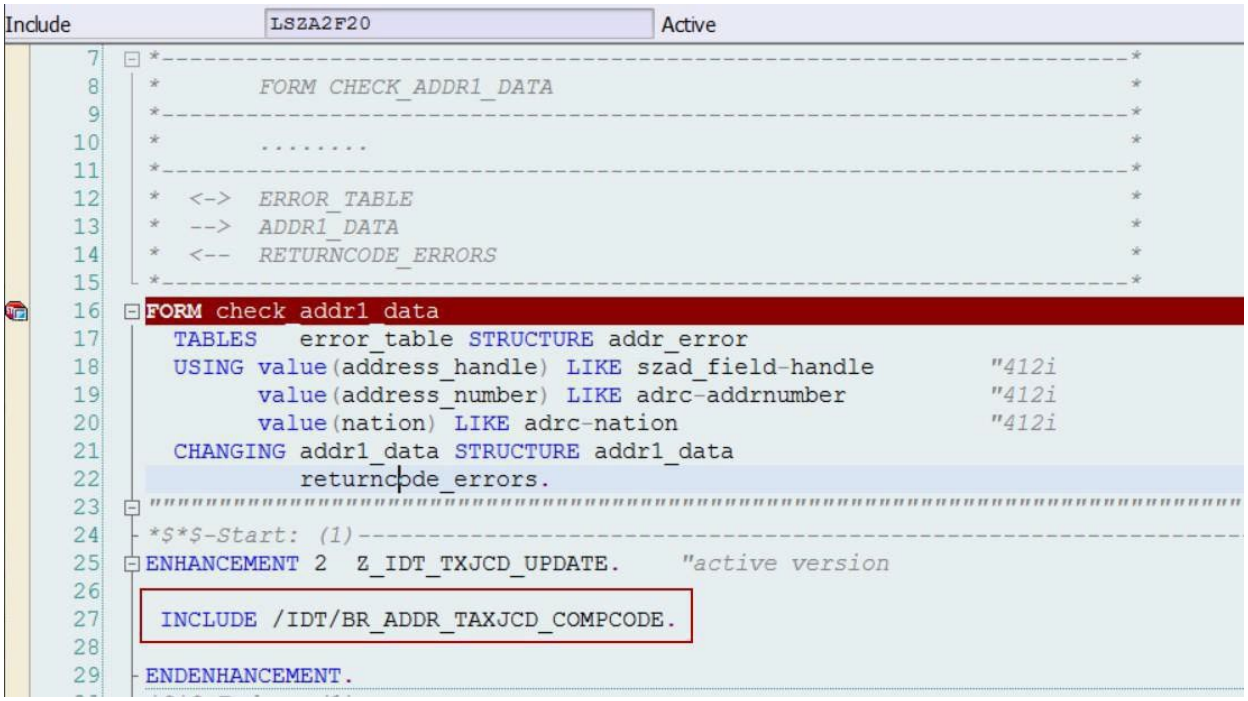


HANA Tax Jurisdiction Check for Brazil

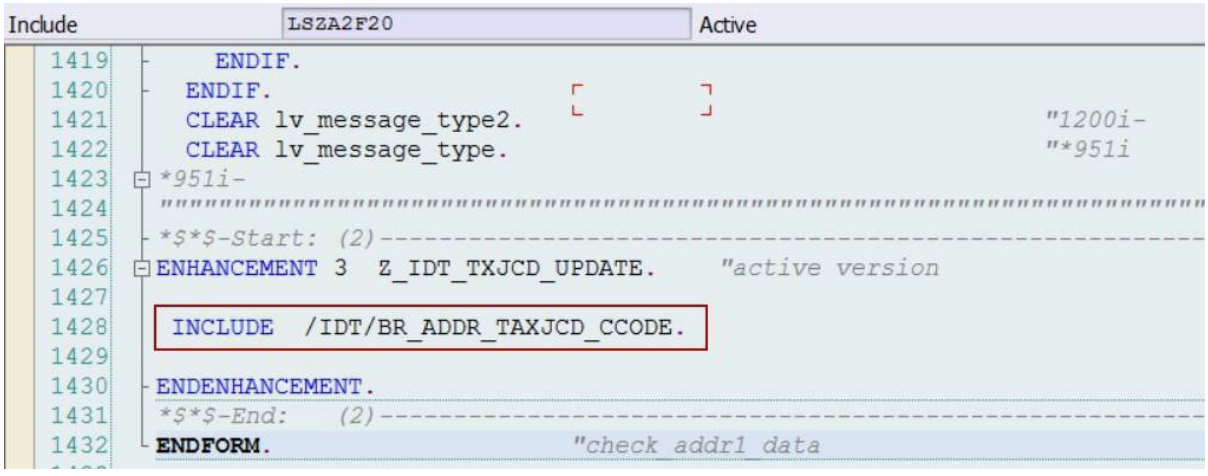
The HANA code has a slight difference in that there is an additional jurisdiction code check in the program that is not present in ECC versions. This requires two added include statements to be added to the programs to make the function work the same in HANA as it does in ECC for this feature. Please add the following two include statements as noted below:

These includes/hooks will be required and applicable only for HANA systems where GN is installed with Brazil functionality.

a. /IDT/BR_ADDR_TAXJCD_COMPCODE - This program will get called as part of an implicit enhancement created at the start of SAP standard include LSZA2F20, FORM - check_addr1_data



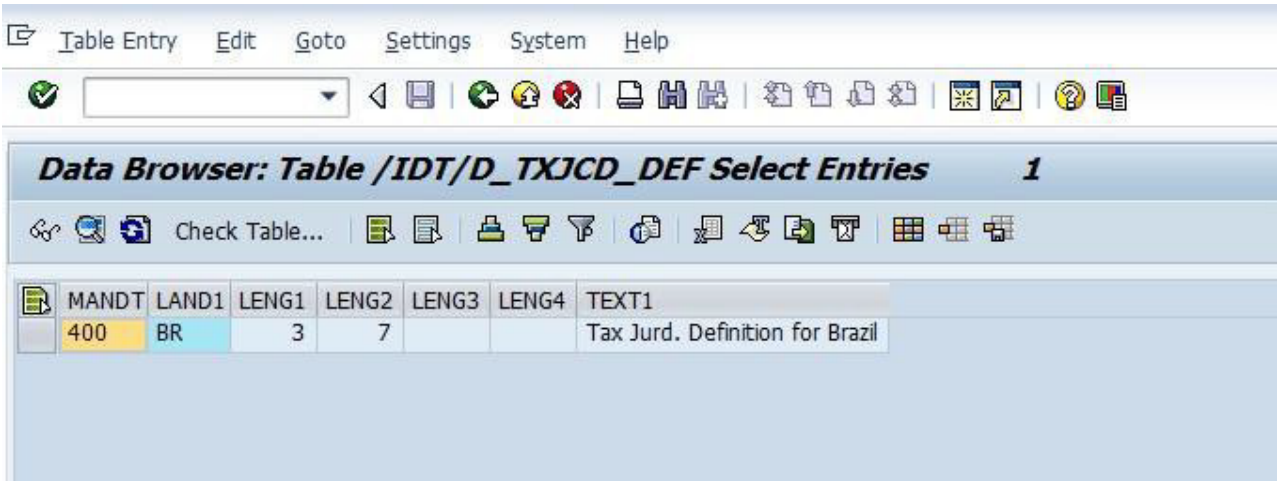
b. /IDT/BR_ADDR_TAXJCD_CCODE - This program will get called as part of an implicit enhancement created at the end of SAP standard Include LSZA2F20, FORM - check_addr1_data



Tax Jurisdiction Code Table for Brazil

A separate table /IDT/D_TXJCD_DEF is part of the configuration transport and this table will hold the Tax Jurisdiction Code Definition for Brazil. An entry on this table can be seen from the screenshot below. This table can be maintained with the transaction code /IDT/D_TXJCD_DEF.

Tax Jurisdiction Code for Brazil is determined from the address and the address will need the Country, Region and correct Postal Code to be filled in. The SAP table J_1BTREG_CITY will be used to get the Tax Jurisdiction Code value from the address data, and it needs to be populated with the valid.



Tax Region Determination for Brazil SD

Create a new Implicit Enhancement at the start of the function module J_1B_READ_DYNAMIC_TABLE. Include program /IDT/SD_REGION_BR_JURD_CODE needs to be added.

INCLUDE /IDT/SD_REGION_BR_JURD_CODE.

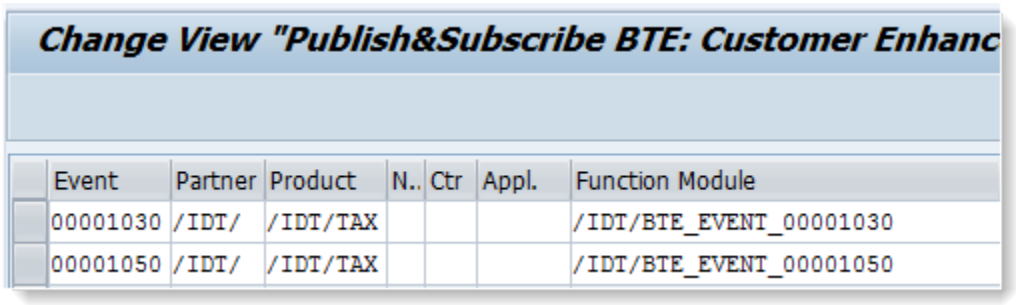
LIV Nota Fiscal Taxes and Tax Laws

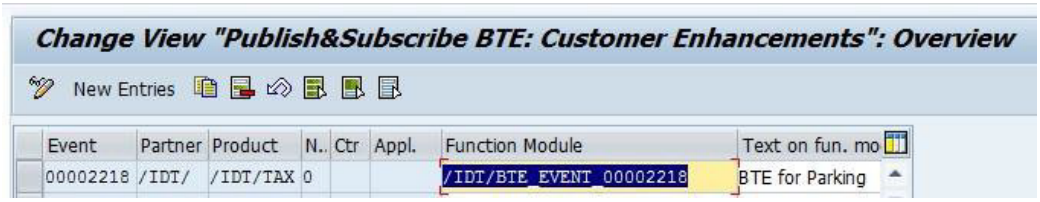
Create a new Implicit Enhancement at the start of the function module J_1B_PROCESS_TKOMV. Include program /IDT/MM_IV_ADD_NF_TAXES needs to be added.

INCLUDE /IDT/MM_IV_ADD_NF_TAXES.

REVIEWING BTE EVENTS

Business Transaction Event (BTE) 1030 and 1050 are used to mark the end of a G/L document posting process, update the document numbers in the logs, and to update the audit database. Functions /IDT/BTE_EVENT_00001030 and /IDT/BTE_EVENT_00001050 have been delivered via the earlier imported transports as Partner managed BTE's. BTE 2218 are used to mark the end of a parked document process and update the document numbers in the logs. There is no need to do anything manually, but for completeness they are listed here. To view the BTE's go to Transaction Code FIBF.



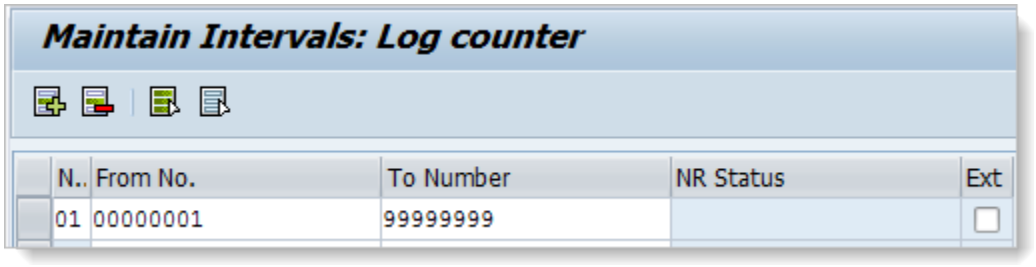


CREATING A LOG NUMBER RANGE

The SOAP request and response can be logged in SAP in XML format. Logging can be configured within the Integration setup, and logs can be searched and viewed in SAP using a provided transaction. Consult the User Guide for more details on logging. For logs to be stored with SAP a log number range for Object /IDT/LOG needs to be setup:

1. Transaction Code **/N/IDT/LOG_NUMBR_RANGE**
2. Press Intervals icon to create a new number range interval.
3. Press  to insert a new line.
4. Make sure that you use a value of 01 for the Number range number. Currently we only support one number range of value 01.

5. Fill out the "From Number and To Number". We recommend opening the number range as wide as possible.



6. SAVE the log number range.

CONNECTING SAP AND DETERMINATION

The ONESOURCE Indirect Tax Integration for SAP uses SAP's Internet Communication Framework to send tax calculation request to Determination and receive tax results back from Determination using the SOAP format.

Within an SAP environment, a proxy is a representation of an outside application which includes an internal interface layer within SAP and can be easily accessed from ABAP programs. Proxies can be generated based on another parties' WSDL definitions. Some WSDL features are not supported by SAP proxies such as the <union> tag and recursive element definitions. See OSS note 944029 for more details. Hence Thomson Reuters packages an SAP specific WSDL file for the proxy generation.

With the new Cloud Determination 2018.3 the tax engine added new fields to the WSDL that were designed to aid in customer utilization of Oil and Gas industry needs. Because the new fields (about 30) are only being offered in the Cloud Determination and not the On-Premise version, this will require a second applicable WDSL and Proxy. This addition requires several tables for field mapping to be updated to address possible conflicts between field mappings if both old and new proxy are used at the same time within the Integration. This will be discussed in several other applicable sections of the guides.

Note: Due to above mentioned limitations an SAP specific WSDL for the Tax Calculation Service in Determination has been packaged with this release. Do not point to the web service URL for importing the WSDL into an SAP Proxy

HOSTED CUSTOMERS CONNECTIVITY

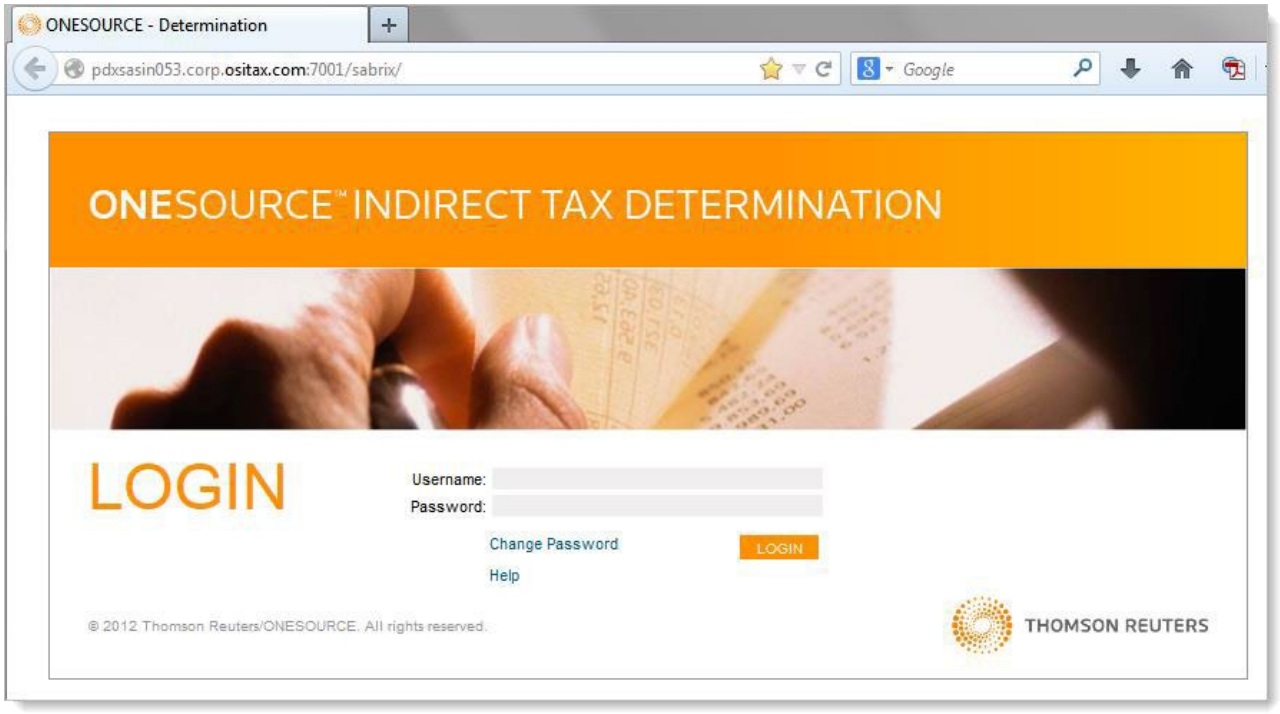
A separate Knowledge Base Article Install Certificate for SAP Global Integration 6.x.x.x. is published out on our customer support web site that has additional instructions needed to install for HTTPS hosted connections.

This KB article includes information on:

1. Location/link for customers to download certificates to be installed on SAP server.
2. Steps to install the certificates in SAP via transaction STRUST.
3. Required expertise in ICM Configuration Expertise as a per-requisite to do the configuration.

ACCESING DETERMINATION UI

It is recommended to first test direct access to the Determination UI by using a browser. Enter the Determination URL. You should see the logon screen.



Note: The Determination URL is http://<host>:<port>/sabrix (replace <host> with the name of the computer hosting the application server and replace <port> with the port number).

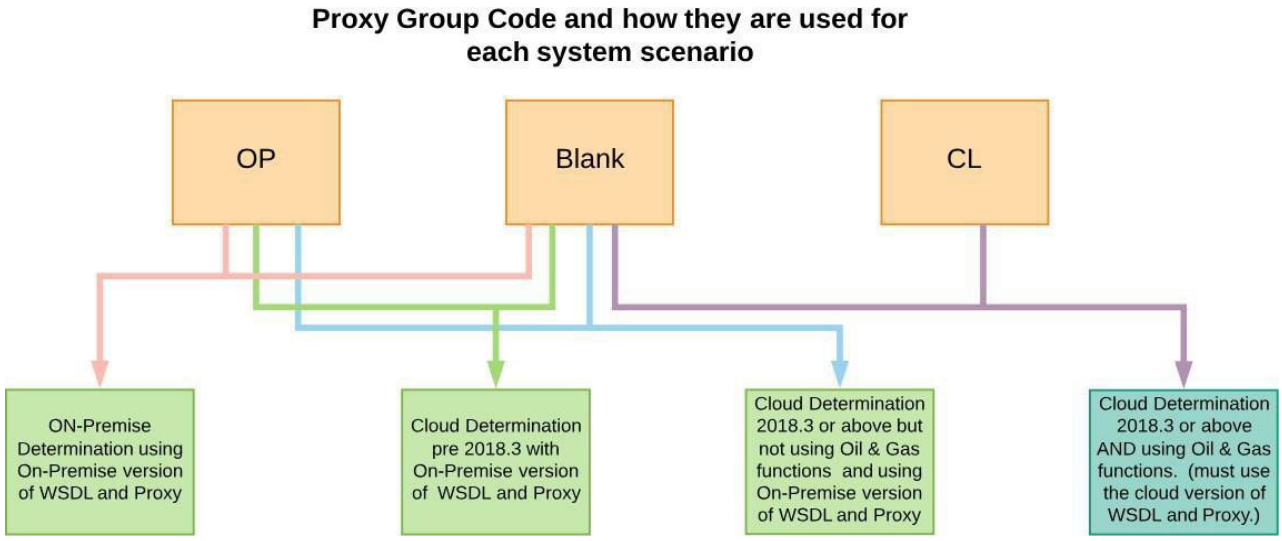
Note: If you can't access the Determination logon screen as outlined above do not proceed, contact a Determination System Administrator to make sure you can access the system first.

SETTING UP THE SOAP INTERFACE PROXY

Use of Cloud Determination Proxy in Tandem with On-Premise

With the release of Cloud Determination 2018.3 there were new fields, functions, calculations, and content added to the tax engine to enable calculations for Oil and Gas customers. Because there are new fields added that are only supported in the Cloud offering of ONESOURCE Indirect Tax, it required the creation of a separate WSDL, XML structure, and Proxy for use with Cloud Determination and Integration. Because these new fields are not available in the regular On-Premise version of our Integration it creates certain complications and can cause system errors if users attempt to map a field that is only available in Cloud version, if both Cloud and On-Premise versions of the Determination are being used at the same time. We have added some additional fields to our configuration tables to prevent mapping caused ST22 errors. The errors can be experienced if there is a conflict between the two separate Determination instances and their respective and different XML structures.

You will not have to worry about this unless you are converting to Cloud Determination and have both a Cloud version of Determination and an On-Premise version of Determination active in your system at the same time. If you are only using one Determination instance, then you will not have an issue and can ignore these fields. We are now providing both versions of the WSD as part of our transports to load the Integration and you will have to select the correct file to create your proxy in the following steps. If you only have one Determination instance, then you can skip this section and proceed to the next section Creating an SAP Proxy. See diagram below which will clarify how the two different WSDL and Proxy configurations could be used.



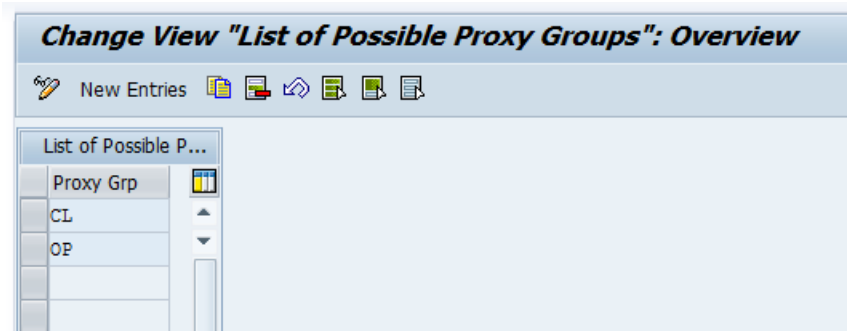
Viewing Proxy Group Designators in /DT/D_PROXYGRPS Table

This table was added and is populated as part of the transports. They will be used in the proxy configuration set up as well as within the field mapping table for you to specifically assign field mappings for the new Cloud Determination to the correct Proxy configuration. It can also be used to limit field mapping lines that are not wanted in the Enterprise Cloud Determination by designating them as “OP” for on premise version only. The fields in this table are standard and not changeable by the user. The three options for this field are:

CL: used on field mapping lines that are only applicable to the Cloud Determination 2018.3 and above

OP: use on field mapping lines that are only applicable to the On-Premise Determination or earlier versions of Cloud Determination 2018.2 and below.

Blank: used for most all fields and means that it can be used for either Cloud or On-Premise Determination versions. (this is our default)



Remember these two proxy group names as you will need to use them for configuration of the Proxy table and for new field mappings for fields that are only available on either the Cloud version of Determination or the On-premise version. This table is not on the standard menu and is only viewable via SE16..

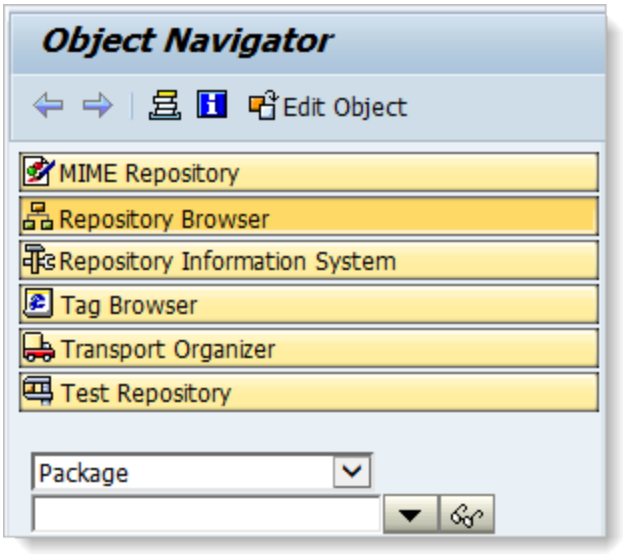
Creating an SAP Proxy

In the following steps we will configure an SAP Proxy and then configure a communication between SAP and Determination using SAP's SOA Manager.

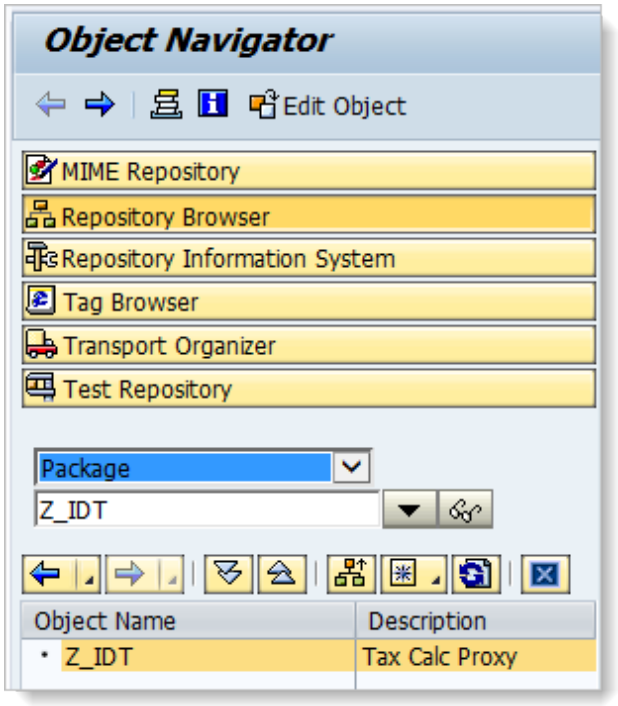
Note: Screens and steps shown in below illustrations are mainly from an SAP ECC 6.0 EHP6. Due to SAP's frequent changes in this area of the system, your specific screens might look slightly different.

The steps below require a developer's key and access to the correction and transport system.

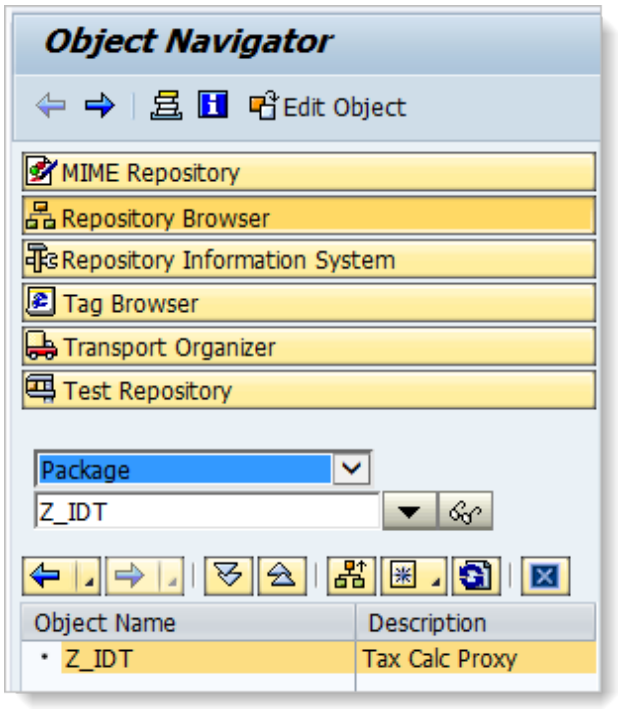
1. Go to Transaction Code SE80 and select the "Package" setting.



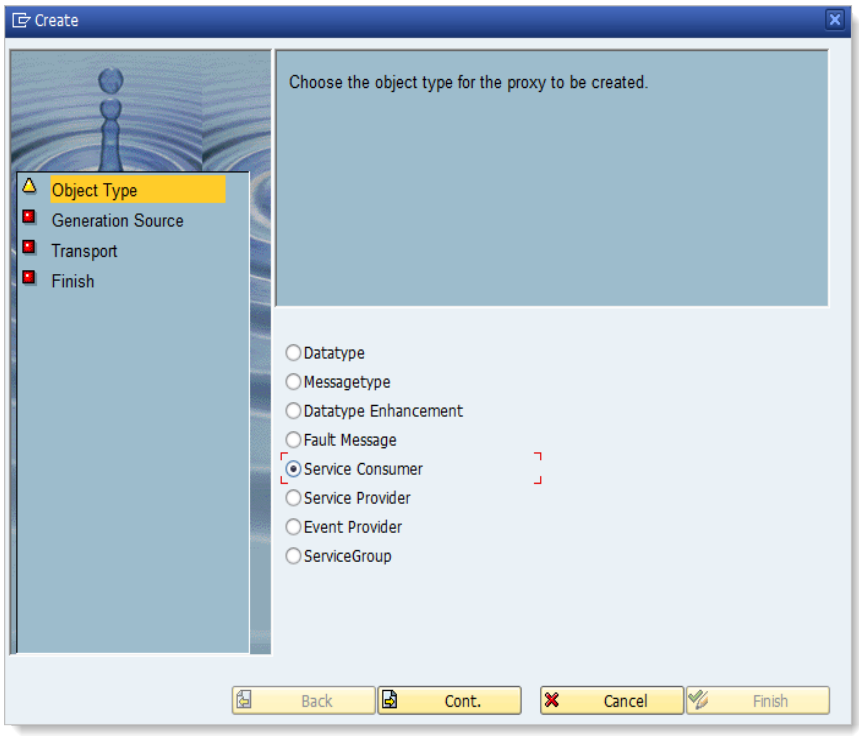
2. Enter the package you want the proxy to be in and press enter. (You may need to create the package at this point and add it to a transport.)



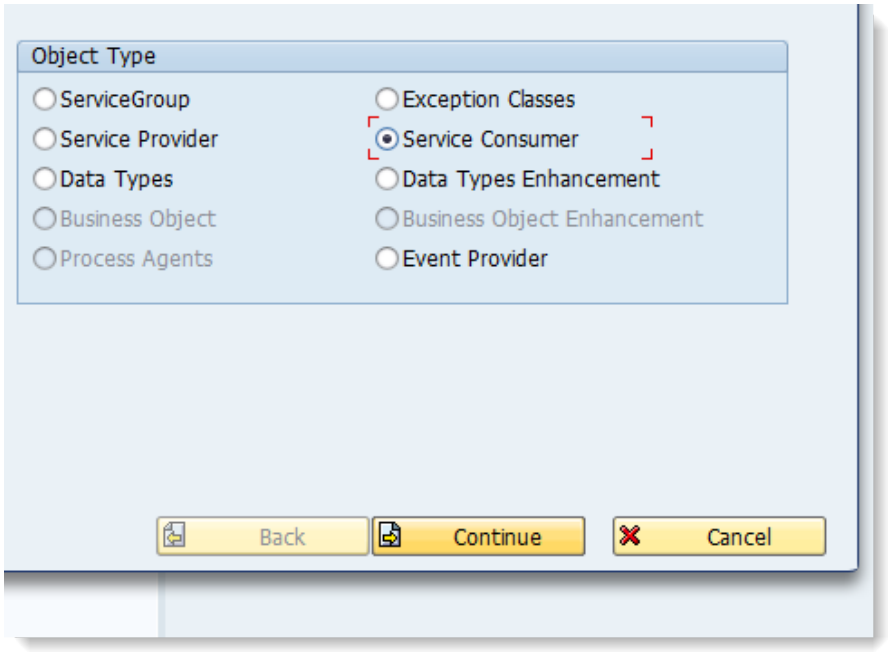
3. Select the package root and right-click. Select Create->Enterprise Service.



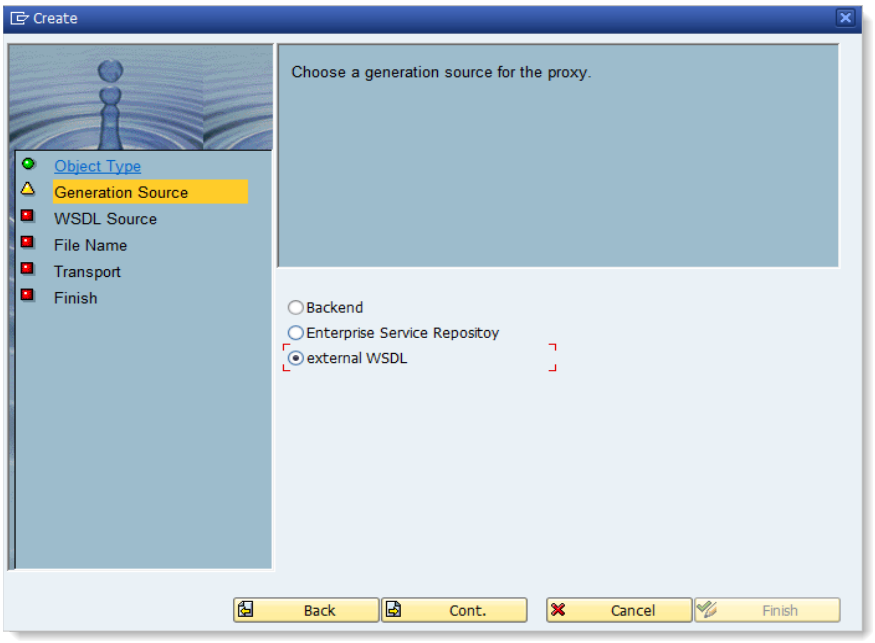
This should take you to something like below picture (this might be slightly different depending on the NetWeaver version used).

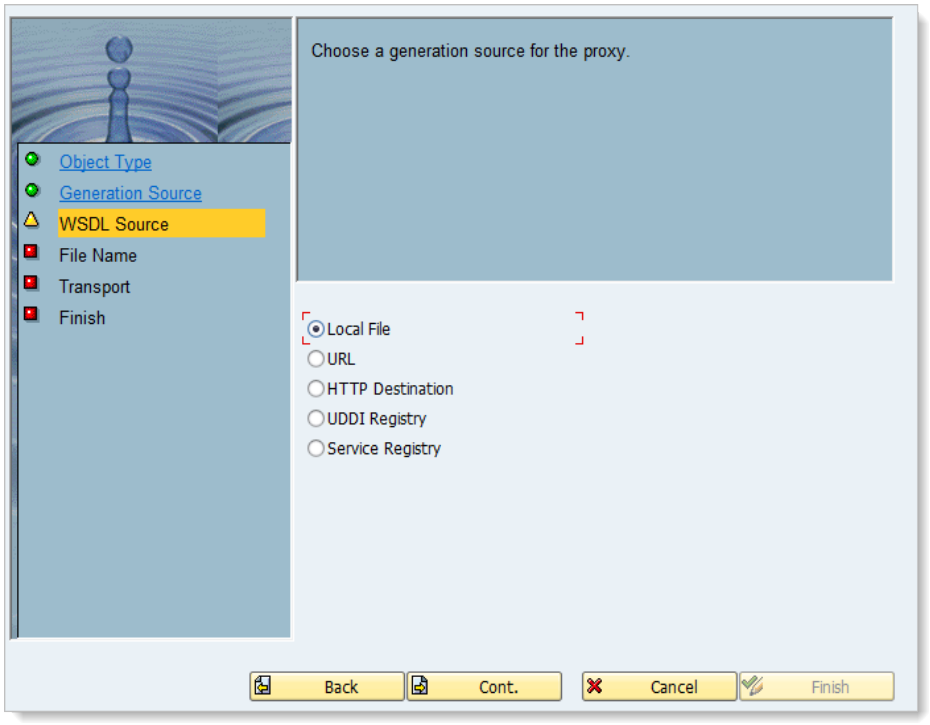


4. Select “Service Consumer”.

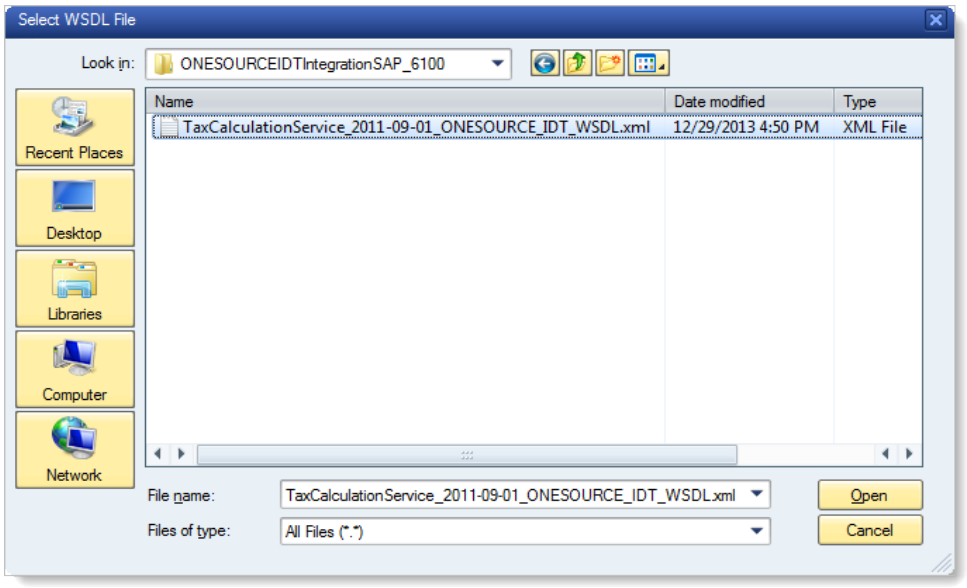


5. Then press cont. icon which should take you to something like the screen below.



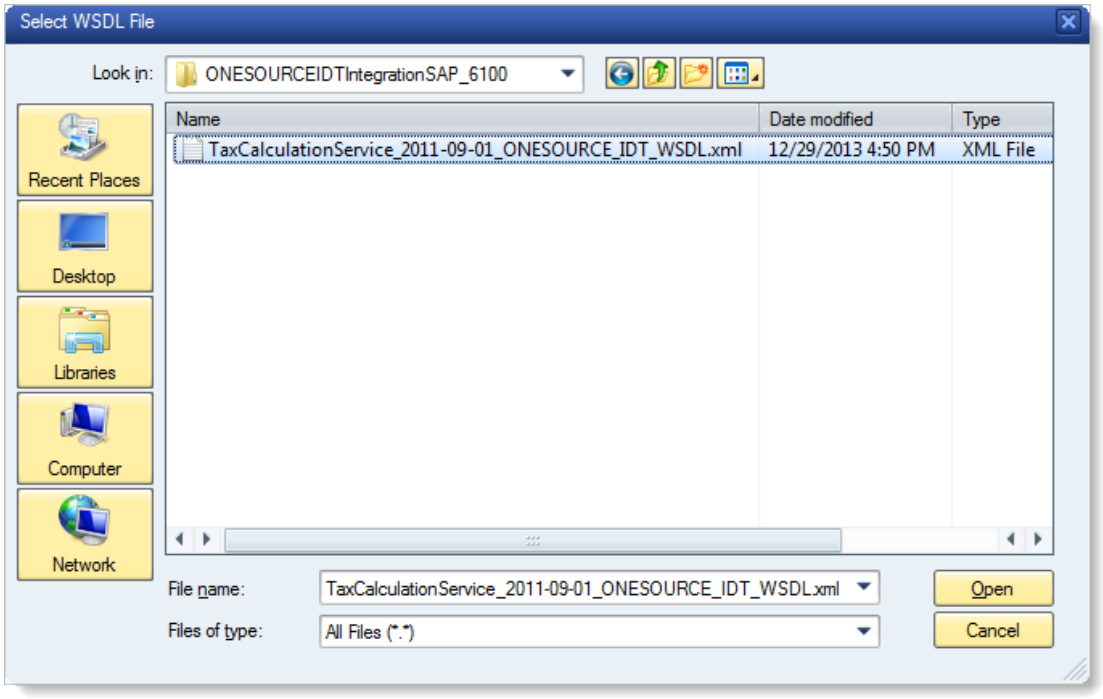


6. Select external WSDL and press cont icon , then select “Local File”.



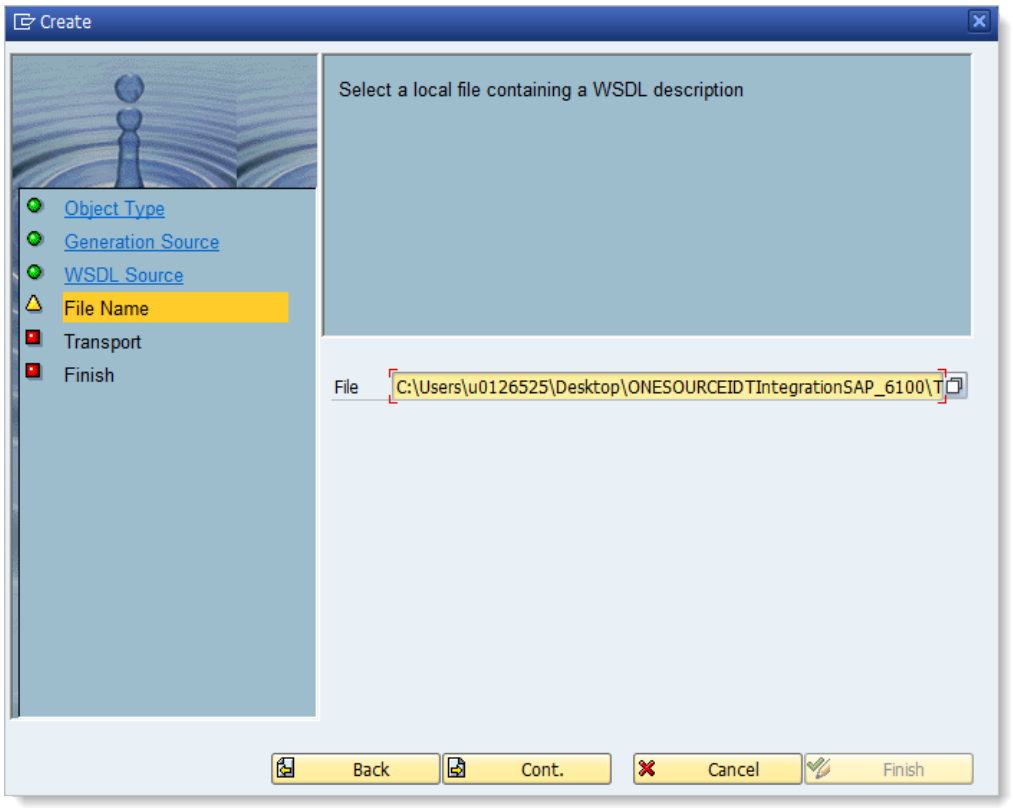
7. Then press Cont icon which displays a file selection option
8. Browse to the TaxCalculationService_2011-09-01_ONESOURCE_IDT_WSDL.xml file found in the Code folder at the location you placed it during the Downloading the Software steps. You will see two WSDL and one will be labeled as Cloud Det 2020.03 and above. Select

the one you need from this list based on which Determination instance you are configuring. If you are doing both on-premise and Cloud Determination 2020.03 then you will likely need to do this process twice.



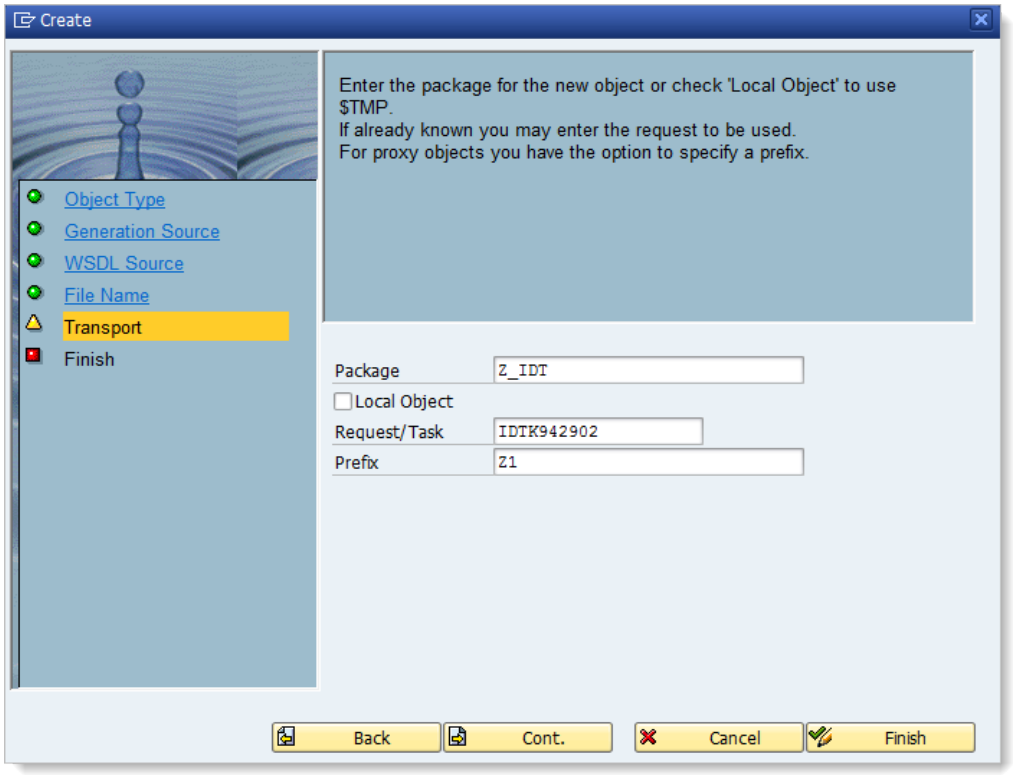
Screen will show 2 and you will need to know which to pick.

9. Press “Open” to get the file selected.



10. Then press Cont icon which takes you to Transport step.

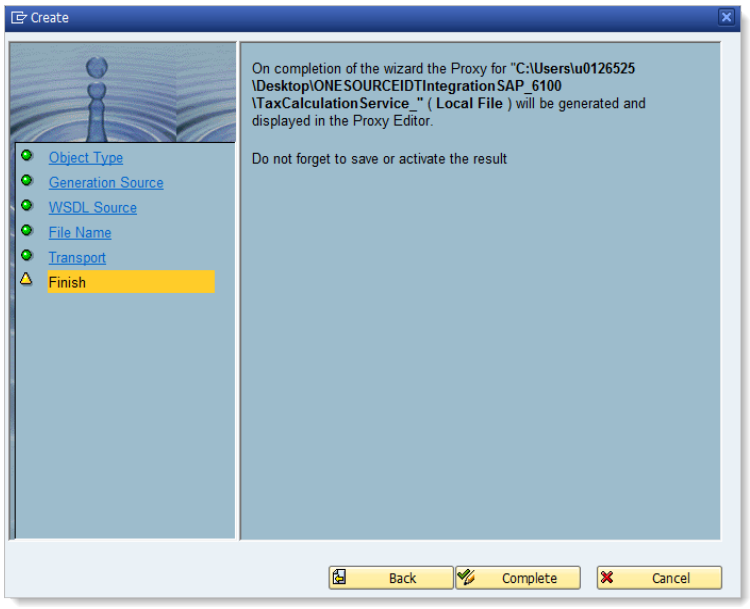
11. Enter the package you are using in the Package field. The prefix can be something like Z1 (The prefix can be up to 4 characters in length). The Request/Task could be the transport you are currently using (With
<F4> you can see what transports are available.).



Note: Important: Do not use more than 4 characters in the prefix! What we have found is that using more than four characters may screw up the proxy and you start getting strange and incorrect XML log line identifiers within the XML logs. Errors may occur if you use more characters.

12. Then press Cont icon, which should take you to a confirmation screen.

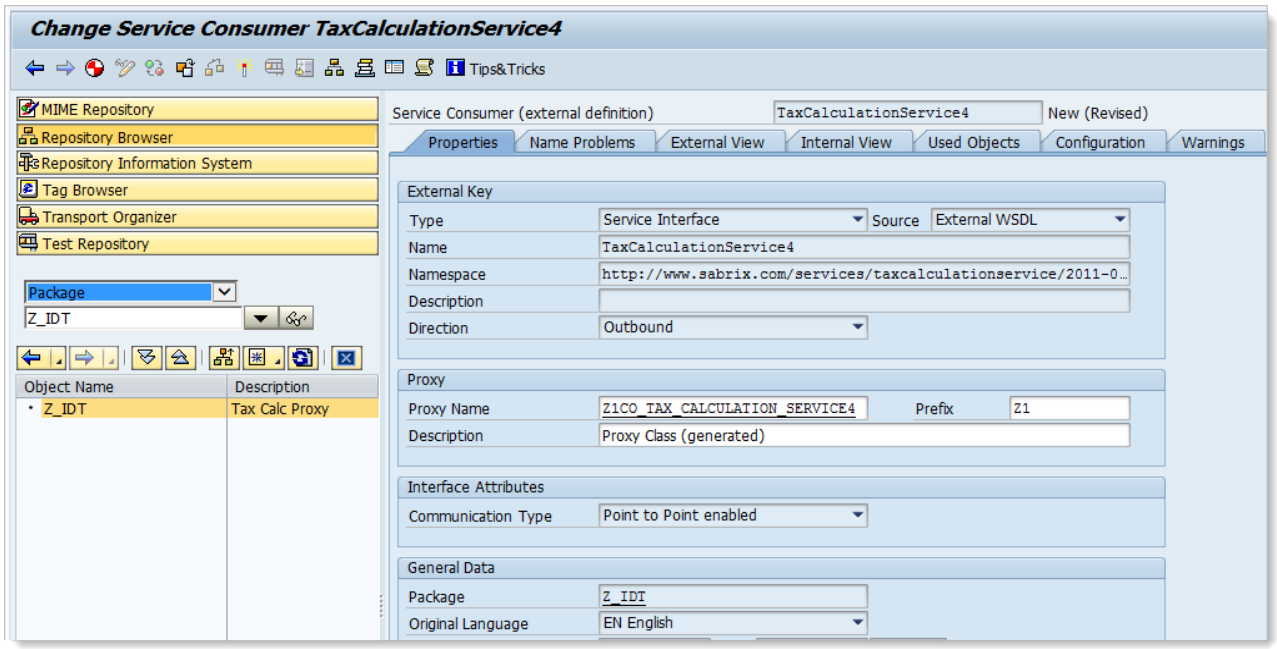
Note: Important: Make special note to what you selected as the prefix as you will have to update the D_STRUCTS table if you use a prefix other than Z1.



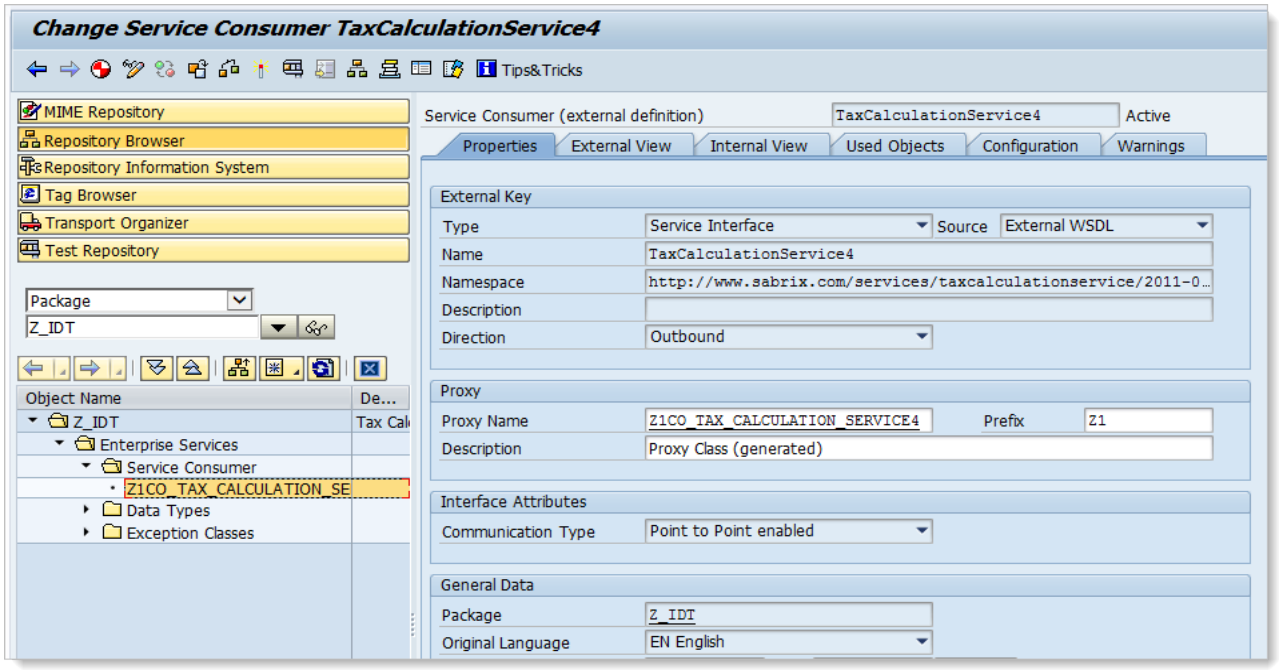
13. Select Complete icon to finish setting up the proxy.

Note: you might receive a warning message by SAP that the system is trying to access a local file, confirm and proceed.

After successful import and creation of the proxy you will be taken back to the Enterprise Service overview page. SAP has assigned a Proxy Name for you.



14. SAVE and/or ACTIVATE the Proxy (this might take several minutes).



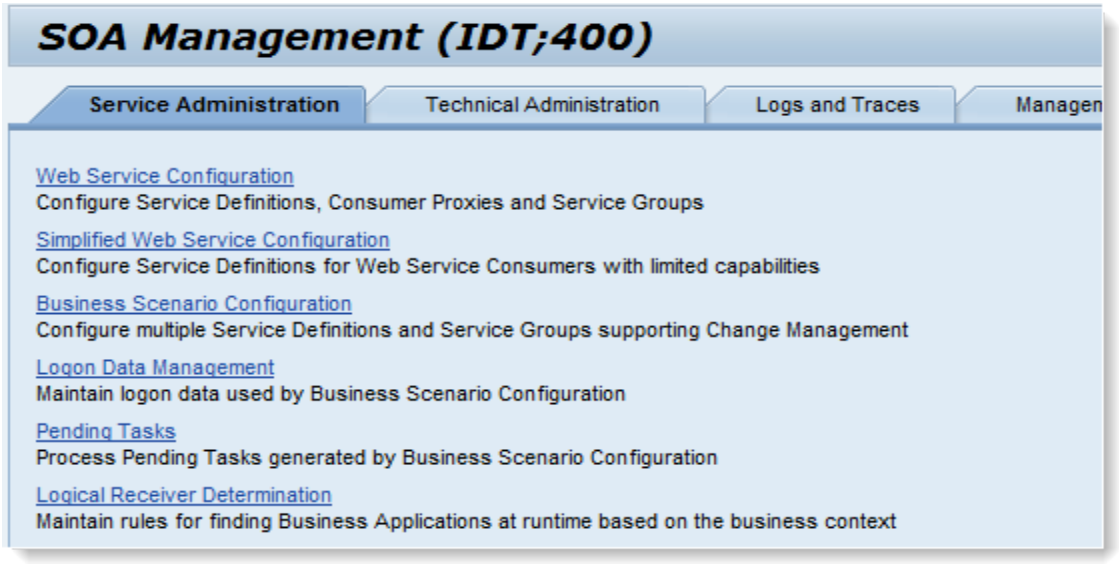
Write down the Proxy Name (in this case Z1CO_TAX_CALCULATION_SERVICE4) and the Service Consumer (in this case TaxCalculationService4).

Settings Up SOAMANAGER

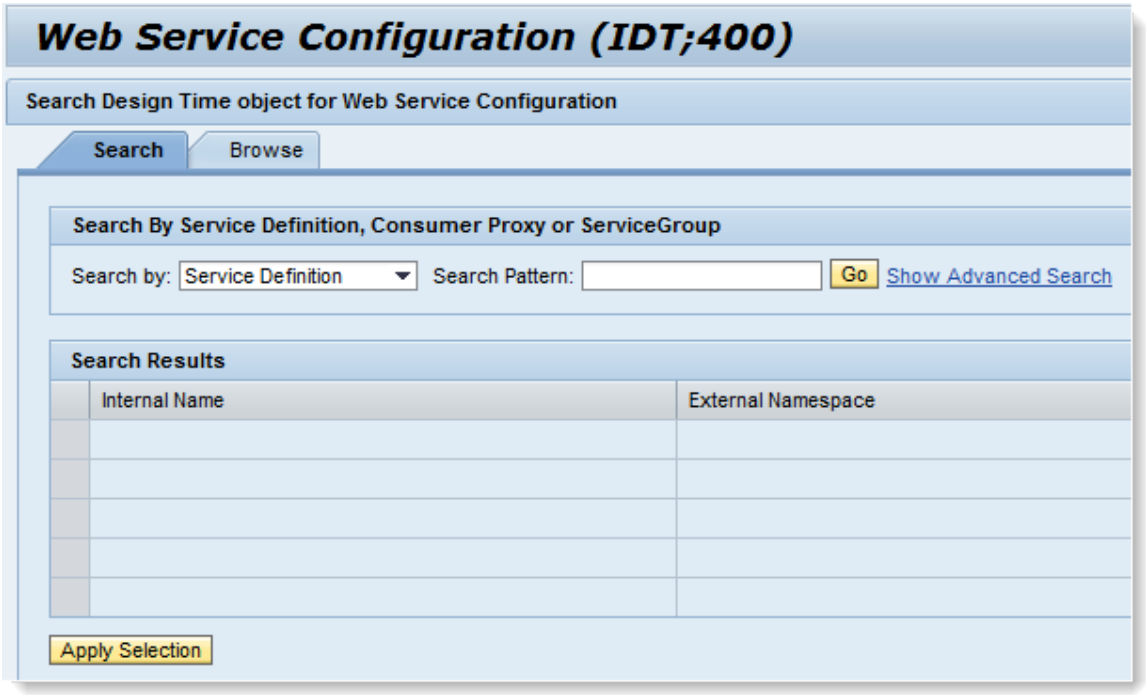
In the next steps we will setup a communication between SAP and Determination via SOAP. This is done in the SOA Management console.

Transaction Code SOAMANAGER Note: your screens may look different because of changes in each SAP release)

1. Logon to the web portal.



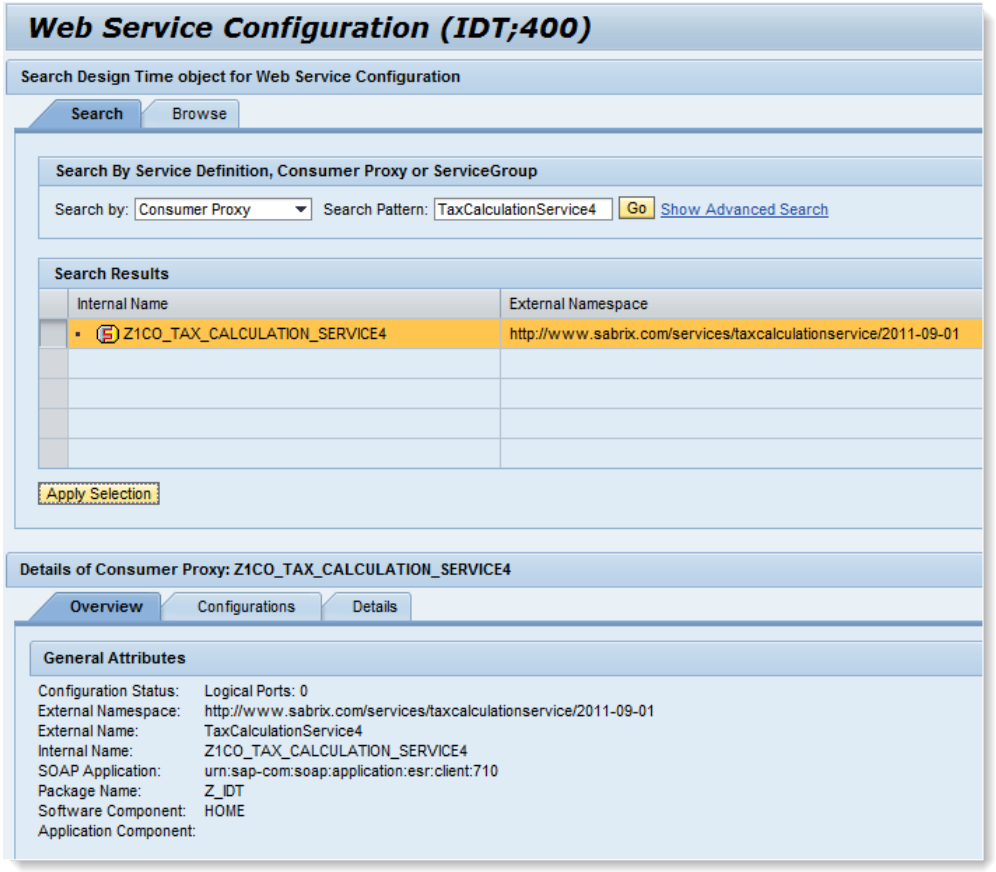
2. Go to the Service Administration tab, select Web Service Configuration.

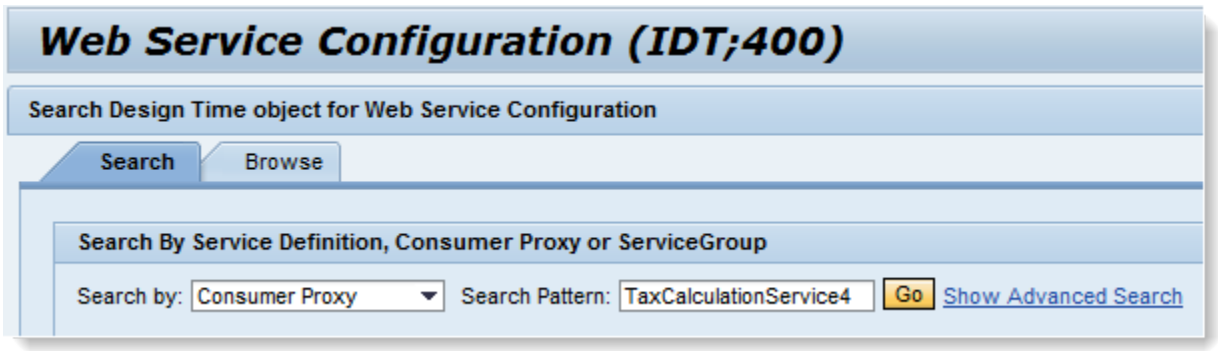


3. In the Search by drop-down select Consumer Proxy. Then enter in the Search Pattern: field the Service Consumer name created in the prior section (in this example TaxCalculationService4). Select Go.

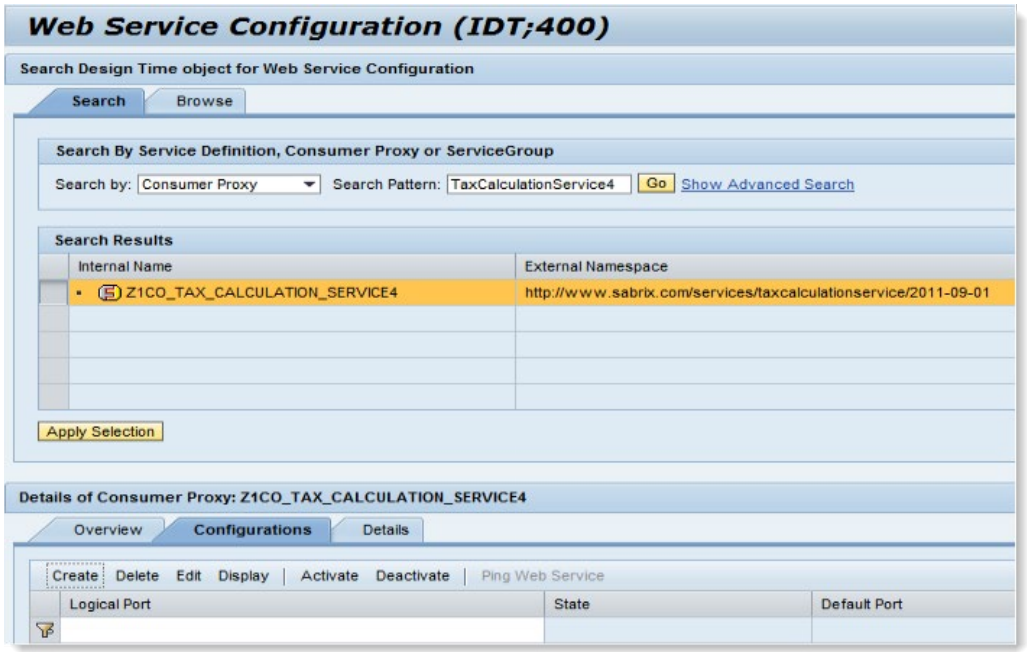
You should see a results list like the screen shot below.

4. Select the entry that matches your Internal Name of the proxy, and then click on Apply Selection.

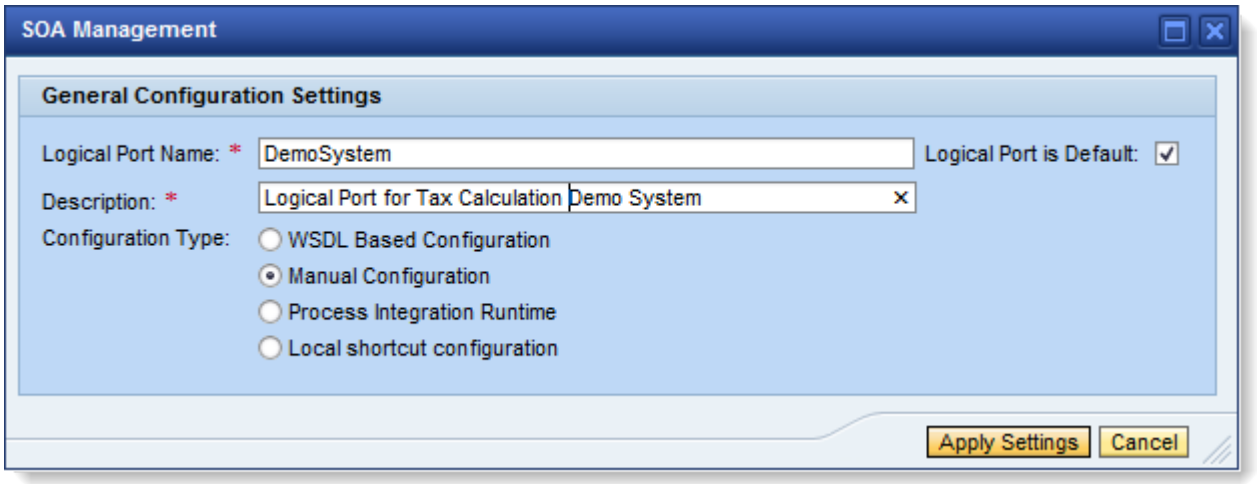




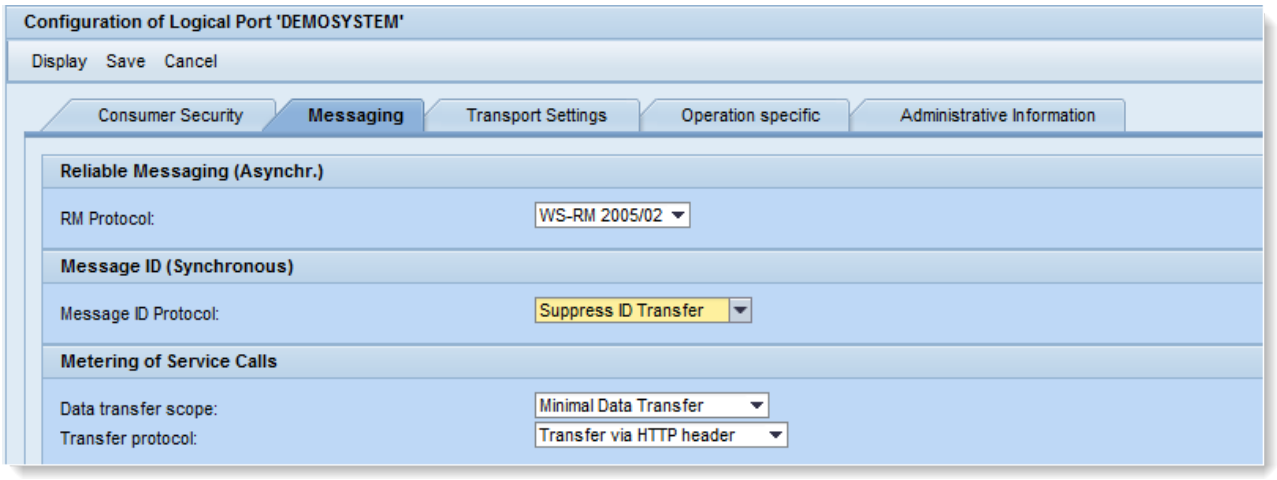
5. Move to the Configurations tab.



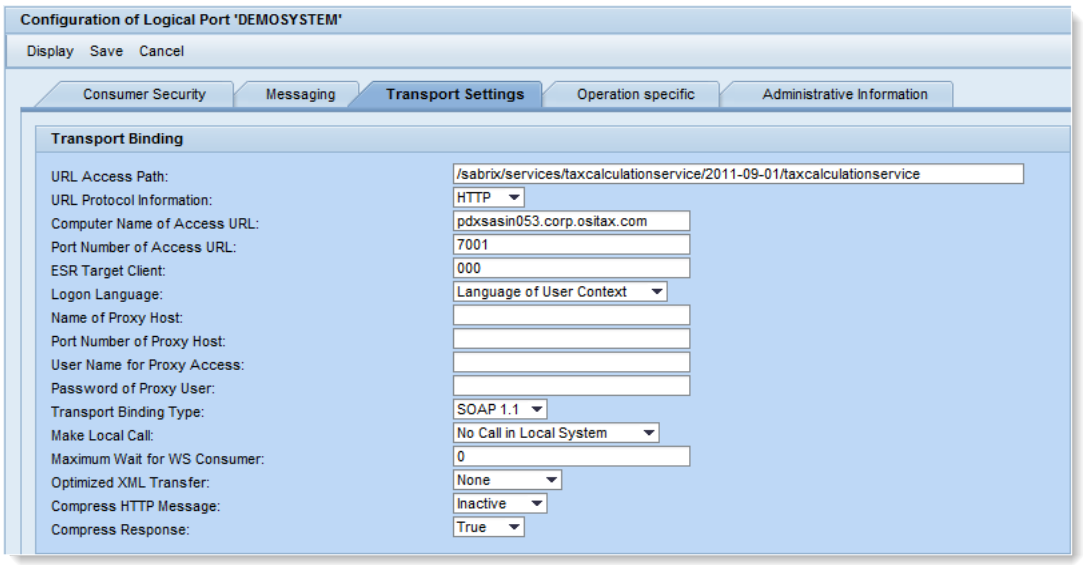
6. Select Create. A new pop-up will be displayed. Fill in the following values: Logical Port Name: Name of the port you want to create
   Description: Text explaining the use of the port Configuration Type: Select Manual Configuration



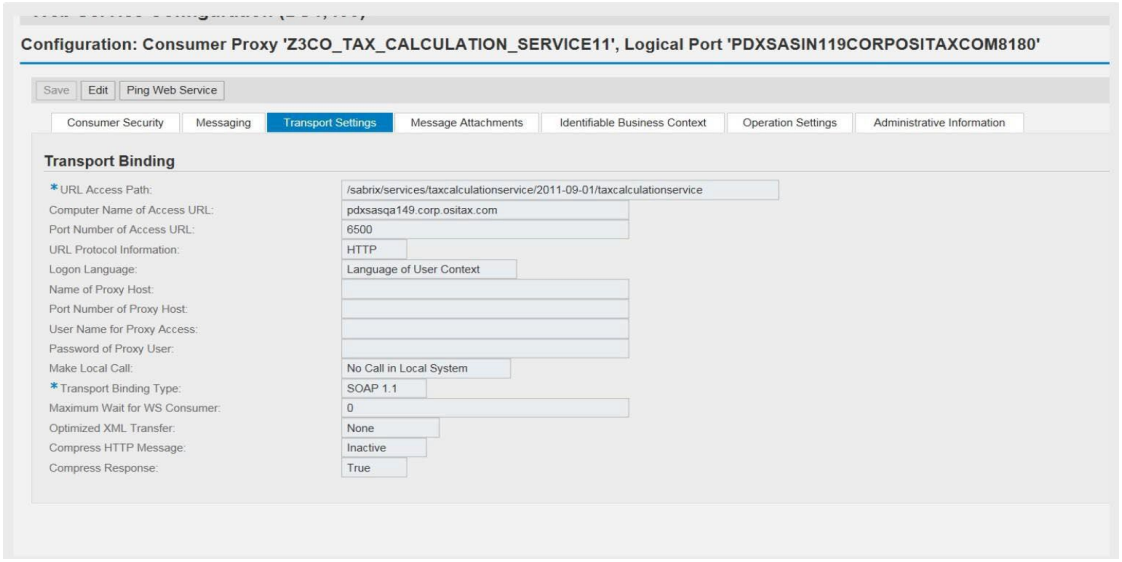
7. Apply Settings. You will be taken back to the Web Service Configuration Screen. In the bottom panel select the Messaging tab and switch the Message ID Protocol: to Suppress ID Transfer.



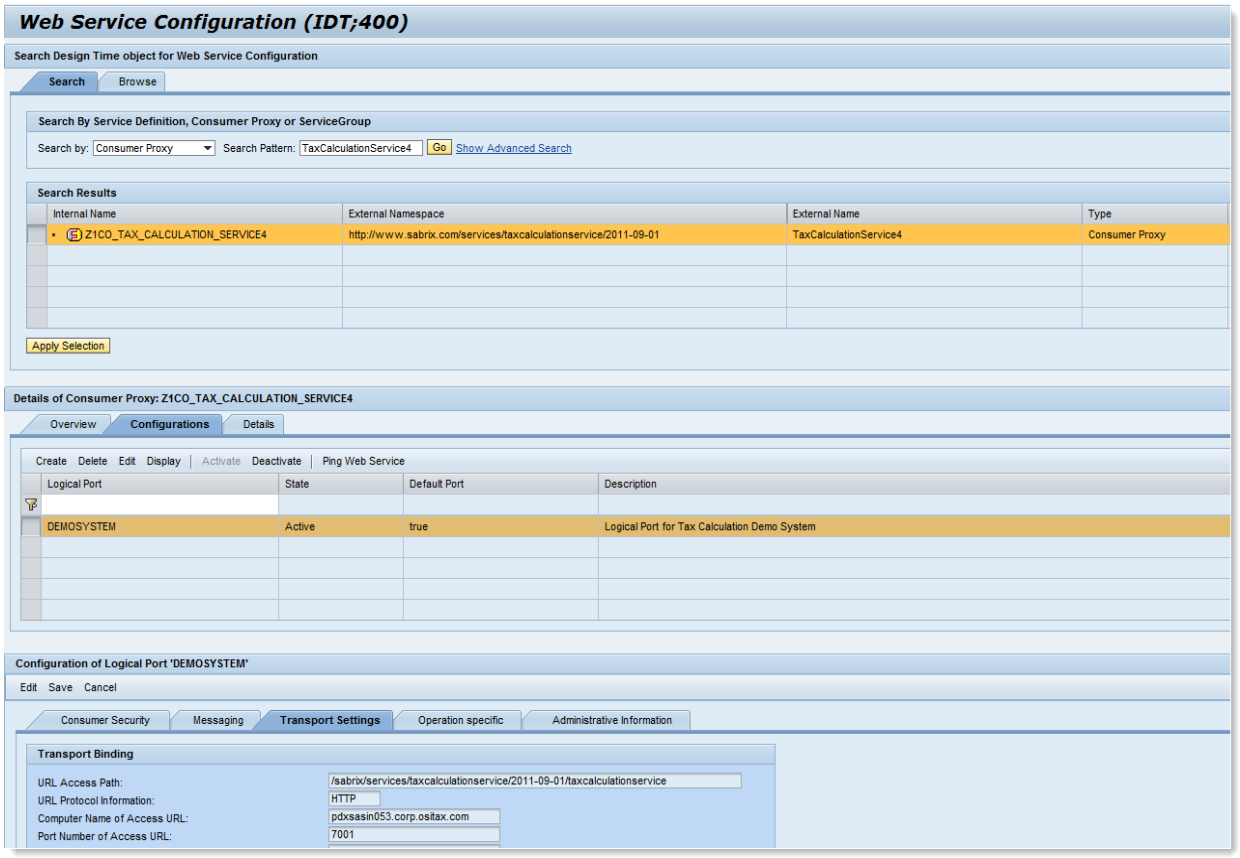
8. Go to the Transport Settings tab and enter the following values:
   URL Access Path: /sabrix/services/taxcalculationservice/2011-09- 01/taxcalculationservice
   Computer Name of Access URL: <name of determination server on the network>
   Port Number of Access URL: <port of determination server on the network>



If you have 2 different WSDL, you will need 2 different URL, etc



9. Save your setup. A confirmation message will be displayed. Your Web Service Configuration should look something like the screen shot below.



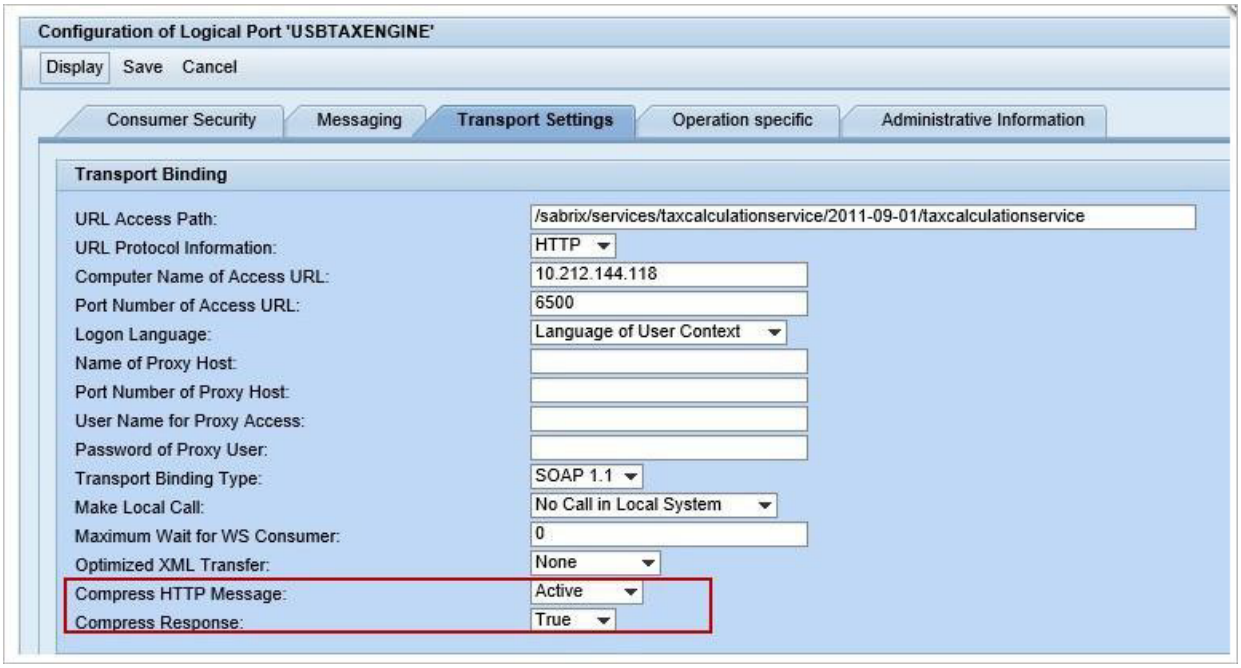
WS SECURITY CONSIDERATIONS ON THE PROXY FOR A HOSTED ENVIRONMENT

The above Proxy Configuration steps may need to be adjusted if you wish to use HTTPS and add security measures to the Proxy. As part of our offering we have created a BADI that can be instantiated and used by the customer to add a security name and password to the proxy communication between Integration and Determination SOAP calls. This will likely be critical for hosted environments.

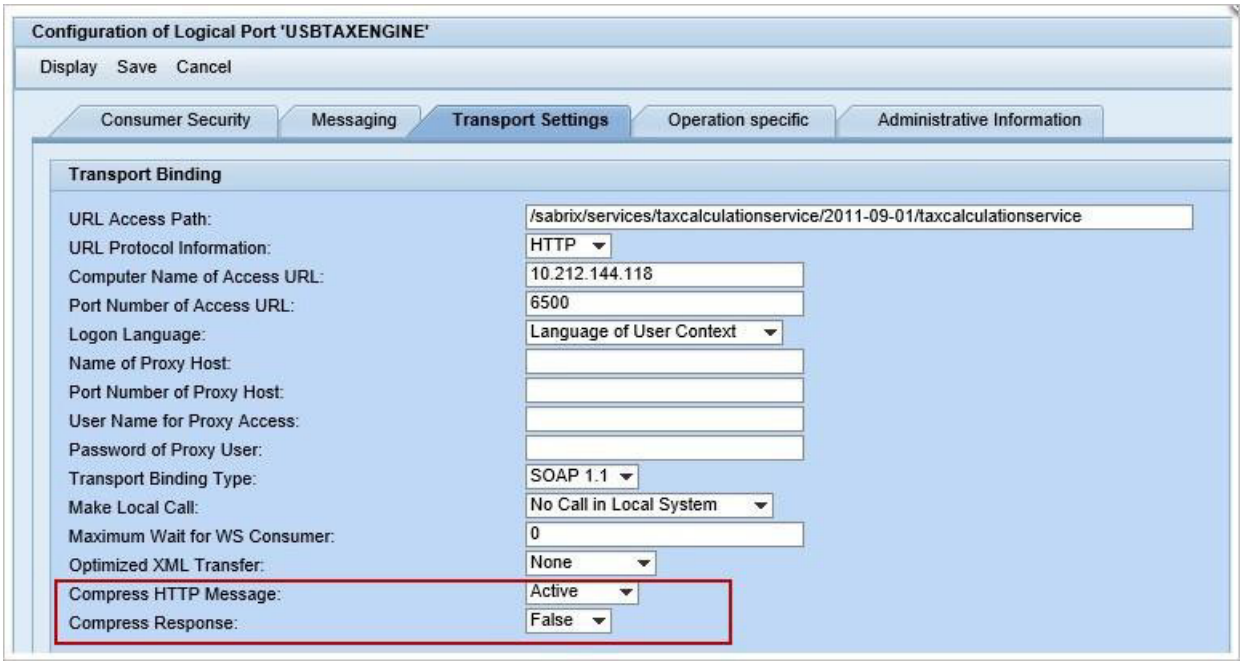
In this section we refer to step 8 in the above configuration as well as steps to activate and use the BADI for your specific security needs. If you wish to utilize GZIP for the compression of the HTTP messages, there are two different options available at the bottom of the screen: “Compress HTTP Message” and “Compress Response”.

Option 1: Using the compression on both the message and the response.

Example screen shot shown below for Proxy Configuration from step 8 on the prior page:

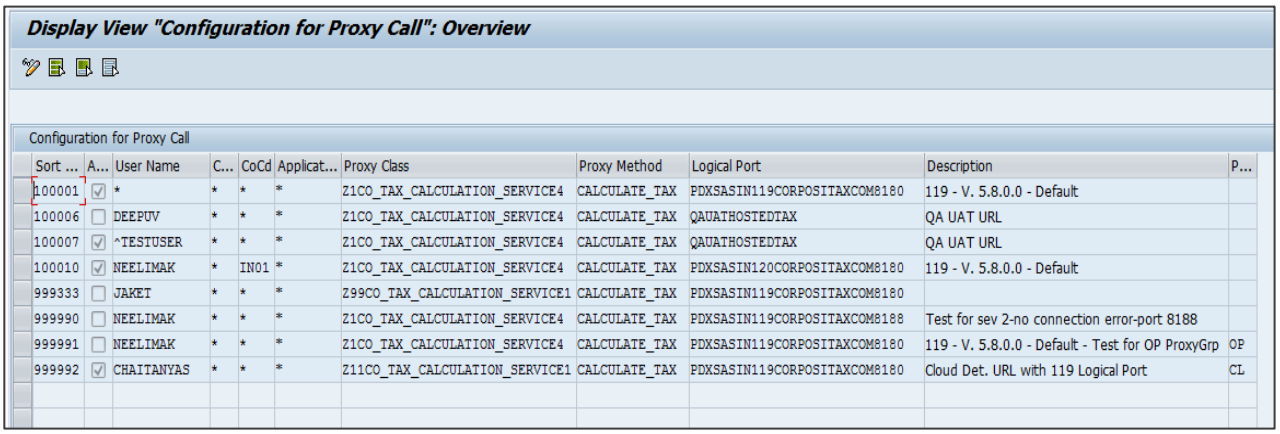


Option 2: Using the compression only on the message. Example of screen shot is shown below:



A transaction code: /N/IDT/WS has also been created for the user to establish the special username and password for each line on the Proxy Configuration table based on the sort order number of the proxy configuration line. The data in the transaction is obfuscated and placed in table /IDT/D_WS. This is an optional configuration step that is utilized by the Security BADI on the proxy. If the table is not populated, then that BADI will ignore the security check. We recommend that you also review the Determination documentation on setting up the security on the Determination side as these two functions must work together.

Below is an example of the Proxy Configuration table.

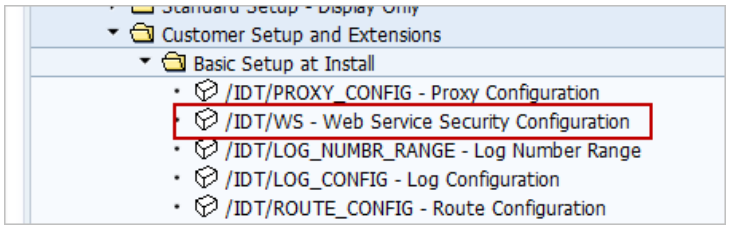


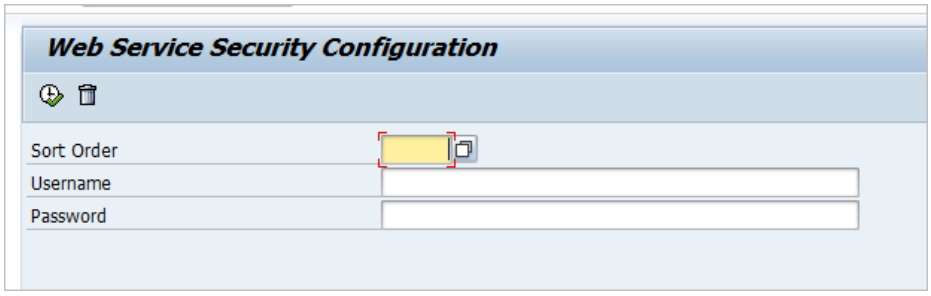
Sort ...	A...	User Name	C...	CoCd	Applicat...	Proxy Class	Proxy Method	Logical Port	Description	P...
100001	☑	*	*	*	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	PDXSASIN119CORPOSITAXCOM8180	119 - V. 5.8.0.0 - Default	
100006	☐	DEEPUV	*	*	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	QAUATHOSTEDTAX	QA UAT URL	
100007	☑	^TESTUSER	*	*	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	QAUATHOSTEDTAX	QA UAT URL	
100010	☑	NEELIMAK	*	IN01	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	PDXSASIN120CORPOSITAXCOM8180	119 - V. 5.8.0.0 - Default	
999333	☐	JAKET	*	*	*	Z99CO_TAX_CALCULATION_SERVICE1	CALCULATE_TAX	PDXSASIN119CORPOSITAXCOM8180		
999990	☐	NEELIMAK	*	*	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	PDXSASIN119CORPOSITAXCOM8188	Test for sev 2-no connection error-port 8188	
999991	☐	NEELIMAK	*	*	*	Z1CO_TAX_CALCULATION_SERVICE4	CALCULATE_TAX	PDXSASIN119CORPOSITAXCOM8180	119 - V. 5.8.0.0 - Default - Test for OP ProxyGrp	OP
999992	☑	CHAITANYAS	*	*	*	Z11CO_TAX_CALCULATION_SERVICE1	CALCULATE_TAX	PDXSASIN119CORPOSITAXCOM8180	Cloud Det. URL with 119 Logical Port	CL

See configuration guide explanation of the column and configuration options.

Note: the sort order field in the table as you will need to use the transaction code to insert a new username and password for this line on the Proxy Configuration table. This is part of the Customer set up menu as shown below:

Transaction Code**: /N/IDT/WS**





Enter the sort order number from the Proxy table for the connection that you wish to use the security logic and then add USERNAME and PASSWORD that the system will use to verify the connection. EXECUTE to save the setup to table /IDT/D_WS. Since the USERNAME and PASSWORD that you enter for this security check are saved in an encrypted form, the original amount you enter is not readable or able to be deleted via SE16.

Optional WS Security BADI for the Proxy

/IDT/BADI_ADJUST_PROXY

This BADI is provided with the install and can be used if you wish to utilize the security process outlined above. This BADI can be used for the proxy security steps as outlined here or may be also utilized by the customer for other possible ABAP programming for security options as needed for your specific environment.

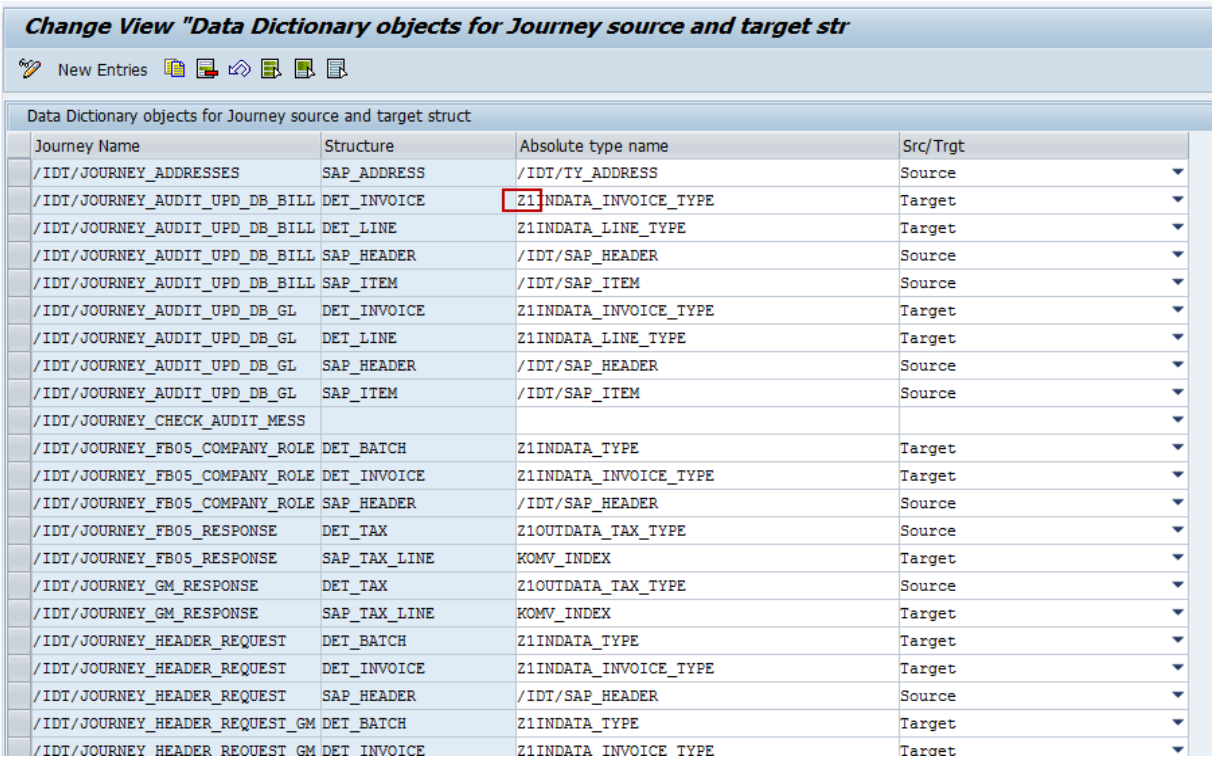
STRUCTURE TABLE MODIFICATION AFTER PROXY SETUP OR CUSTOM JOURNEY CREATION.

The /IDT/D_STRUCTS table is used to support the selection of drop-down values within the field mapping configuration screen. The table /IDT/D_STRUCTS is supplied with standard configuration as part of the configuration transport to install or upgrade Integration. There are line-items which reference the proxy and they must be adjusted to the Absolute Type Name used on the proxy being used. In our proxy set up instructions we recommend using Z1 as the prefix for the proxy name. This table is pre-configured based on the assumption you use our recommended Z1 prefix. If you change this to your own assignment (up to four characters) then this step will be required to put this table in sync with your proxy installation.

Transaction code: SM30 access table /IDT/D_STRUCTS

Access /IDT/D_STRUCTS table from SM30 and update the Z1 value in the Absolute Type Names (highlighted in red in the example below) and update with your proxy's name (prefix) if it is different from Z1 in your system. Do this for all lines in the table that have Absolute Type Name starting with Z1. Do not change any of the values that start with /IDT/. Save the table with these changes in place.

Note also that if you add a new custom journey to the system you will have to maintain this table as well with the new journey name, structure, etc.



Change View "Data Dictionary objects for Journey source and target str

New Entries

Data Dictionary objects for Journey source and target struct

Journey Name	Structure	Absolute type name	Src/Trgt
/IDT/JOURNEY_ADDRESSES	SAP_ADDRESS	/IDT/TY_ADDRESS	Source
/IDT/JOURNEY_AUDIT_UPD_DB_BILL	DET_INVOICE	Z1INDATA_INVOICE_TYPE	Target
/IDT/JOURNEY_AUDIT_UPD_DB_BILL	DET_LINE	Z1INDATA_LINE_TYPE	Target
/IDT/JOURNEY_AUDIT_UPD_DB_BILL	SAP_HEADER	/IDT/SAP_HEADER	Source
/IDT/JOURNEY_AUDIT_UPD_DB_BILL	SAP_ITEM	/IDT/SAP_ITEM	Source
/IDT/JOURNEY_AUDIT_UPD_DB_GL	DET_INVOICE	Z1INDATA_INVOICE_TYPE	Target
/IDT/JOURNEY_AUDIT_UPD_DB_GL	DET_LINE	Z1INDATA_LINE_TYPE	Target
/IDT/JOURNEY_AUDIT_UPD_DB_GL	SAP_HEADER	/IDT/SAP_HEADER	Source
/IDT/JOURNEY_AUDIT_UPD_DB_GL	SAP_ITEM	/IDT/SAP_ITEM	Source
/IDT/JOURNEY_CHECK_AUDIT_MESS			
/IDT/JOURNEY_FB05_COMPANY_ROLE	DET_BATCH	Z1INDATA_TYPE	Target
/IDT/JOURNEY_FB05_COMPANY_ROLE	DET_INVOICE	Z1INDATA_INVOICE_TYPE	Target
/IDT/JOURNEY_FB05_COMPANY_ROLE	SAP_HEADER	/IDT/SAP_HEADER	Source
/IDT/JOURNEY_FB05_RESPONSE	DET_TAX	Z1OUTDATA_TAX_TYPE	Source
/IDT/JOURNEY_FB05_RESPONSE	SAP_TAX_LINE	KOMV_INDEX	Target
/IDT/JOURNEY_GM_RESPONSE	DET_TAX	Z1OUTDATA_TAX_TYPE	Source
/IDT/JOURNEY_GM_RESPONSE	SAP_TAX_LINE	KOMV_INDEX	Target
/IDT/JOURNEY_HEADER_REQUEST	DET_BATCH	Z1INDATA_TYPE	Target
/IDT/JOURNEY_HEADER_REQUEST	DET_INVOICE	Z1INDATA_INVOICE_TYPE	Target
/IDT/JOURNEY_HEADER_REQUEST	SAP_HEADER	/IDT/SAP_HEADER	Source
/IDT/JOURNEY_HEADER_REQUEST_GM	DET_BATCH	Z1INDATA_TYPE	Target
/IDT/JOURNEY_HEADER_REQUEST_GM	DET_INVOICE	Z1INDATA_INVOICE_TYPE	Target

If you are using the Cloud version of the Proxy and XSD then you will want to have the drop down, use the 2018.3 Cloud Proxy's absolute type name so that the additional fields will show in your dropdown screens in the field mapper table. If you are using both Cloud and On-Premise Determination, then select the Cloud version here but you will need to remember that some fields in the field mapping drop down lists will not be correct for a On Premise mapping selection.

UPDATES TO CUSTOM TABLES

Configuration DETERMINE_CONDITION_TYPE_FOR_TAXES

Note: After populating this table during a new installation, ensure you review it with each product update. This will help you identify and add any new routes introduced in the latest release

Example: Separate routes were added for Settlement Management:

/IDT/ROUTE_GROUP_BILLING_SM_SD

/IDT/ROUTE_GROUP_BILLING_SM_MM

And one route for Deferred tax Reverse charge scenario

/IDT/ROUTE_NON_GROUP_DOC_DT_RC

Three lines for each route would have to be added: one for Fee, Percentage, and Exempt statuses for Nature of Tax along with the correct condition type.

CLEARING OF DYNAMICALLY GENERATED PROGRAMS

After you have completed the above steps you must run this program to clear all the dynamically generated programs that are created behind the scenes for the field mapping table. If any changes to the functions of field mapping are included as part of this release, then the clearing of these programs will ensure that they get the new changes picked up into the program logic next time the field mappings are used and the journeys utilized.

Transaction code: SE38 /IDT/DEL_ALL_DYN_PROGS

After executing this program make sure that table that contains the dynamically created field mapping programs is empty and does not contain any remaining items.

Transaction SE16: /IDT/D_DYN_PROG

Check this table and make sure it is blank. There shouldn't be any program starting with /IDT/DG. Once this table has been cleared, new programs will be generated and entered into this table once transactions start again, and field mapping logic is used.

Note: dynamically generated programs are created for each client version on a given system. However, this clearing program only clears the current client. You will need to run this program again in each client of your system.

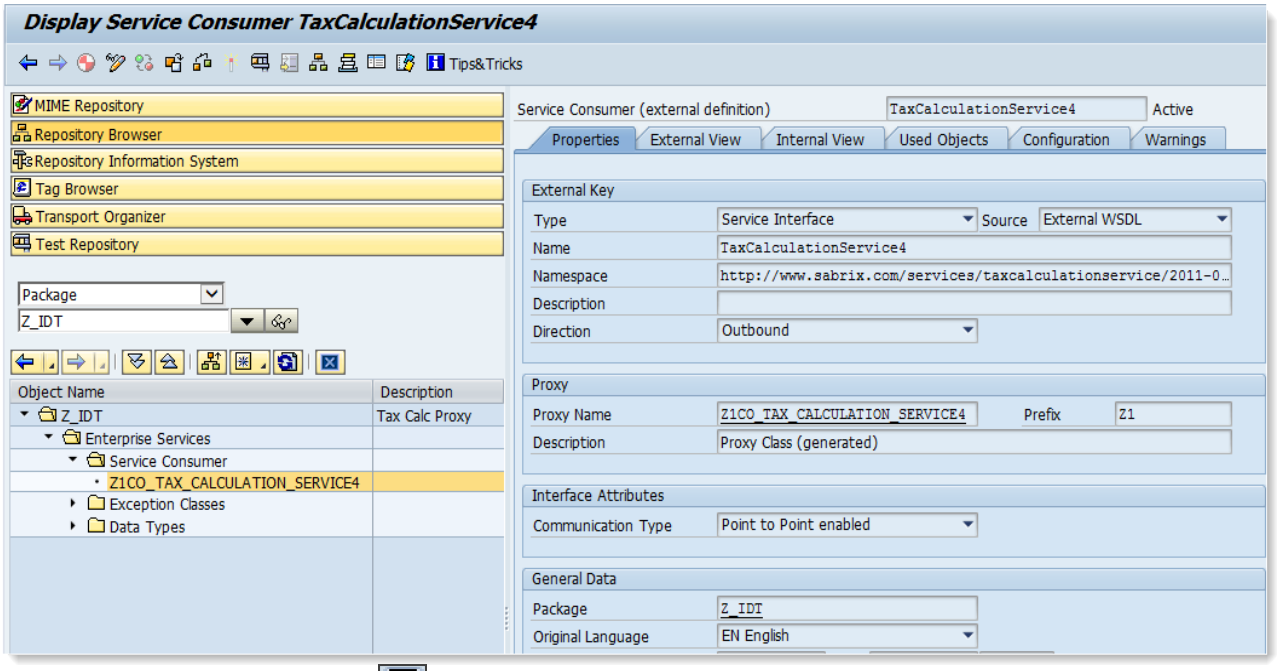
SOAP TAX CALCULATIONS TEST

In this section we will make a SOAP test call to Determination using the Proxy we setup above. This will confirm communication between SAP and Determination. It is a prerequisite for a later successful tax calculation from SAP business transactions that the communication layer between the two systems works.

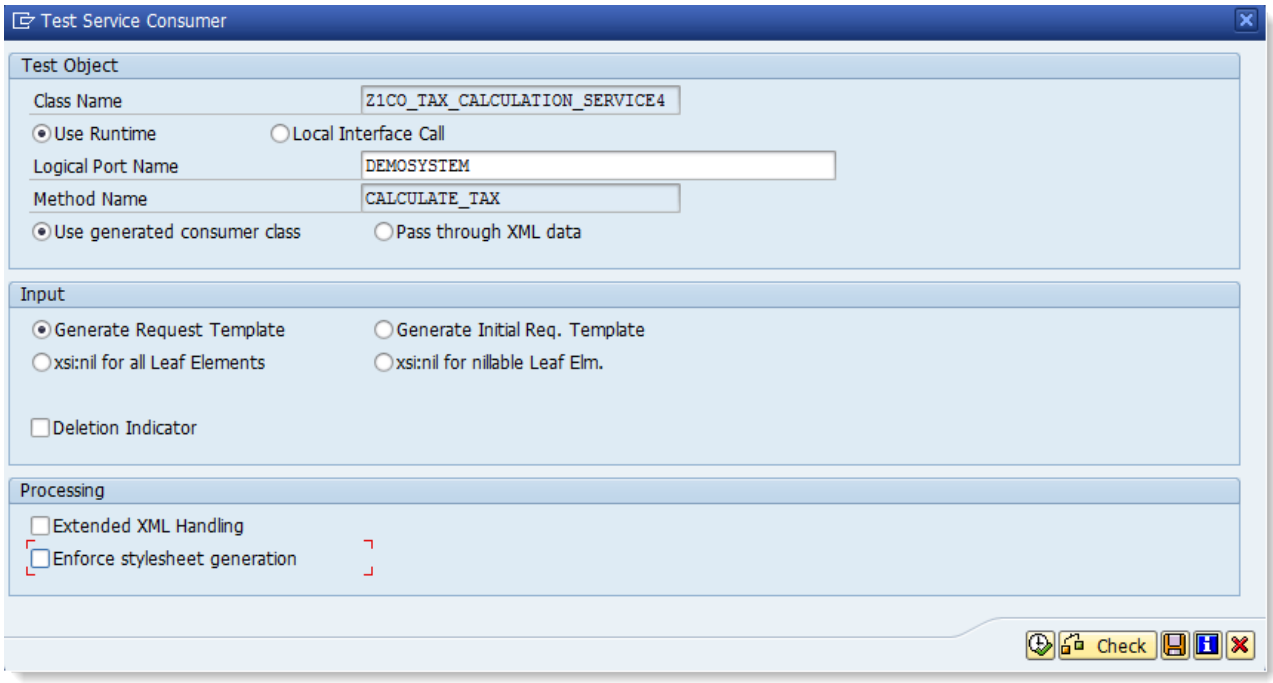
SOAP Testing

Transaction Code SE80

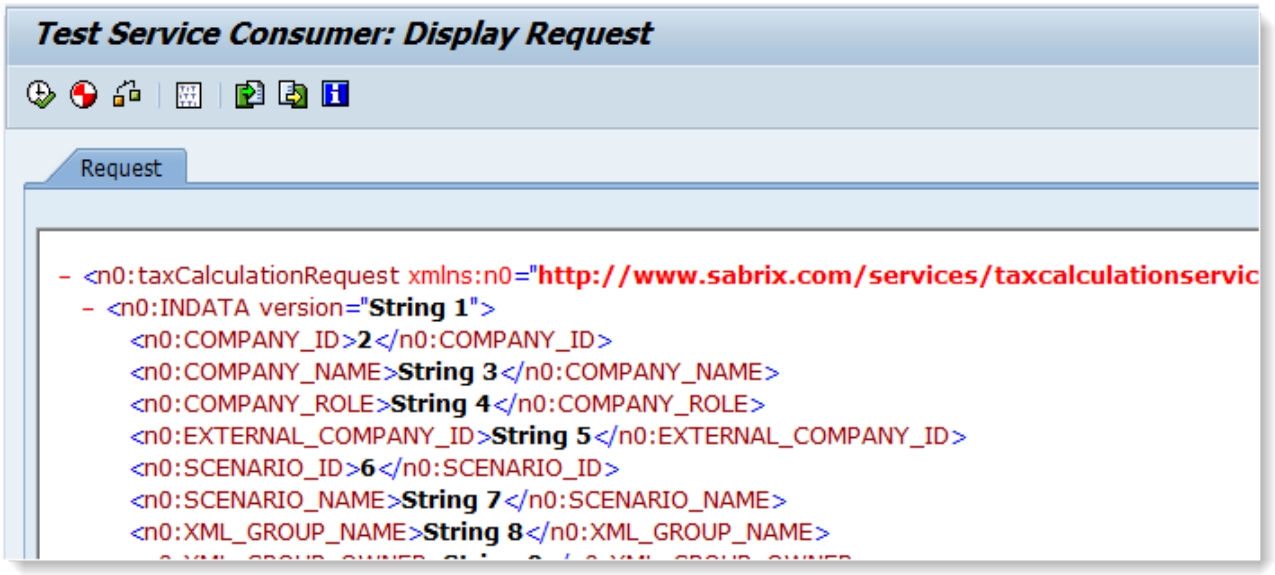
1. Select the Package name you created during the SAP Proxy setup (Z_IDT in this example), then drill down to Enterprise Service ,Service Consumer to see your consumer proxy. Then double click the proxy.



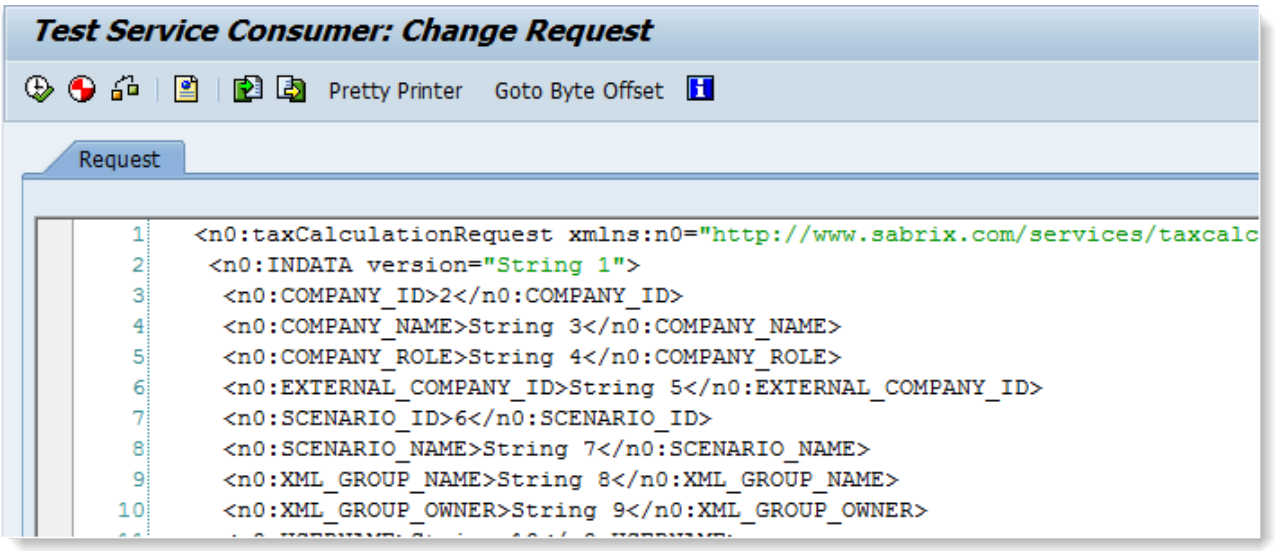
2. Press the Test icon  . On the pop-up select the Logical Port you created in prior steps.



3. Select the Execute icon which will take you to the Test Service Consumer screen.



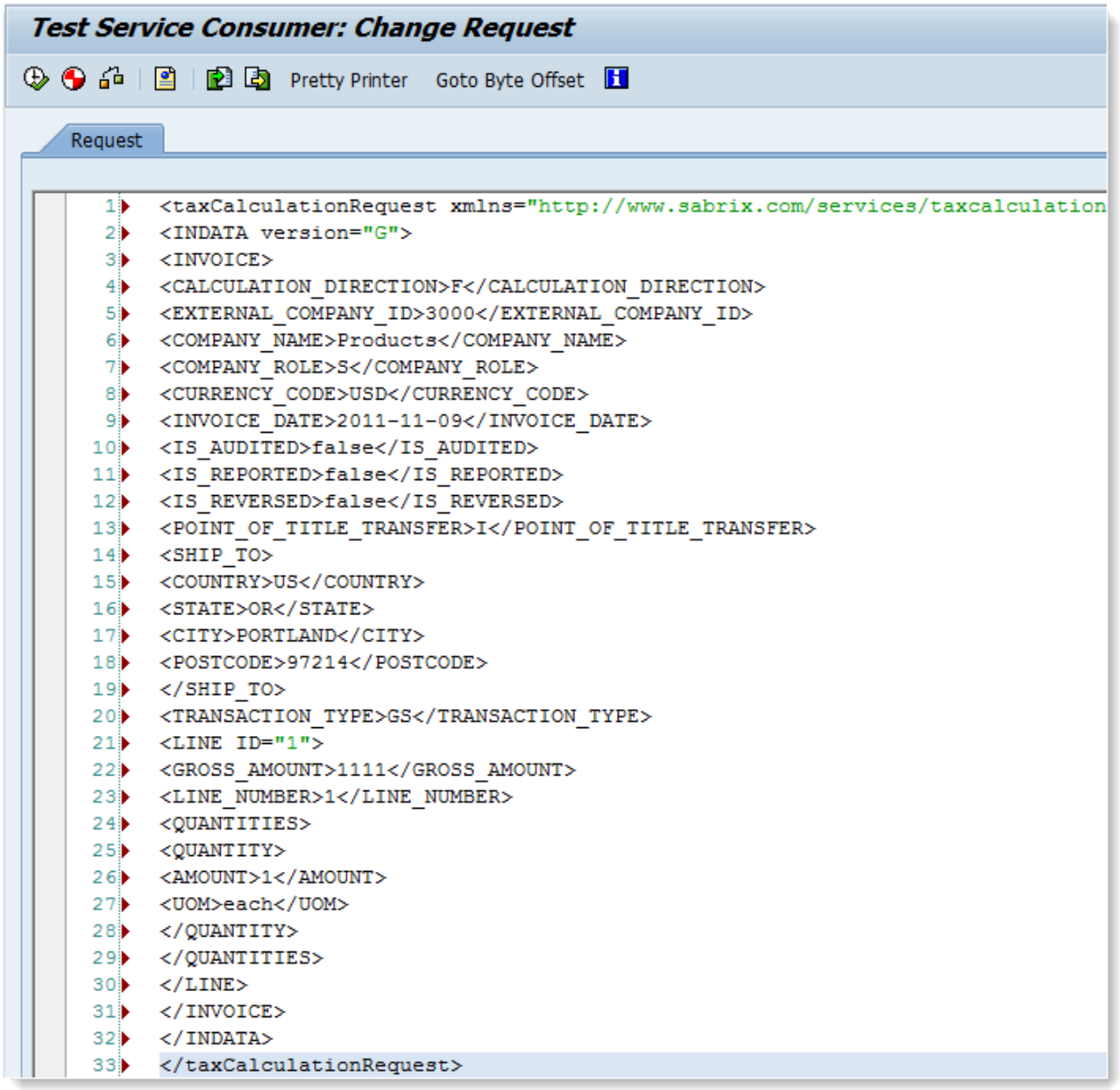
4. Select the XML Editor icon  to change the request data.



5. Use CTRL+A to select all data, then delete the test data defaulted by SAP. You should have an empty screen now.
6. Copy and paste the below sample calculation request into the SOAP test screen:

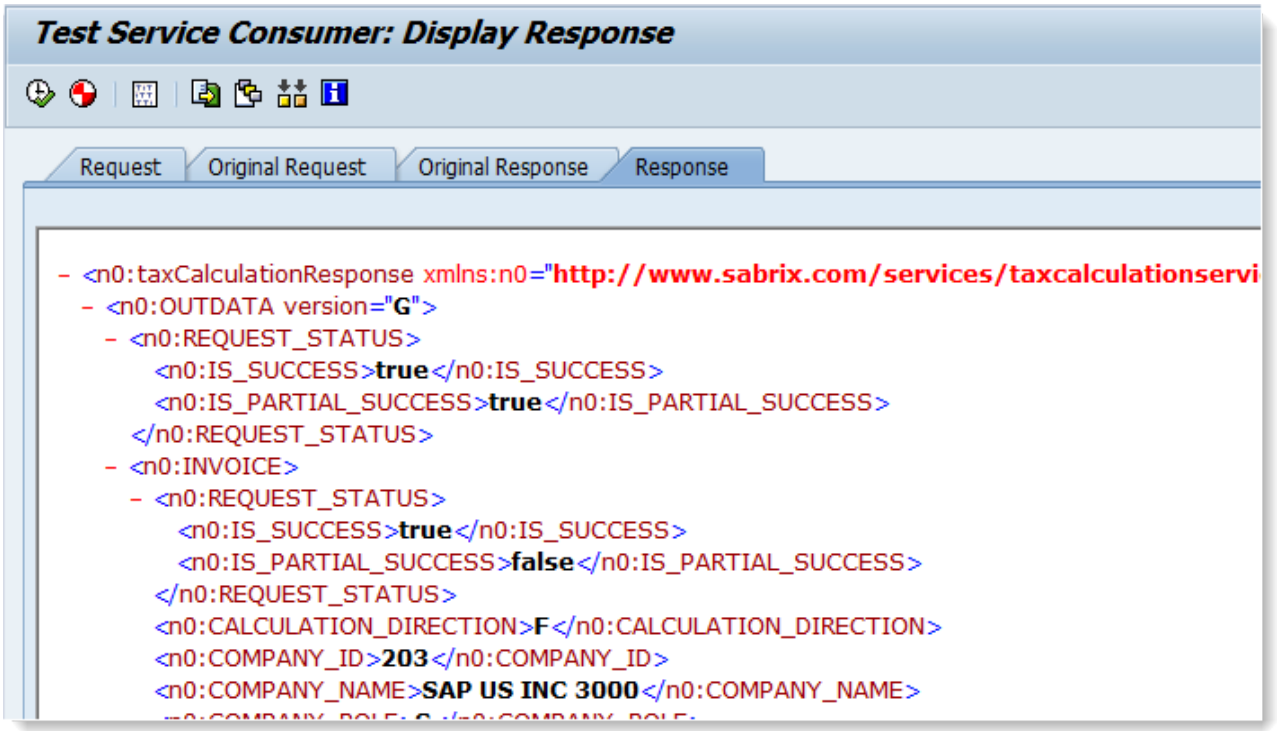
```
<n0:taxCalculationRequest xmlns:n0="http://www.sabrix.com/services/taxcalculationservice/2011-09-01">
<n0:INDATAversion="G">
<n0:INVOICE>
<n0:CALCULATION_DIRECTION>F</n0:CALCULATION_DIRECTION>
<n0:EXTERNAL_COMPANY_ID>OSAP-1000</n0:EXTERNAL_COMPANY_ID>
<n0:COMPANY_NAME>Products</n0:COMPANY_NAME>
<n0:COMPANY_ROLE>S</n0:COMPANY_ROLE>
<n0:CURRENCY_CODE>USD</n0:CURRENCY_CODE>
<n0:INVOICE_DATE>2011-11-09</n0:INVOICE_DATE>
<n0:IS_AUDITED>false</n0:IS_AUDITED>
<n0:IS_REPORTED>false</n0:IS_REPORTED>
<n0:IS_REVERSED>false</n0:IS_REVERSED>
<n0:POINT_OF_TITLE_TRANSFER>I</n0:POINT_OF_TITLE_TRANSFER>
<n0:SHIP_TO>
<n0:COUNTRY>US</n0:COUNTRY>
<n0:STATE>OR</n0:STATE>
<n0:CITY>PORTLAND</n0:CITY>
<n0:POSTCODE>97214</n0:POSTCODE>
</n0:SHIP_TO>
<n0:TRANSACTION_TYPE>GS</n0:TRANSACTION_TYPE>
<n0:LINEID="1">
<n0:GROSS_AMOUNT>1111</n0:GROSS_AMOUNT>
<n0:LINE_NUMBER>1</n0:LINE_NUMBER>
<n0:QUANTITIES>
<n0:QUANTITY>
<n0:AMOUNT>1</n0:AMOUNT>
<n0:UOM>each</n0:UOM>
</n0:QUANTITY>
</n0:QUANTITIES>
</n0:LINE>
</n0:INVOICE>
</n0:INDATA>
</n0:taxCalculationRequest>
```

7. Change the value for <EXTERNAL_COMPANY_ID> to a valid value (see your Determination expert if you do not know a value).



Note: You might want to save your test data in SAP for later use. You can do so by selecting the Save Variant icon, and then give it a name. You will be able to reload that test data later via the Get Variant icon .

8. Now run your test via the Execute icon. You should get a response shown in the Response tab.



Success of a tax calculation is indicated by the IS_SUCCESS and IS_PARTIAL_SUCCESS set to true. In this case you should also see tax details in the <TAX> block of the response.

The last step is to configure the tax interface to use the SOAP Proxy and Logical Port during SAP business transactions. You do this via transaction code /N/IDT/PROXY_CONFIG. You can find details on how to do this setup in the Configuration Guide section of ONESOURCE Proxy Configuration.

PROGRAMMER GUIDE

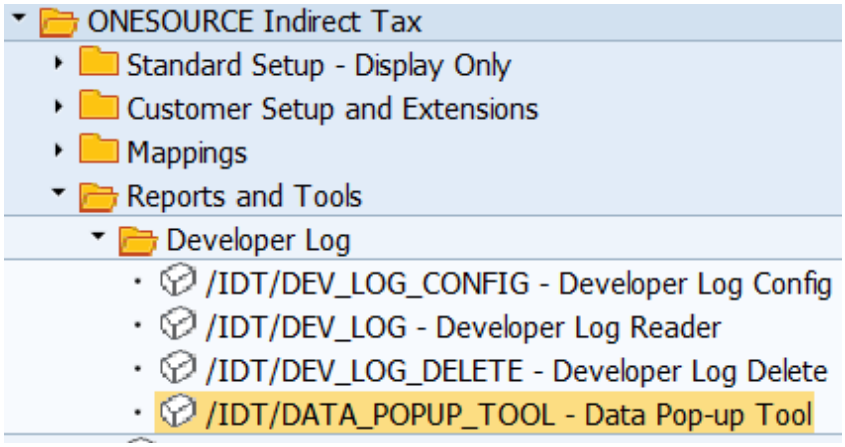
The Integration has been designed and built with extensibility and maintainability in mind. There are several areas the code can be enhanced by Thomson Reuters, Partners, or you our customers. Generally, our Professional Services team can assist in assessing your needs and determining if the provided framework allows meeting the need directly or if custom code should be developed. This section covers the most common needs customers have for creating custom code; user-exit based field mappings and custom address sources.

Data Popup Tool

The Data Popup Tool is a feature for users needing to do field mapping or debugging to determine which tables and fields are available and populated with data within the current transaction you are testing. It is a valuable tool to determine issues with a transaction or identify additional fields that will work for future or current field mapping. This tool is to be used only in a test environment as a tool for analysis of the transaction as it will interrupt the transaction and pop-up within the transaction you are testing when a call is made to Determination for a tax calculation. There is a configuration table that you access to turn on the feature within the Reports and tools menu that will allow you to target the data based on a single journey or set of journeys that are used within the specific transaction. If multiple journeys are selected the screen will advance to the next journey once you get out of the current journey pop-up and will continue to do so until you have gone through the list of journeys that you selected and are used in the transaction,

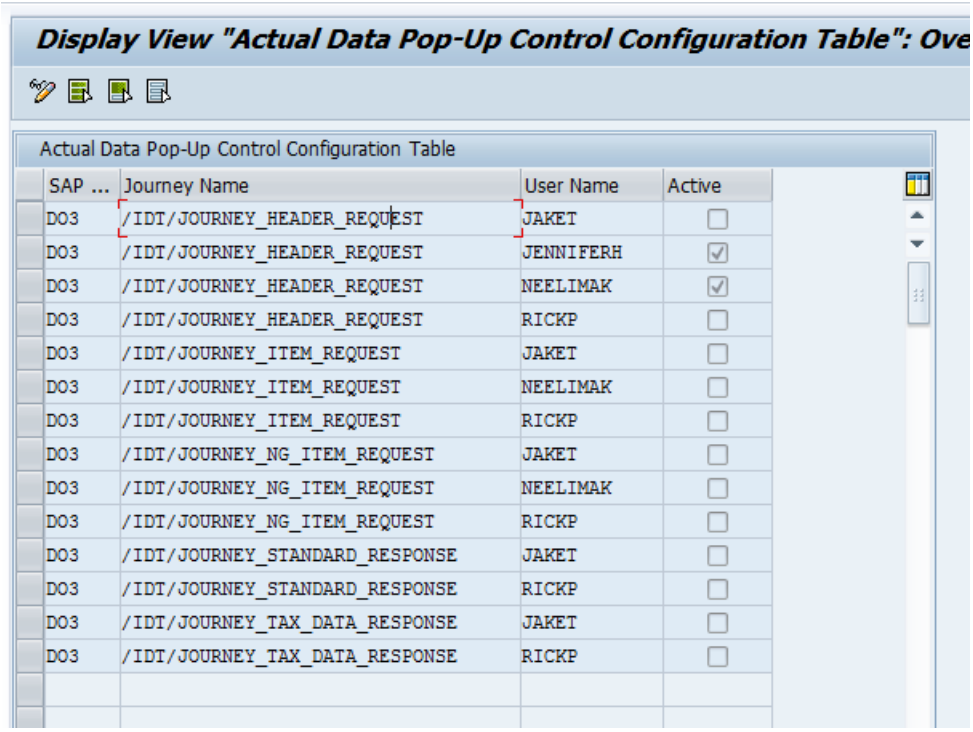
Some transactions make many calls to Determination during the transaction: a ME21N Purchase Order is especially invasive in this respect and can cause the system to literally loop again and again through this pop-up feature and not let you out of the transaction. If this occurs, simply start a second session and change/ deactivate the popup tool and return to your prior session. Doing this delete process will allow you to proceed with the transaction without further issues. See screen shots below to access this feature, configure it, and view how it works.

Transaction Code: **/N/IDT/DATA_POPUP_TOOL**

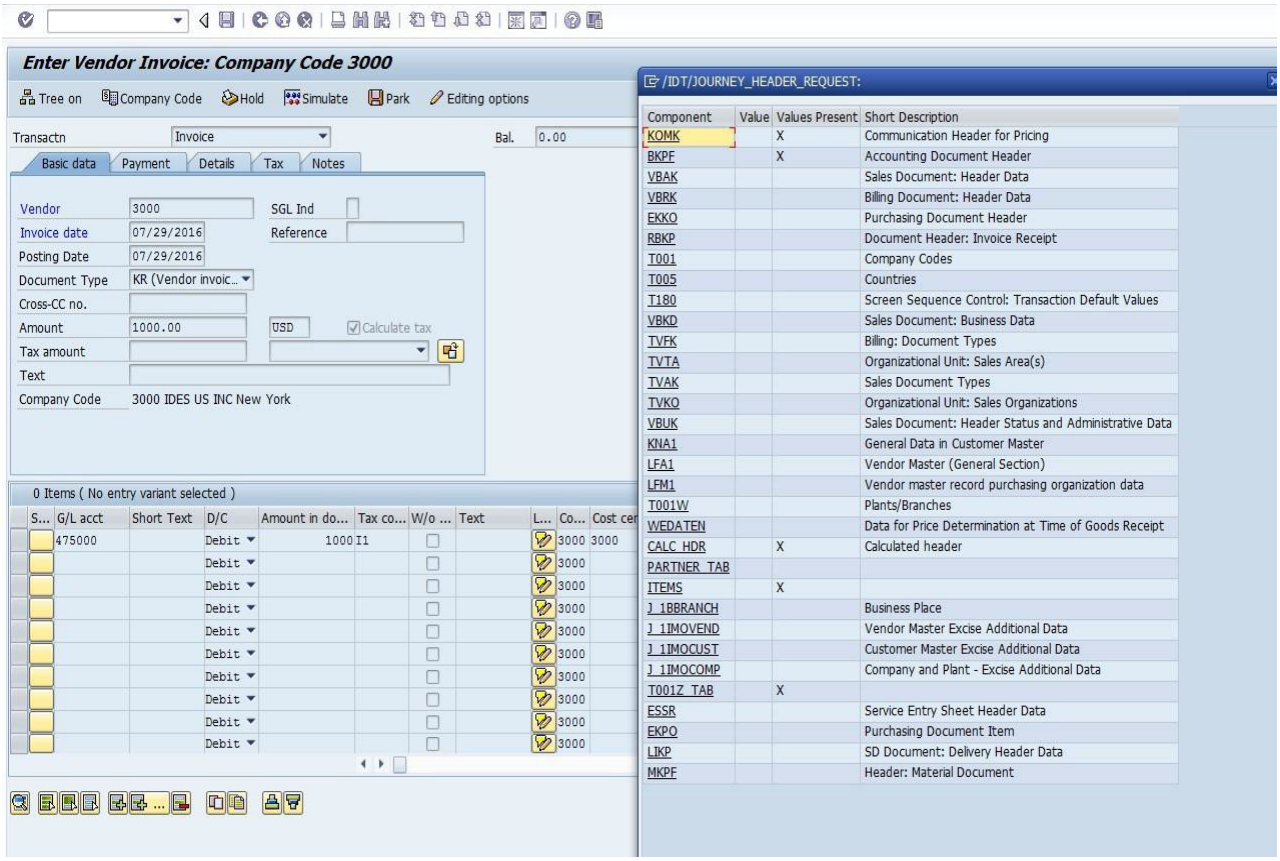


Once in the screen you will be able to change it to change mode and add lines to the table for your username and activate the journey. Currently the tool is available for the below list of journeys. More journeys will be added to this tool in the future and a drop-down list added for easier use. To deactivate a given journey, just return to the menu and turn off the active button. Journeys currently available include:

- Journey Header Request
- Journey Item Request
- Journey NG Item Request
- Journey Standard Response
- Journey Tax Data Response



Next go to the desired transaction you wish to use this tool on and enter your data. When a call to Determination is made within the transaction the tool will activate and pop-up the data for the first journey selected.



For example, we have done a FB60 invoice which will use the standard request, non-group line-item request, standard response, and tax data table response journeys. You see in the screen shot above that the header request journey displayed and shows the list of tables. An X in the Values Present column tells you there is data available within this table for this given transaction.

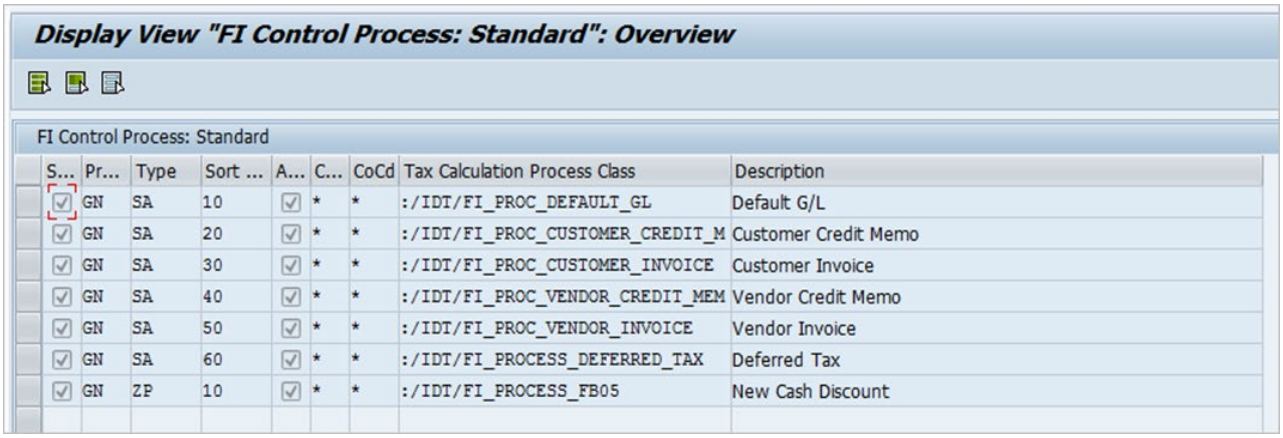
Double click on one of the tables listed and the popup will take you one level down into the actual field names and data populate within this transaction.

/IDT/JOURNEY_HEADER_REQUEST:KOMK

Component	Value	Values Present	Short Description
MANDT	400	X	Client
BELNR			Sales and Distribution Document Number
KNUMV			Number of the document condition
VKORG			Sales Organization
VTWEG			Distribution Channel
SPART			Division
BZIRK			Sales district
VKGRP			Sales Group
BRSCH			Industry key
VKBUR			Sales Office
KONDA			Price group (customer)
KDGRP			Customer group
PLTYP			Price list type
ALAND	US	X	Departure country (country from which the goods are sent)

In this example we clicked on the KOMK table line. Notice the header of this screen shows the journey name and the table you selected. You see the actual data that was available by field name on this screen. This will aid in analysis by a Business Analyst as to what data is also available for further field mapping without the need to use complex program debugging tools. To return to the journey level simply use the “X” in the top right corner of this popup level to move one step back. You can then opt to select another table to view or X out of the journey.

Exiting out of the journey will bring up the pop-up for the next journey that was selected in the config if it is used in the transaction. Continue with the analysis tool until you go through all of the available journey screens at which point the tool will end until another call is made to Determination.

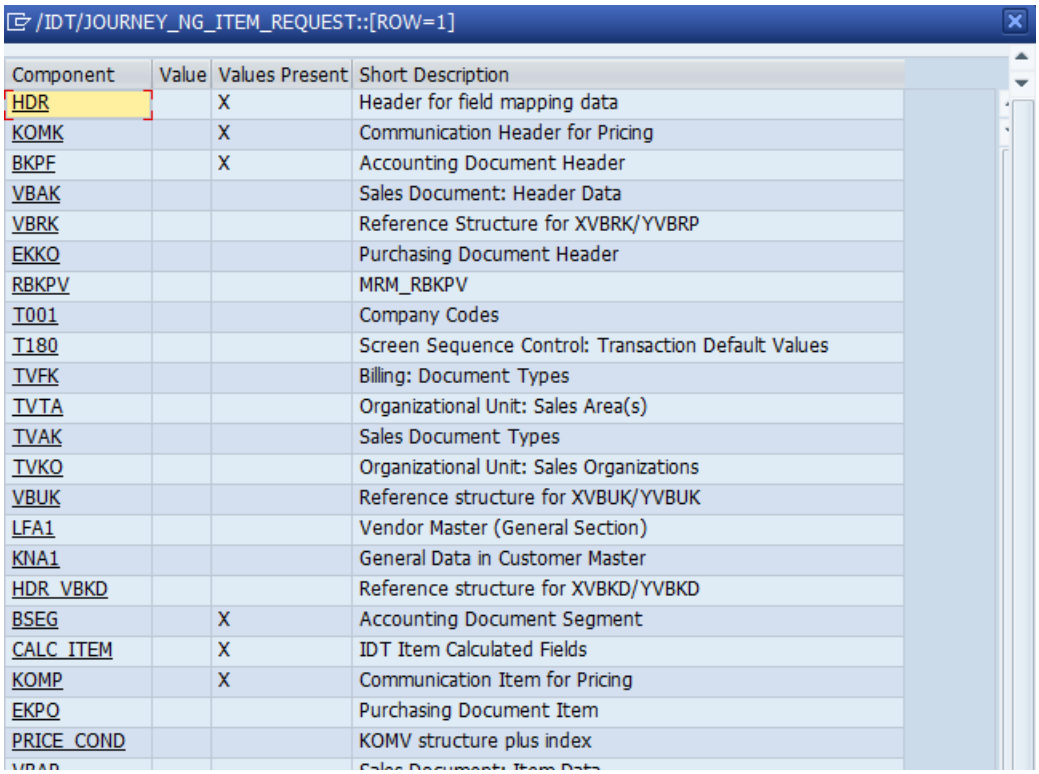


Display View "FI Control Process: Standard": Overview

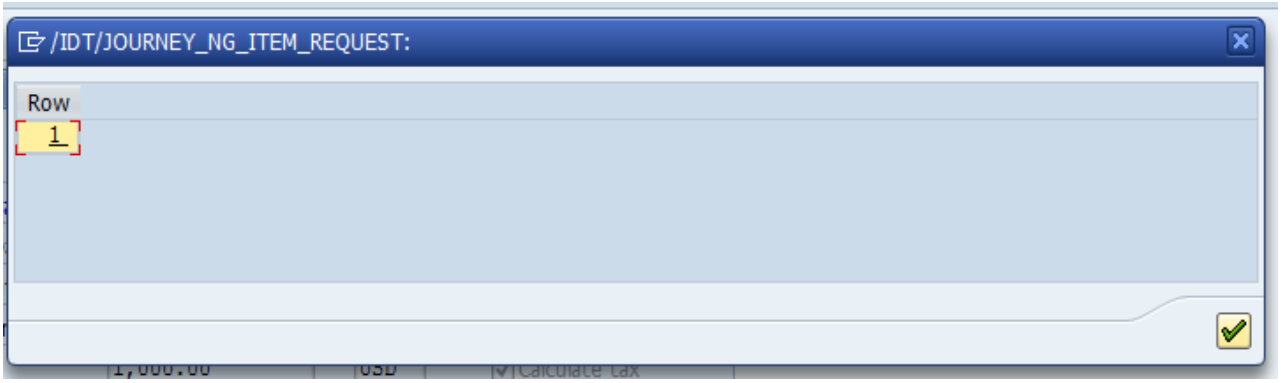
FI Control Process: Standard

S...	Pr...	Type	Sort ...	A...	C...	CoCd	Tax Calculation Process Class	Description
☑	GN	SA	10	☑	*	*	:/IDT/FI_PROC_DEFAULT_GL	Default G/L
☑	GN	SA	20	☑	*	*	:/IDT/FI_PROC_CUSTOMER_CREDIT_M	Customer Credit Memo
☑	GN	SA	30	☑	*	*	:/IDT/FI_PROC_CUSTOMER_INVOICE	Customer Invoice
☑	GN	SA	40	☑	*	*	:/IDT/FI_PROC_VENDOR_CREDIT_MEM	Vendor Credit Memo
☑	GN	SA	50	☑	*	*	:/IDT/FI_PROC_VENDOR_INVOICE	Vendor Invoice
☑	GN	SA	60	☑	*	*	:/IDT/FI_PROCESS_DEFERRED_TAX	Deferred Tax
☑	GN	ZP	10	☑	*	*	:/IDT/FI_PROCESS_FB05	New Cash Discount

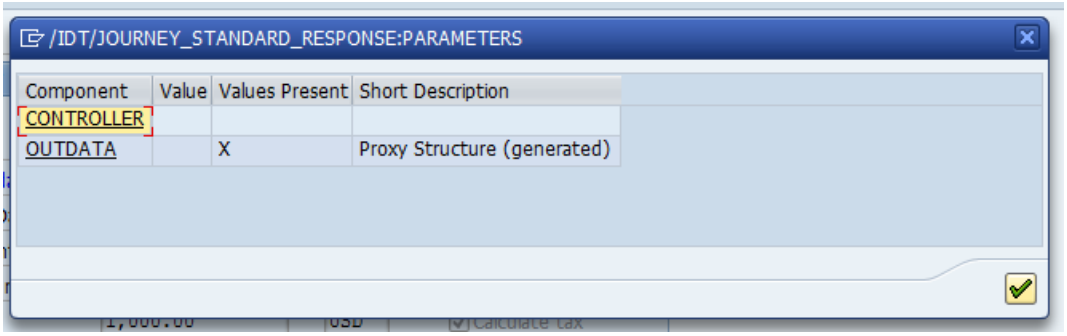
More screen examples below:



The item request journey screen will look a little different as it first shows the list of the row numbers at the line-item level. Click to then select a line and go further down into the table detail for the line selected.

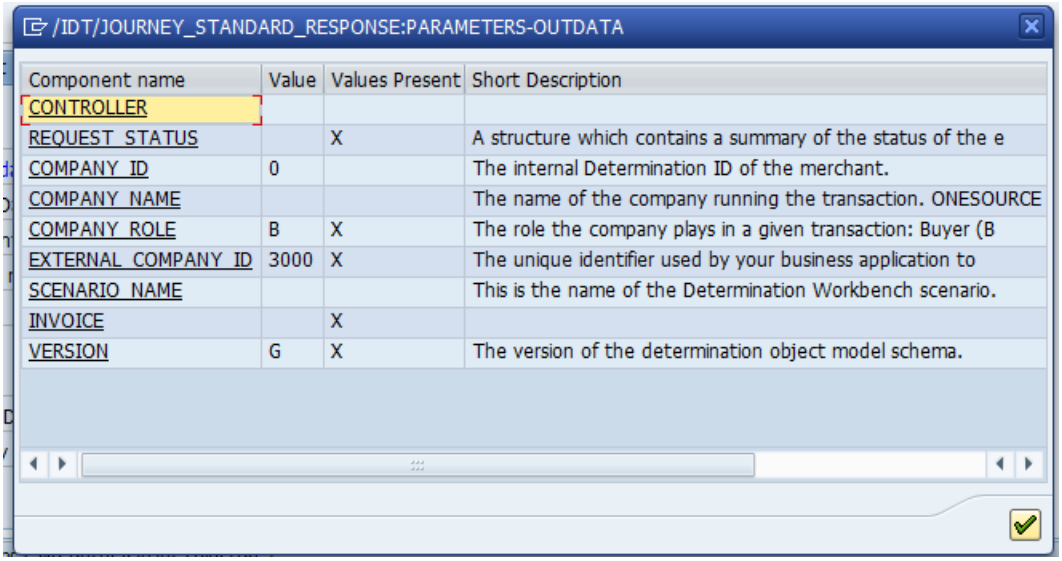


The standard response journey will look a little different as well. Selecting parameters will get you to the next screen below:



Then to the outdata...

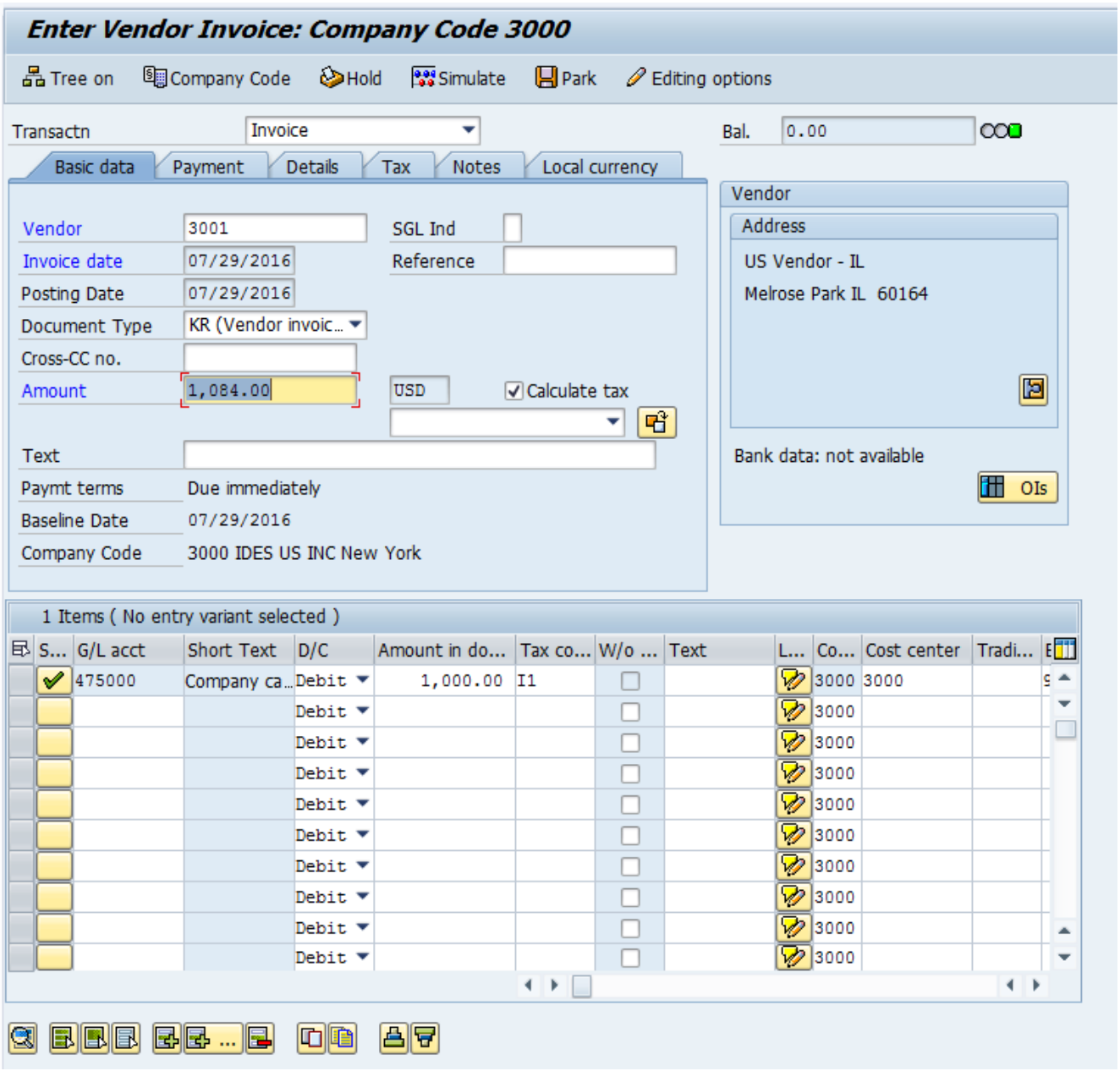
... and invoice level data...



/IDT/JOURNEY_STANDARD_RESPONSE:PARAMETERS-OUTDATA-INVOICE:[ROW=1]

Component name	Value	Values Present	Short Description
CONTROLLER			
REQUEST_STATUS		X	A structure which contains a summary
BASIS_PERCENT			A percentage of the gross amount of
CALCULATION_DIRECTION			There are three calculation directions,
CALLING_SYSTEM_NUMBER	400	X	A pass-through element that contains
COMPANY_ID	15	X	The internal Determination ID of the n
COMPANY_NAME	SAP US INC 3000	X	The name of the company running the
COMPANY_ROLE	S	X	The role the company plays in a given
CURRENCY_CODE	USD	X	The currency associated with a transac
CURRENCY_NAME	United States Dollar	X	The currency used for this invoice. Loc
MIN_ACCOUNTABLE_UNIT	0.01	X	Allows empty tags for decimal amounts
ROUNDING_PRECISION	2	X	Number of decimal places to round the
ROUNDING_RULE	1	X	The rounding rule from the TB_CURRE
CUSTOMER_NAME			The customer's name. Used in both V
CUSTOMER_NUMBER			The customer's number as passed in b
END_USER_NAME			Identifier of the user who made a char
EXTERNAL_COMPANY_ID	3000	X	The unique identifier used by your bus
FISCAL_DATE	2016-07-29	X	Stores a transaction by a fiscal date in
FUNCTIONAL_CURRENCY_CODE			Reserved for future use.
HOST_SYSTEM	DO3	X	The name of the ERP instance sending
INDATA			Contains batch level request fields as v
INVOICE_DATE	2016-07-29	X	The date of the invoice.Note: Determ
INVOICE_NUMBER			This number is passed through to the
IS_AUDIT_UPDATE			Indicates whether or not to permit au
IS_BUSINESS_SUPPLY			Indicates whether a supply is being us
IS_CREDIT			Indicates whether the current transac
IS_REPORTED			Controls whether the invoice being pro
IS_REVERSED			Indicates whether the current transac
MESSAGE			
NATURE_OF_TRANSACTION_CODE			Used in Intrastat reports in all countrie
ORIGINAL_INVOICE_NUMBER			This number enables a credit invoice to
STATISTICAL_PROCEDURE			A statistical indicator used for large ship
TOTAL_TAX_AMOUNT	84.00	X	Proxy Data Element (generated)
TRANSACTION_DATE	2016-07-29	X	The system date at the time the trans
UNIQUE_INVOICE_NUMBER	3000||S|2016	X	The unique identifier for this invoice, a
VENDOR_NAME	US Vendor - IL	X	The vendor's name. If this is a B2B tra
VENDOR_NUMBER	0000003001	X	The vendor's number. If this is a B2B t
VENDOR_TAX			The amount of tax charged by the Ve
USER_ELEMENT		X	
LINE		X	
ORIGINAL_DOCUMENT_ID			The original document id as shown on
ORIGINAL_DOCUMENT_ITEM			The original document item as shown
ORIGINAL_DOCUMENT_TYPE			The original document type as shown
ORIGINAL_INVOICE_DATE			The date of the original invoice. Used
ORIGINAL_MOVEMENT_DATE			The date of the original movement. Us
CUSTOMER_GROUP_NAME			Name of the Customer Group to use f
CUSTOMER_GROUP_OWNER			Name of the Company owning the Cus
TAX_SUMMARY			A summary of tax amounts and warnin

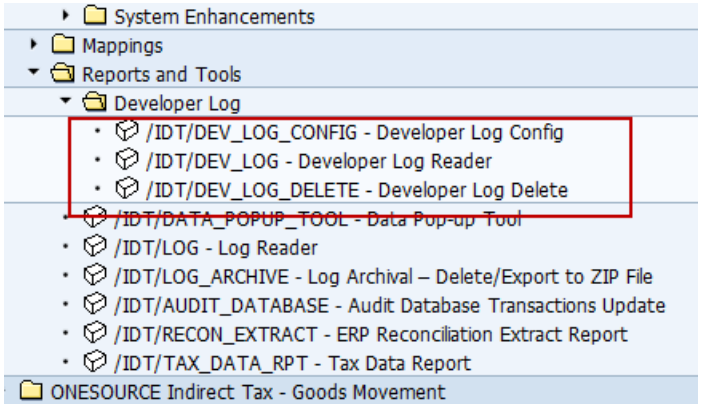
... and finally, back to the transaction.



Note: Reminder that depending on the transaction you are testing you may need to start another session to deactivate the popup tool before you can exit your tested transaction scenario. If multiple calls to determination are invoked, the tool can possibly loop depending on the document.

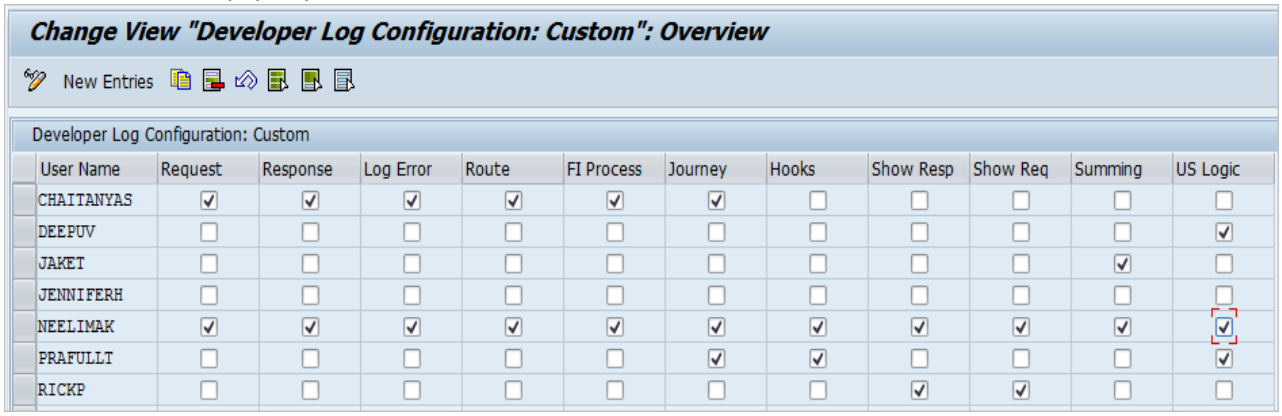
DEVELOPER'S LOG

A log tool has been designed for developer use that gives additional flexibility and traceability features while debugging of a transaction or process. It is designed to aid the developer in understanding the program logic and cut short the time needed for debug. This log is separate from the normal transaction log and can be configured to turn on various tracking tools and features depending on what is needed for analysis. It is intended that this tool is only accessed by the developer and is in the Reports and Tools menu at the bottom of the ONESOURCE menu. Three separate transactions allow the developer to turn on the log for their use with company code combination and tailor the output for various features, select logs using a log viewer, and to delete logs no longer needed. This is intended to be a temporary analysis tool and logs created are not stored or archived for long term use.+



Developer Log Configuration

Transaction Code: /N/IDT/D_DL_CONFIG



User Name	Request	Response	Log Error	Route	FI Process	Journey	Hooks	Show Resp	Show Req	Summing	US Logic
CHAITANYAS	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐
DEEPUV	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑
JAKET	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
JENNIFERH	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
NEELIMAK	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
PRAFULLT	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☑
RICKP	☐	☐	☐	☐	☐	☐	☐	☑	☑	☐	☐

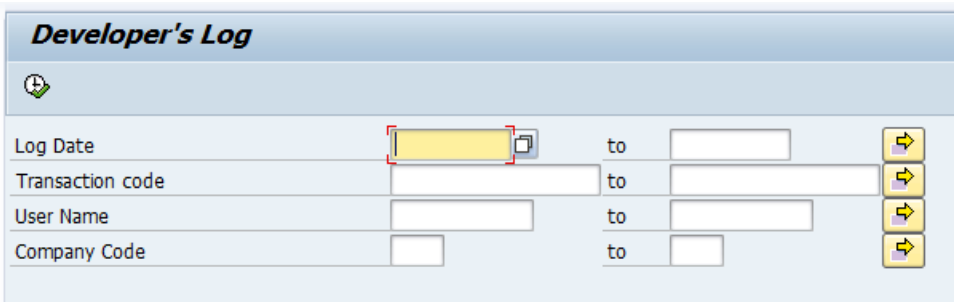
In this first iteration of the developer log the developer can enter their logon and activate the various features as noted below:

1. Request The request data only will display as a separate log file
2. Response The response data only will display as a separate log file
3. Log Error Include error messages
4. Route Display the routes that were used on the call
5. FI Process Display the FI processes that were used on the call
6. Journey Display the journeys that were used on the call
7. Hooks Display the hooks that are used on the call
8. Show Resp to record changes to the response fields per journey (where the data came from)
9. Show Req to record changes to the request fields per journey (where the data came from
10. Summing List of the items that were summarized within the line-item details.
11. US Logic Display the US Specific journey and US specific logic data when used

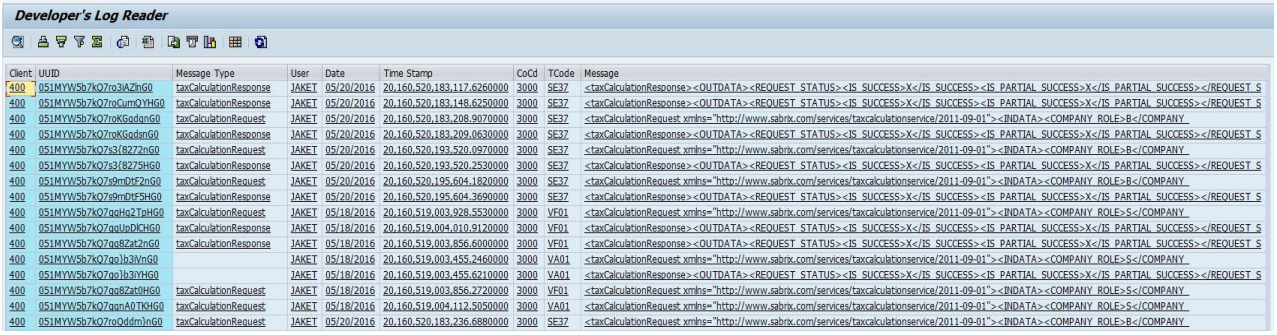
Note: The intention of the developer log is that you activate it when you need to debug a transaction. This is to be used only by the developers to investigate issues or make needed enhancements to the program. Other features may likely be added to this report in the future.

Developer Log Viewer

Transaction Code: /N/IDT/DEV_LOG



Display of the Developer log selection screen



View of the log reader after selection

Double click on a log line to display the XML log that was recorded.

Deletion of Developer Logs

Transaction Code: /N/IDT/DEV_LOG_DELETE

This transaction will take you to a deletion screen in order to delete the developer logs that are no longer needed. Developers should clear their logs on a regular basis after the debugging and analysis is complete to avoid unnecessary storage usage.

PERFORMANCE IMPROVEMENT WITH DYNAMICALLY GENERATED CODE

When a change to the field mapper is generated the system will create a transport request. This transport request will then be able to be released and moved to the next system for update of the field mapper table. Normally this process of field mapper table update would be done in the development system and then transported to a testing or QA system and tested before being moved to production. The update of the dynamic code that is used with the mapper is then re-generated at the time the new mapping is used for the first time after the change is made or transported to the target system. Users will experience a very small delay the first time a line is utilized while the code is regenerated and saved. This is similar to time frame when a program is re-compiled after a change. This process ensures that the code being used on the transaction always correctly represents the corresponding field mapper line. Once the code is dynamically generated after the field mapper line change, it is stored and reused until the next change of the field mapper line-item for that specific client and system.

Clean Up Program for Dynamically-Generated Field Mapping Code

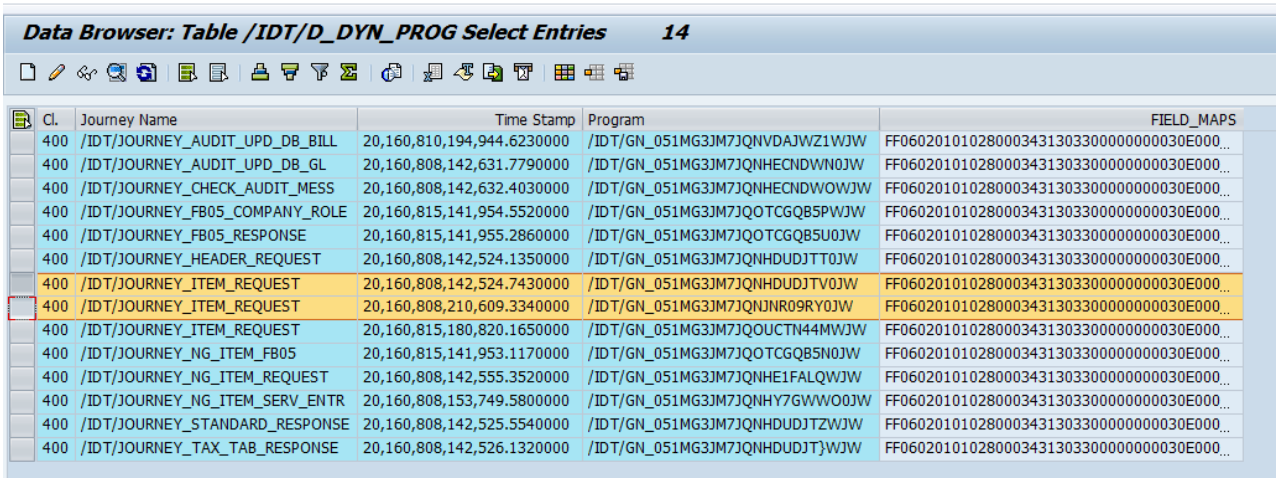
With the dynamically generated code for the performance of the field mapping function, new code is created behind the scenes for each line of the field mapper. As changes are made to the field mapper the older code that was created from the prior change is still in the table but is no longer used as the new code updated is used

instead. This can cause a build-up of old and outdated dynamically generated code that will need to be removed from the system to keep the system clean. We have provided a program that you will need to run in background on a regular basis to clean up and remove the older code that is no longer being used.

Program name: **/IDT/DELETE_DYN_PROGRAMS**

We recommend that you have this program set up to run on a background job on a regular basis to keep your system clean of older dynamic code that has been replaced by new mappings. Depending on how frequently you are changing field mapping this will likely be once a month or if a lot of changes are being performed possibly once a week. The program does not have any selection criteria and would be simple to run and schedule by your IT staff.

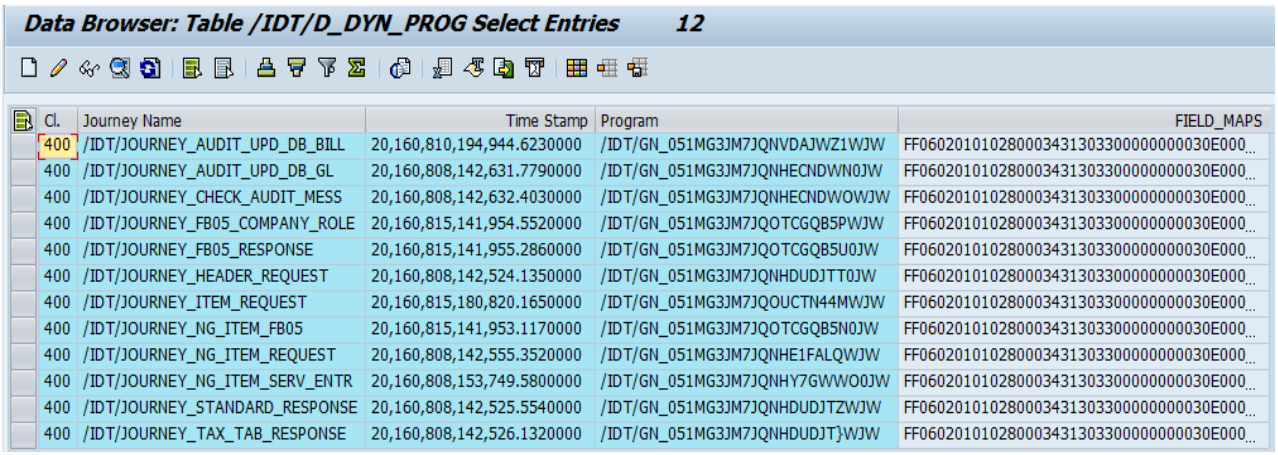
In the example below, the two highlighted lines should be the only ones deleted after the program is run. You can see by this table that three entries exist for this journey and that the two highlighted are the older ones that need to be removed:



Data Browser: Table /IDT/D_DYN_PROG Select Entries 14

Cl.	Journey Name	Time Stamp	Program	FIELD_MAPS
400	/IDT/JOURNEY_AUDIT_UPD_DB_BILL	20,160,810,194,944.6230000	/IDT/GN_051MG3JM7JQNVDAJWZ1WJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_AUDIT_UPD_DB_GL	20,160,808,142,631.7790000	/IDT/GN_051MG3JM7JQNHECNDWN0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_CHECK_AUDIT_MESS	20,160,808,142,632.4030000	/IDT/GN_051MG3JM7JQNHECNDWOWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_FB05_COMPANY_ROLE	20,160,815,141,954.5520000	/IDT/GN_051MG3JM7JQOTCGQB5PWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_FB05_RESPONSE	20,160,815,141,955.2860000	/IDT/GN_051MG3JM7JQOTCGQB5U0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_HEADER_REQUEST	20,160,808,142,524.1350000	/IDT/GN_051MG3JM7JQNHDUDJTT0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_ITEM_REQUEST	20,160,808,142,524.7430000	/IDT/GN_051MG3JM7JQNHDUDJTV0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_ITEM_REQUEST	20,160,808,210,609.3340000	/IDT/GN_051MG3JM7JQNJNR09RY0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_ITEM_REQUEST	20,160,815,180,820.1650000	/IDT/GN_051MG3JM7JQOUCTN44MWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_FB05	20,160,815,141,953.1170000	/IDT/GN_051MG3JM7JQOTCGQB5N0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_REQUEST	20,160,808,142,555.3520000	/IDT/GN_051MG3JM7JQNHE1FALQWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_SERV_ENTR	20,160,808,153,749.5800000	/IDT/GN_051MG3JM7JQNHY7GWWO0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_STANDARD_RESPONSE	20,160,808,142,525.5540000	/IDT/GN_051MG3JM7JQNHDUDJTZWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_TAX_TAB_RESPONSE	20,160,808,142,526.1320000	/IDT/GN_051MG3JM7JQNHDUDJT}WJW	FF06020101028000343130330000000030E000...

After running the program, you can see it is deleting the correct entries and that the most current line for Journey_Item_Request is now the only one on the list:



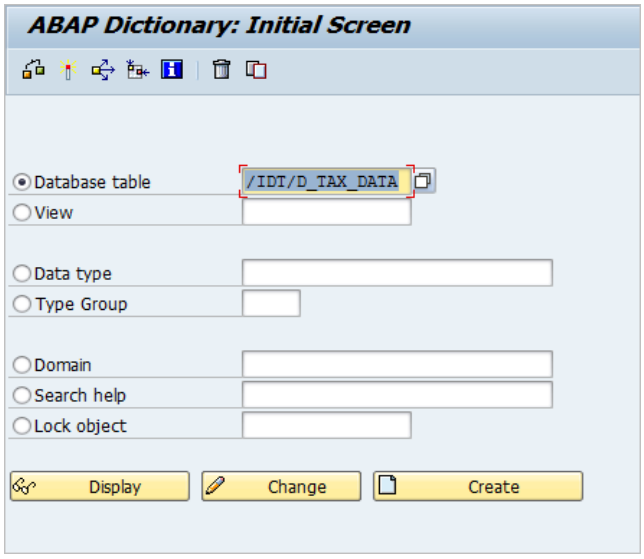
Data Browser: Table /IDT/D_DYN_PROG Select Entries 12

Cl.	Journey Name	Time Stamp	Program	FIELD_MAPS
400	/IDT/JOURNEY_AUDIT_UPD_DB_BILL	20,160,810,194,944.6230000	/IDT/GN_051MG3JM7JQNVDAJWZ1WJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_AUDIT_UPD_DB_GL	20,160,808,142,631.7790000	/IDT/GN_051MG3JM7JQNHECNDWN0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_CHECK_AUDIT_MESS	20,160,808,142,632.4030000	/IDT/GN_051MG3JM7JQNHECNDWOWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_FB05_COMPANY_ROLE	20,160,815,141,954.5520000	/IDT/GN_051MG3JM7JQOTCGQB5PWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_FB05_RESPONSE	20,160,815,141,955.2860000	/IDT/GN_051MG3JM7JQOTCGQB5U0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_HEADER_REQUEST	20,160,808,142,524.1350000	/IDT/GN_051MG3JM7JQNHDUDJTT0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_ITEM_REQUEST	20,160,815,180,820.1650000	/IDT/GN_051MG3JM7JQOUCTN44MWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_FB05	20,160,815,141,953.1170000	/IDT/GN_051MG3JM7JQOTCGQB5N0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_REQUEST	20,160,808,142,555.3520000	/IDT/GN_051MG3JM7JQNHE1FALQWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_NG_ITEM_SERV_ENTR	20,160,808,153,749.5800000	/IDT/GN_051MG3JM7JQNHY7GWWO0JW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_STANDARD_RESPONSE	20,160,808,142,525.5540000	/IDT/GN_051MG3JM7JQNHDUDJTZWJW	FF06020101028000343130330000000030E000...
400	/IDT/JOURNEY_TAX_TAB_RESPONSE	20,160,808,142,526.1320000	/IDT/GN_051MG3JM7JQNHDUDJT}WJW	FF06020101028000343130330000000030E000...

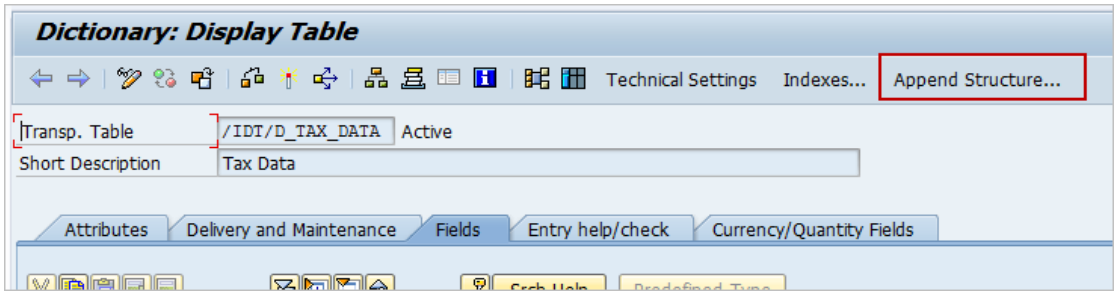
ADDING CUSTOM FIELDS TO /IDT/D_TAX_DATA TABLE

A system user may have need to add other fields to the /IDT/D_TAX_DATA table for reporting or reconciliation processes. They can append the table with their own custom field and then use the field mapping table feature to be able to populate the new field per their needs. A simple procedure of appending the table is outlined below:

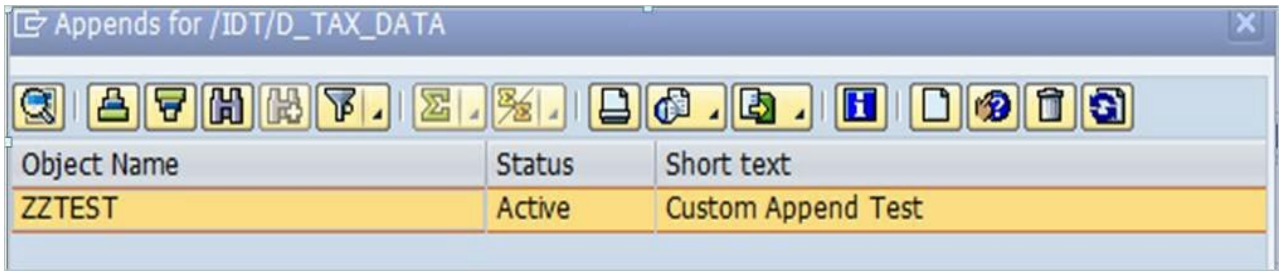
Transaction: SE11



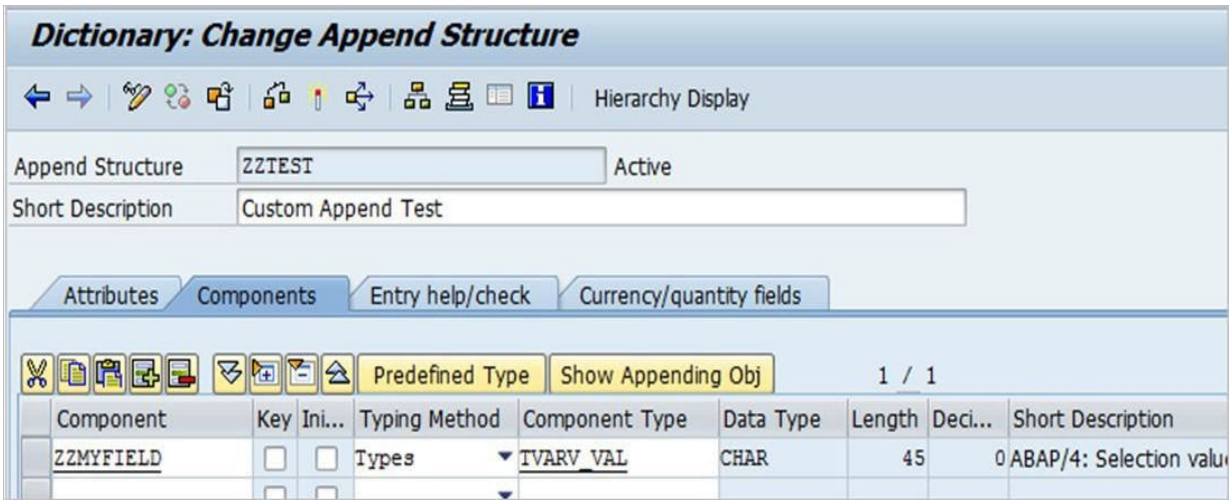
Select “Display”. Once in the table, use the “Append Structure (F5)” menu option.



Create a new append in the customer name space or using ZZ* naming convention. Include the change in a transport:



Add custom fields as desired:



Save and activate the changes.

The newly added fields can now be used in the Global Next Flexible Field Mapper to which to map data.

SIMPLE EXPRESSIONS

Simple expressions are just like a line of code, but they are added to the field mapping line as a qualifier, only if the expression is fulfilled the mapping will be considered.

The syntax of a simple expression is as follows:

SOURCE FIELD | OPERAND | CHECK VALUE (FIELD)

Some requirements are:

- SAP table-field names must always be wrapped with ampersands (&)
- Values must always be wrapped with single quotes (‘)
- Operations can be stringed together by AND or OR commands
- Supported operands are:
- EQ, = Equal To
- NE, <>, >< Not Equal To
- LT, < Less Than
- LE, <= Less Than or Equal To
- GT, > Greater Than
- GE, >= Greater than or Equal To
- CO Contains Only
- CN Contains Not only
- CA Contains Any
- NA Contains Not Any

- CS Contains String
- NS Contains No String
- CP Matches Pattern
- NP Does Not Match Pattern
- IS_INITIAL Field is initial value

IS_INITIAL is a special command that can be used with EQ or NE to further delineate if a field has been populated or if it has been set to the initial value of blank for this transaction. See example below:

Journey Name	Sort O...	Ac...	Source Ba...	Source Field	Target Base	Target Field	O...	Description	Simple Expression
/IDT/JOURNEY_HEADER_REQUEST	100011	☑	SAP_HEAD...	KOMK-PRSDT	DET_INVOICE	INVOICE_DATE	☐	MM INVOICE_DATE	
/IDT/JOURNEY_HEADER_REQUEST	100012	☑	SAP_HEAD...	KOMK-AUDAT	DET_INVOICE	INVOICE_DATE	☐	SD INVOICE_DATE	&KOMK-AUDAT& NE IS_INITIAL
/IDT/JOURNEY_HEADER_REQUEST	100013	☑	SAP_HEAD...	BKPF-BLDAT	DET_INVOICE	INVOICE_DATE	☐	FI INVOICE_DATE	&BKPF-BLDAT& NE IS_INITIAL

In this example the IS_INITIAL is used in the simple expression after “NE” to populate the invoice date if it has been set in this transaction and is not equal to the initial value of blank.

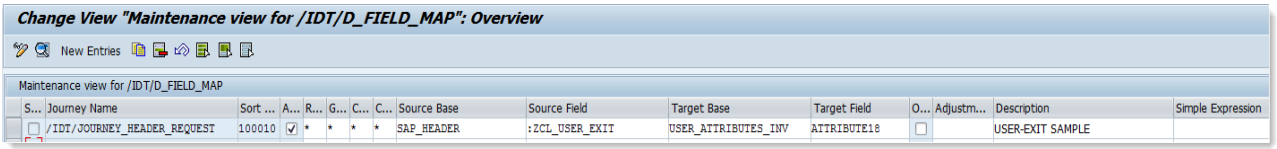
Other example for operands listed above:

EXPRESSION	EXPLANATION
&KOMK-VKBUR& = '1030'	Only maps the field if the Sales Office value is 1030.
&VBAK-ERDAT& NE &SY-DATUM&	Only uses the mapping if the system date isn’t the same as the documents create date.
(&KOMK-WAERK& = 'USD' and &VBAK-ERDAT& = &SY-DATUM&) or &SY-TCODE& CP 'VA’	Maps the field if the Document Currency is USD and the Document Create date is the system date OR of the transaction code starts with the letters “VA”
'NL_RC_TR_ZE_ZC' CS &TAX_TYPE&	Only uses the mapping of the Tax Type contains any of these values: NL, RC, TR, ZE, or ZC.

USER-EXIT IN FIELD MAPPER

The Field Mapping (/N/IDT/FIELD_MAPPINGS) allows for dynamic mapping of SAP source fields to Determination and vice versa. In most cases the options of doing mappings by journey, routes, route groups, country groups, or company code are enough to meet most customer requirements, especially in combination with the Simple Expression feature that allows for some ABAP syntax to be added in the mapper directly.

However, in some complex situations, or when the Thomson Reuters provided source bases aren’t covering a table required for a custom mapping, a customer might implement a user-exit based mapping.



Such mappings are still added to the Field Mapping table but are linked to a unique ABAP class which will contain the code in support of that mapping. The result would look something like this sample:

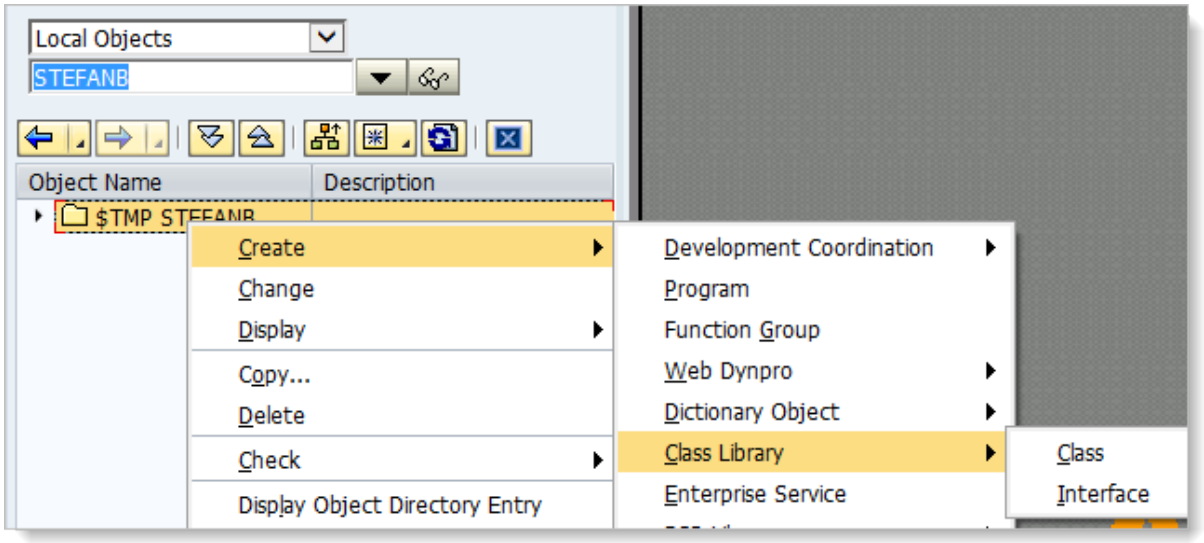
As you can see in above sample the Source Field has been replaced with an ABAP Class. This class must be created first, and then assigned in the field mapping to the appropriate line. The class name must be pre-pended with a colon (:).

A user-exit based mapping can be combined with allowed Adjustments as well as with Simple Expressions and any other table field like country group, etc.

Creating a user-exit based mapping requires a few simple steps of coding. The actual code logic within the user-exit might be complex however depending on the use case.

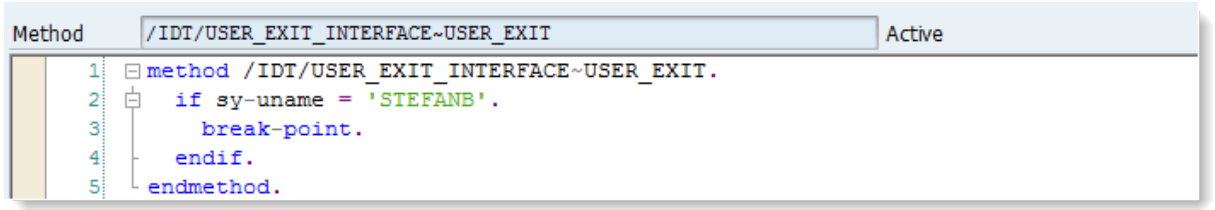
To create your own ABAP Class based user-exit, go to Transaction Code SE80.

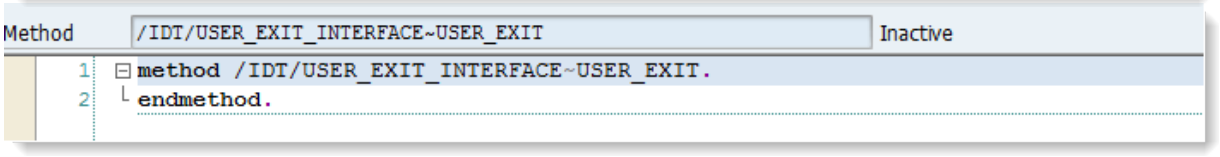
1. Create your class.

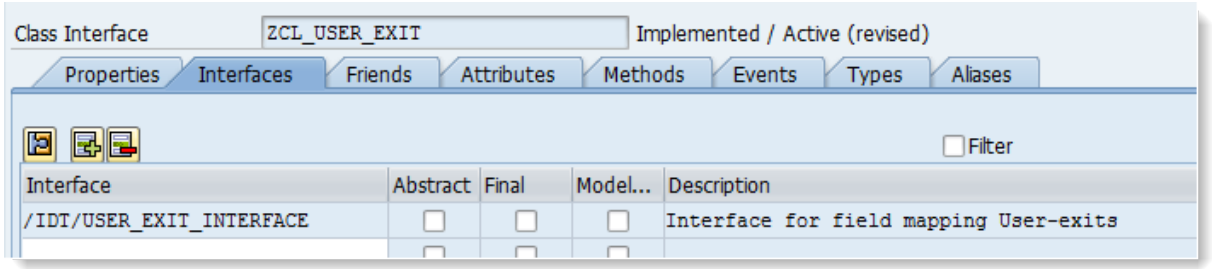


2. Save your class. You will be taken to the new class. Navigate to the Interfaces tab. Here you will need to include the predefined interface /IDT/USER_EXIT_INTERFACE for the Field Mapping which inherits all the necessary data for the mapper.
3. Now switch over to the Methods tab, double click on the predefined method and you will be taken into the code.

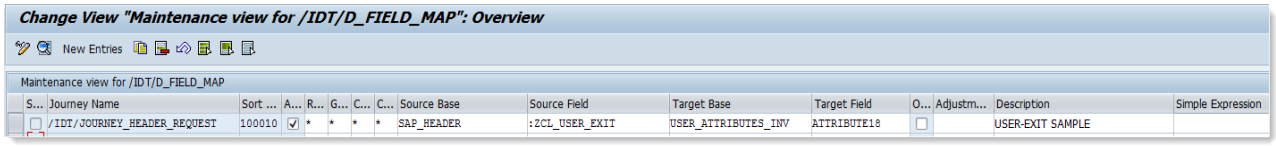
4. Add your custom code in this method (this is just a sample).







5. Once coding is done Activate your class.
6. Now you can go and assign your newly created class to the relevant line in the Field Mapper. Test your mapping by activating the line and executing a test transaction.



Item Data Now Available in Header and Item User Exits

As an example, this feature could be used in the following two scenarios:

1. If you need to determine at the header level a field that is stored at the line-item level to pass that to the request.

2. If you want to look at a line that is a consequence of another line like a freight charge or surcharge and you need to refer to the parent line to get some information needed to properly calculate tax on the related child line.

To accomplish either of these scenarios via ABAP programming, you can use this field in the header called “Items”. This Items field is a pointer that allows you to get the item data needed for the above two purposes. It increases the function of the user exit and simple expressions to use for some fringe cases where this may be needed. You may never need this, but it is available if needed.

Code example of how to use this:

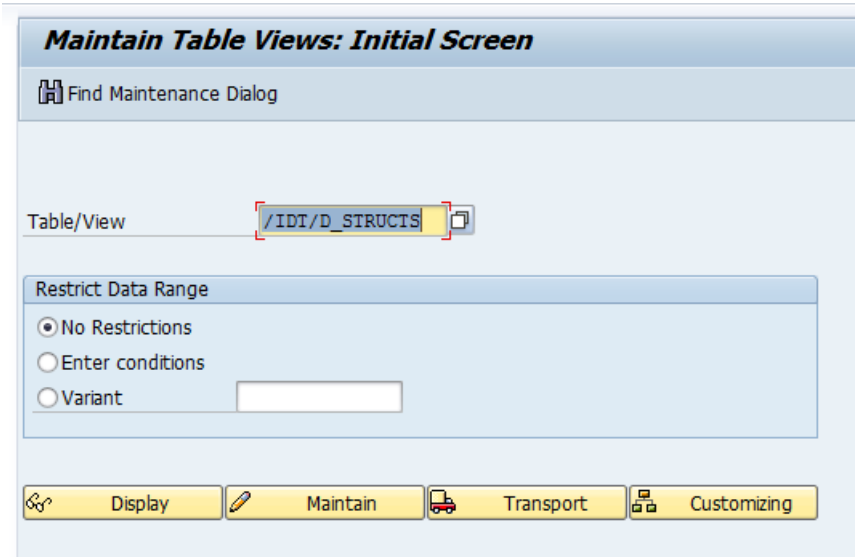
```
DATA     :     M_REF_UTIL_ITEMS     TYPE     REF     TO
       /IDT/REFERENCE_UTILITY, M_REF_UTIL_ITEM TYPE
       REF  TO  /IDT/REFERENCE_UTILITY,  MV_VARIABLE
                              TYPE STRING.

 M_REF_UTIL_ITEMS   =   I_REF_UTIL_SOURCE_DATA->PATH('HDR-
 >ITEMS'). DO M_REF_UTIL_ITEMS->COUNT( ) TIMES.

        M_REF_UTIL_ITEM  =  M_REF_UTIL_ITEMS->ROW(  SY-
        INDEX ). "GET FIELD
```

FIELD MAPPER JOURNEY DROP-DOWN LIST

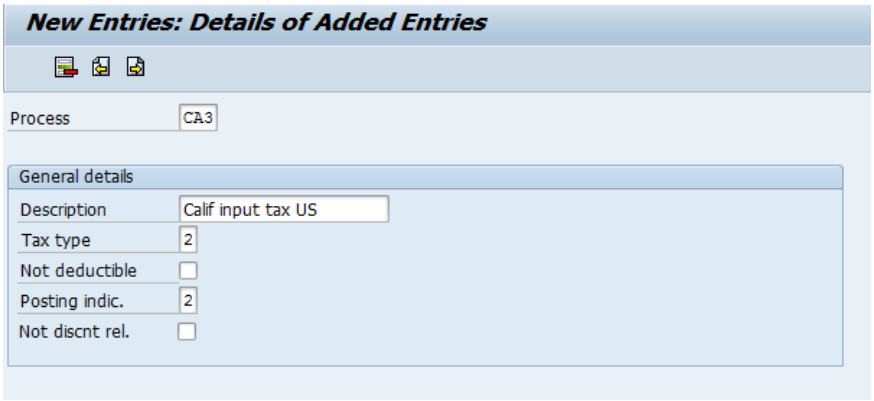
The drop-down list of available journeys is a feature of the Field Mapping Table. This drop-down list comes from an internal table within the Integration called /IDT/D_STRUCTS. If a user creates a new custom journey that will be used for field mapping purposes, the user will want to make sure that the new custom journey is available in the drop-down list so it can be selected for a field mapping. To do this the user will need to make an additional entry to the /IDT/D_STRUCTS table.

Transaction: SM30

Go to SM30 and enter table name as /IDT/D_STRUCTS. Click on Maintain.



Click on New Entries and add the Journey Names in the first column.



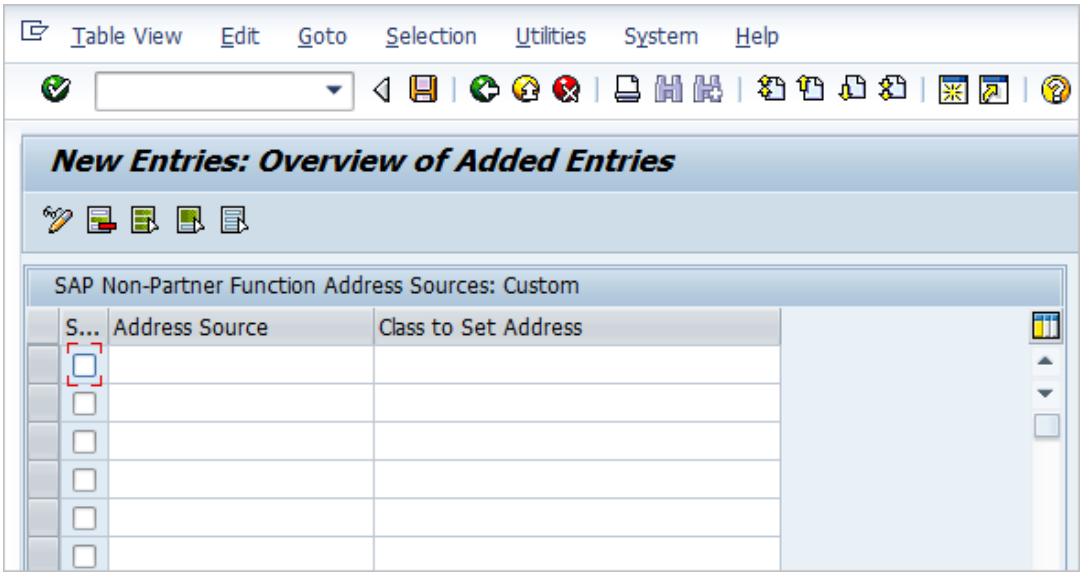
Click on save and attach to a Customizing Transport Request.

CUSTOM ADDRESS SOURCE

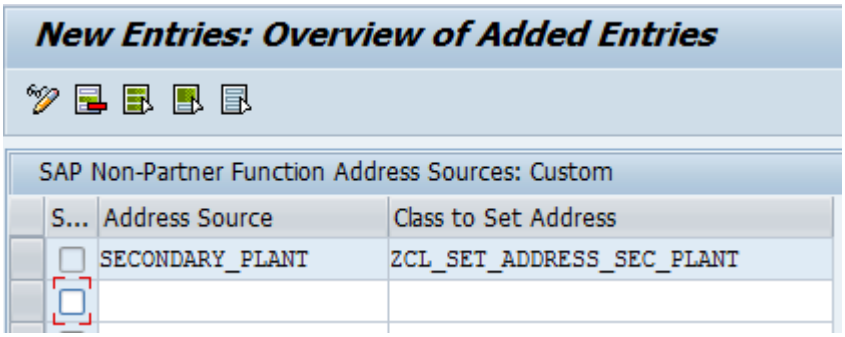
Thomson Reuters provides an extensive list of address sources which can be used in the Address Mapping for tax calculations. In some cases, the provided sources might not meet the unique business requirements you might have. You will be able to create your own address source, add code to a user-exit to gather the address data from your source, and then add it to the mapping for use.

Note: All code samples provided below are for reference purposes only to give an experienced ABAP developer some directions on how to code your own solution. They are not intended to work when copied as is.

1. Transaction Code /N/IDT/ADDRESS_SOURCES to create a new Address Source. Name your address source depending on your business needs.



Create the class which will have methods to have logic to populate secondary plant address. Step 3 explains more about class creation procedure and example.

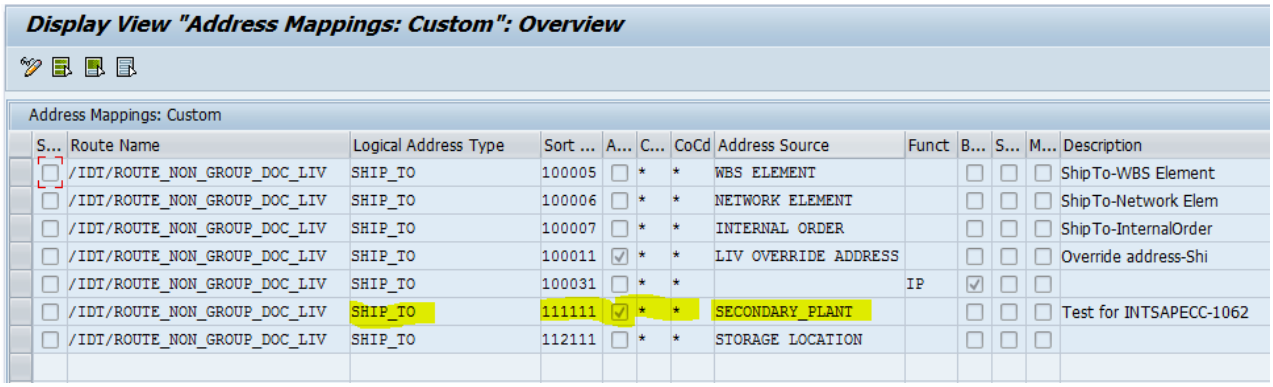


New Entries: Overview of Added Entries

SAP Non-Partner Function Address Sources: Custom

S...	Address Source	Class to Set Address
☐	SECONDARY_PLANT	ZCL_SET_ADDRESS_SEC_PLANT
☐		

2. Configure transaction /N/IDT/ADDRESS_MAPPING to map the new address source to an address type.
3. Class ZCL_SET_ADDRESS_SEC_PLANT can be created by copying any existing /IDT/SET_ADDRESS* class.



Display View "Address Mappings: Custom": Overview

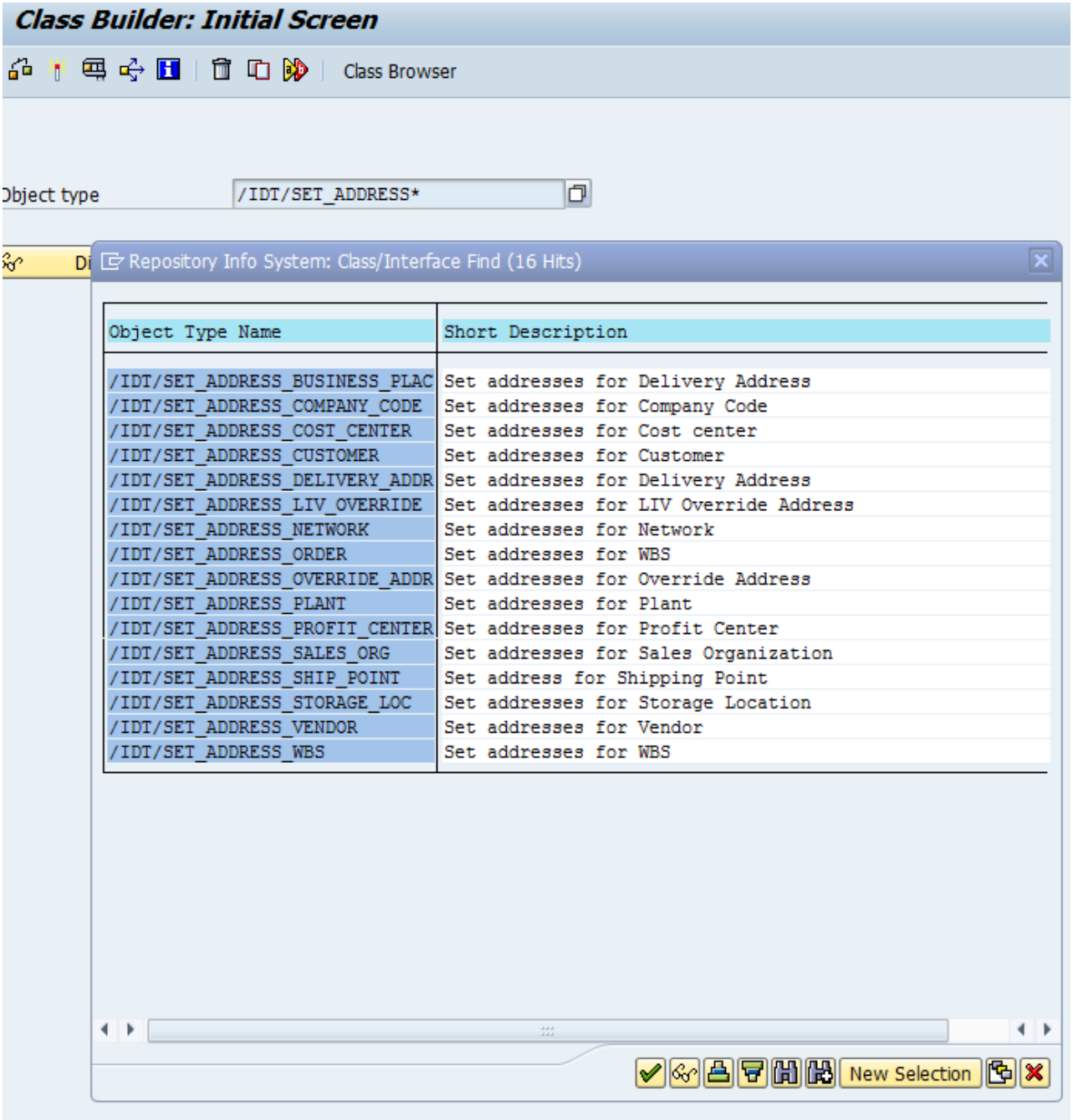
Address Mappings: Custom

S...	Route Name	Logical Address Type	Sort ...	A...	C...	CoCd	Address Source	Funct	B...	S...	M...	Description
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	100005	☐	*	*	WBS ELEMENT		☐	☐	☐	ShipTo-WBS Element
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	100006	☐	*	*	NETWORK ELEMENT		☐	☐	☐	ShipTo-Network Elem
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	100007	☐	*	*	INTERNAL ORDER		☐	☐	☐	ShipTo-InternalOrder
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	100011	☑	*	*	LIV OVERRIDE ADDRESS		☐	☐	☐	Override address-Shi
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	100031	☐	*	*		IP	☑	☐	☐	
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	111111	☑	*	*	SECONDARY_PLANT		☐	☐	☐	Test for INTSAPECC-1062
☐	/IDT/ROUTE_NON_GROUP_DOC_LIV	SHIP_TO	112111	☐	*	*	STORAGE LOCATION		☐	☐	☐	

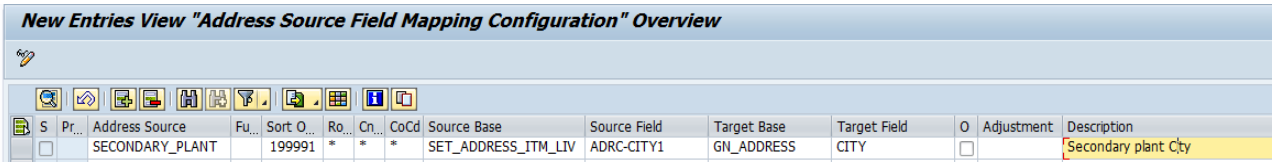
Or, the class can be created by using existing interface - /IDT/IF_BUILD_ADDRESS.

As an example – we have copied class /IDT/SET_ADDRESS_PLANT to create ZCL_SET_ADDRESS_ SEC_PLANT.

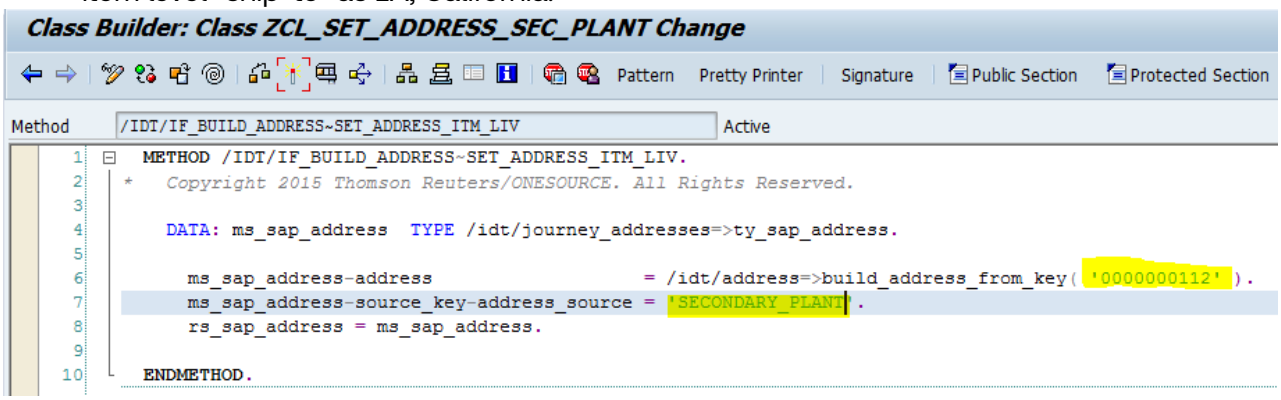
Make sure to delete existing code from the methods of the newly created class if it is copied from existing one.



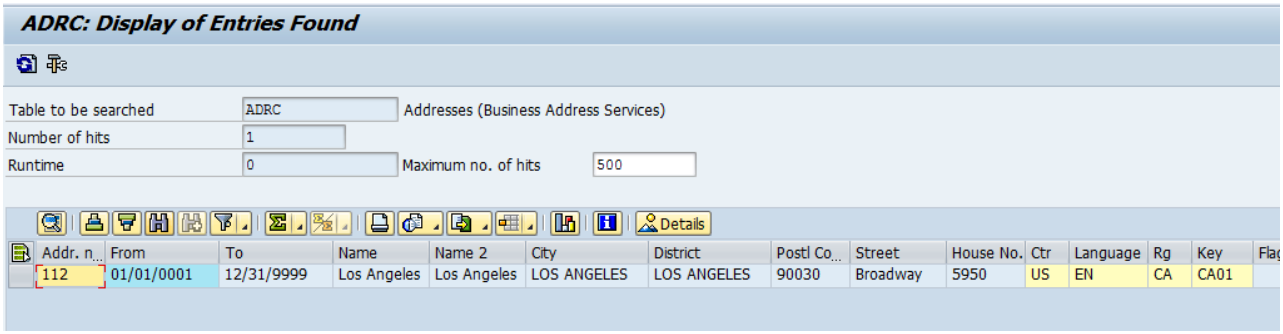
4. Maintain the address source mapping for secondary plant in the transaction code - /IDT/ADDR_FIELD_ MAP. (Optional)



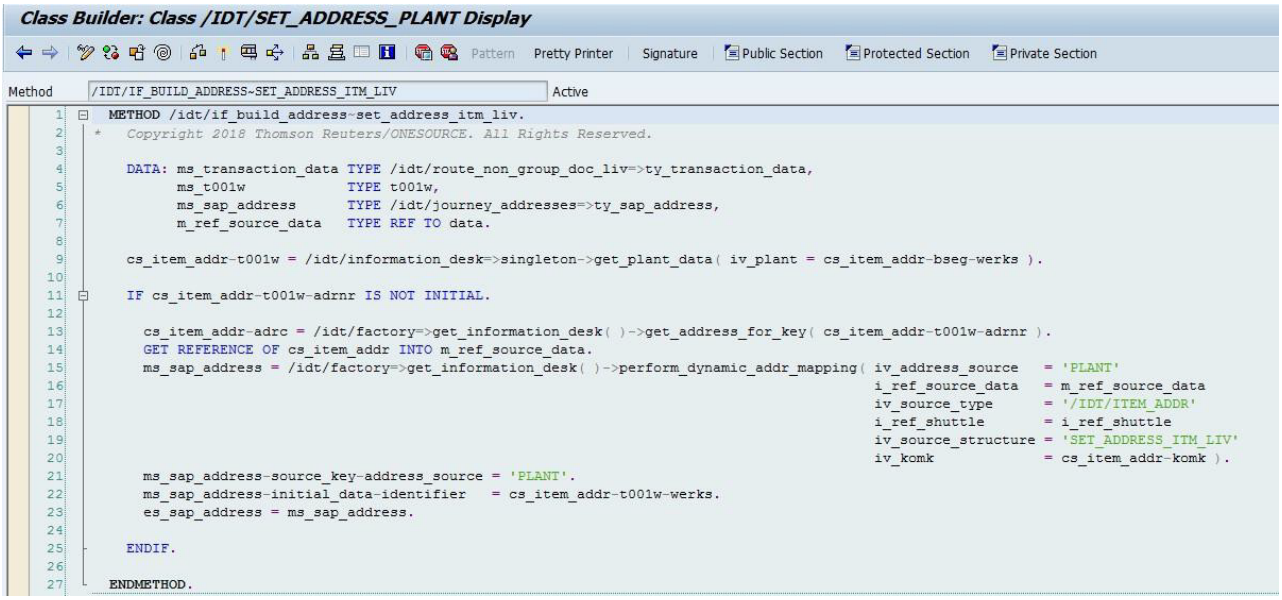
5. Now code your logic to populate address in respective method. For example, if you need an address to be populated at the line-item level of MIRO/LIV, then code the logic in that method SET_ADDRESS_ITM_LIV. Or if you need at sales order header level, then code your logic in the method SET_ADDRESS_HDR_ SALES.
   In our example we are having code for LIV at item level. We have hard-coded value 112 expecting item level “ship-to” as LA, California.



The above screen shot of SET_ADDRESS_ITM_LIV shows logic prior to 6.4.4.0. The code from 6.4.4.0 looks like



below example–

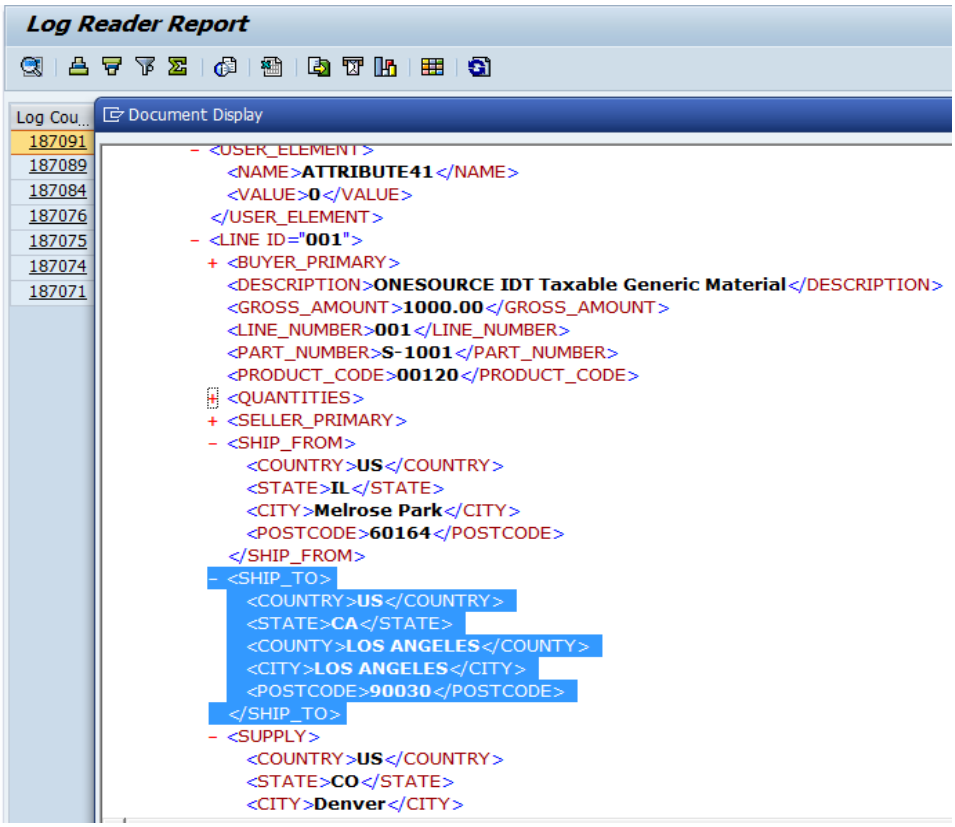



```
METHOD /idt/if_build_address~set_address_itm_liv.
*   Copyright 2018 Thomson Reuters/ONESOURCE. All Rights Reserved.

    DATA: ms_transaction_data TYPE /idt/route_non_group_doc_liv=>ty_transaction_data,
          ms_t001w            TYPE t001w,
          ms_sap_address      TYPE /idt/journey_addresses=>ty_sap_address,
          m_ref_source_data   TYPE REF TO data.

    cs_item_addr-t001w = /idt/information_desk=>singleton->get_plant_data( iv_plant = cs_item_addr-bseg-werks ).

    IF cs_item_addr-t001w-adrnr IS NOT INITIAL.

      cs_item_addr-adrc = /idt/factory=>get_information_desk( )->get_address_for_key( cs_item_addr-t001w-adrnr ).
      GET REFERENCE OF cs_item_addr INTO m_ref_source_data.
      ms_sap_address = /idt/factory=>get_information_desk( )->perform_dynamic_addr_mapping( iv_address_source   = 'PLANT'
                                                                                             i_ref_source_data   = m_ref_source_data
                                                                                             iv_source_type      = '/IDT/ITEM_ADDR'
                                                                                             i_ref_shuttle       = i_ref_shuttle
                                                                                             iv_source_structure = 'SET_ADDRESS_ITM_LIV'
                                                                                             iv_komk             = cs_item_addr-komk ).
      ms_sap_address-source_key-address_source = 'PLANT'.
      ms_sap_address-initial_data-identifier   = cs_item_addr-t001w-werks.
      es_sap_address = ms_sap_address.

    ENDIF.

ENDMETHOD.
```

From 6.4.4.0. we are having code to populate addresses by using dynamic programs which gets generated based on the address source field mapping done in the step 4 above.

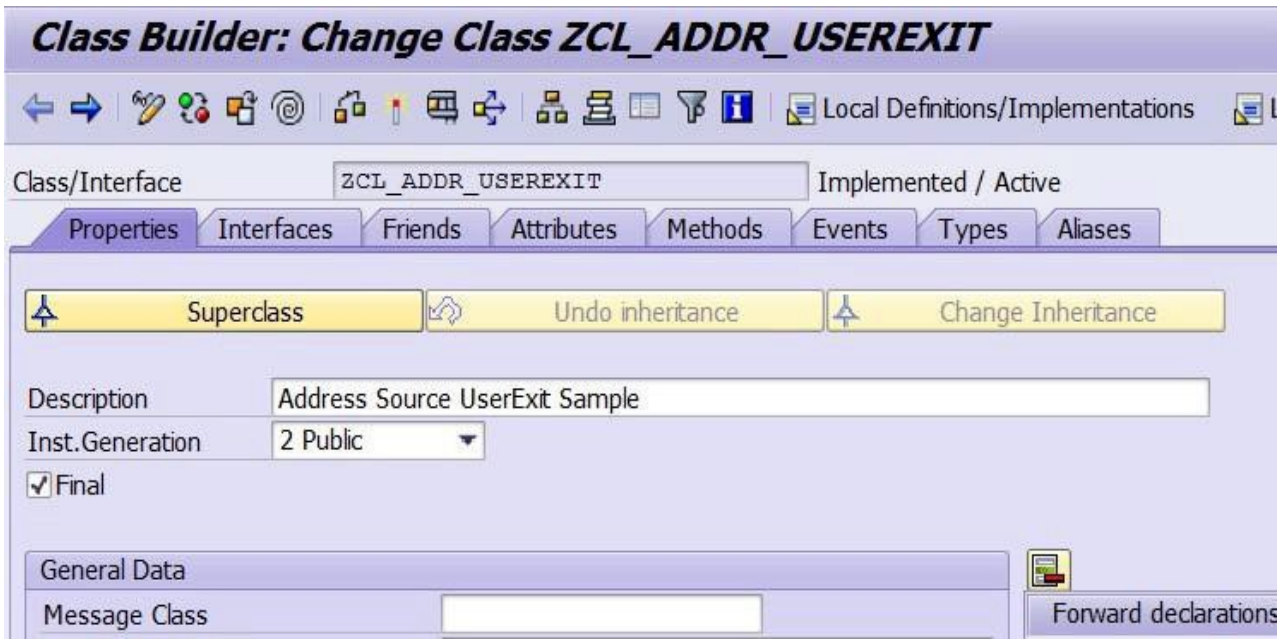
Customers can use either logic to populate ADRC/SADRVB tables in the exporting parameter CS_ITEM_ ADDR-ADRC or CS_ITEM_ADDR-SADRVB or they can write their own logic to populate CS_ITEM_ADDR- ADRC.

6. Activate the code and you are done. Create a MIRO document and check logs to confirm if it is populating the correct address for “ship to” at line-item level.

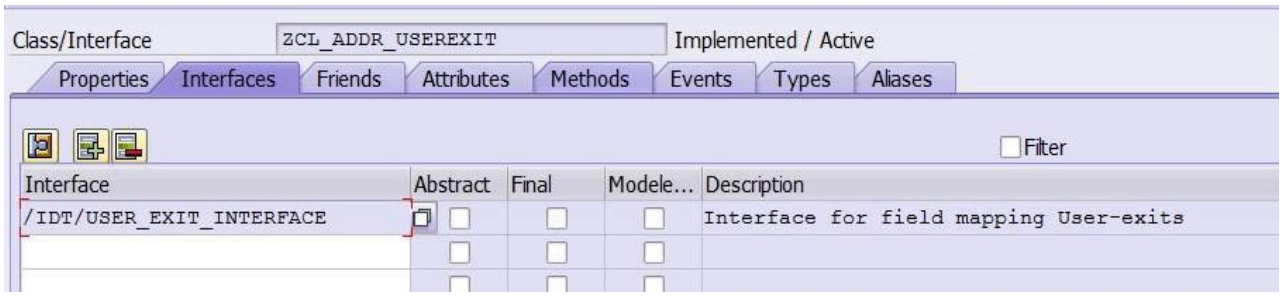


CREATING A USER EXIT MAPPING IN NEW ADDRESS SOURCE MAPPER

a. Create a new class using SE24. Sample class ZCL_ADDR_USEREXIT shown below:

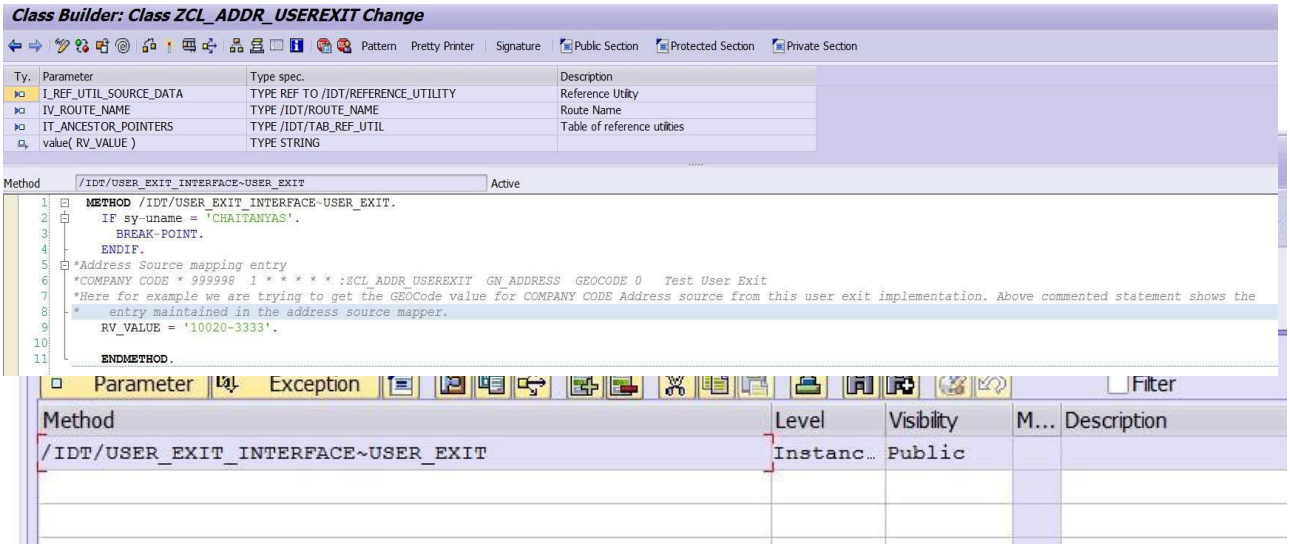


b. Under Interfaces tab – add predefined interface /IDT/USER_EXIT_INTERFACE

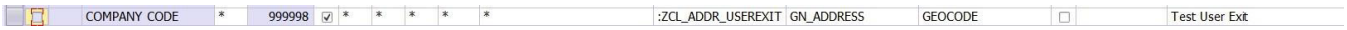


c. Double click on the method generated under Methods tab.

d. Add custom code as per your requirement to update the RV_VALUE Parameter value. Example shown below:

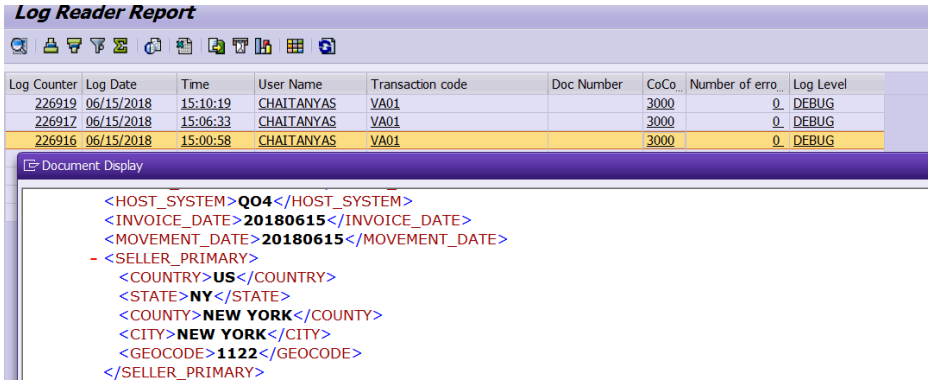


e. Maintain an entry in Address Source mapper as shown below. In our example, we are using ZCL_ADDR_ USEREXIT class to overwrite the GEOCODE Value for Company Code Address Source

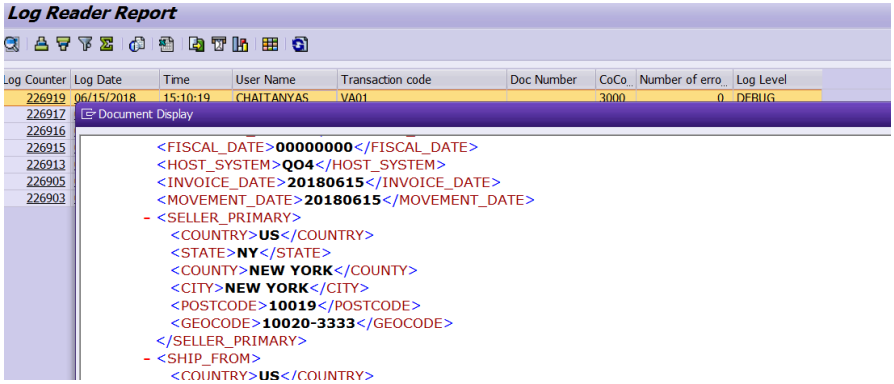


f. Sample sales order creation and results -
With the User Exit entry in Address source mapper, log shows updated Geocode value (10020-3333) in the address.

Without User Exit Entry



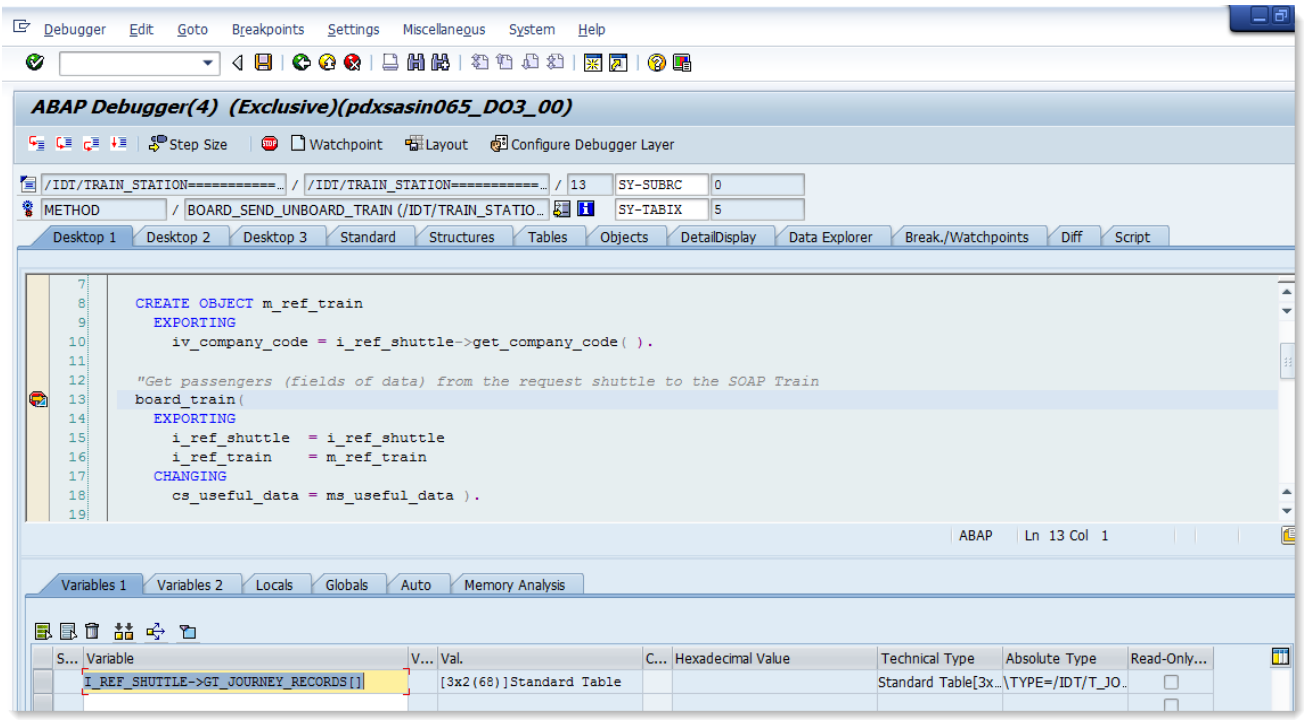
With User Exit Entry



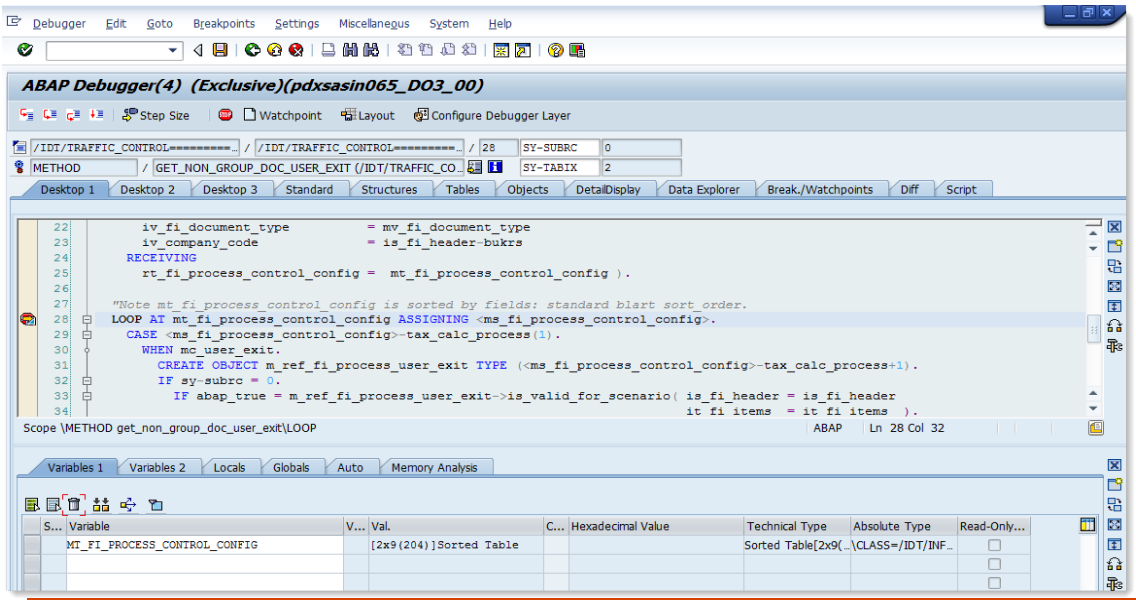
DEBUGGING

To know what Route/Journeys have been used, put a break-point at

/IDT/TRAIN_STATION=>BOARD_SEND_UNBOARD_TRAIN( ). From there you can see the actual list of Journeys for boarding and/or unboarding in methods BOARD_TRAIN( ) and UNBOARD_TRAIN( ).



To debug the process where an exit is chosen in an FI transaction, put a break-point in the /IDT/TRAFFIC_CONTROL=>GET_NON_GROUP_DOC_USER_EXIT( ).



OPTIONAL BADI OR FUNCTIONS THAT CAN BE INSTANTIATED

/IDT/BADI_ADJUST_TAX_SUMMATION

This BADI is used as part of the summarization of line-items in table /IDT/V_TAX_SUM_C to be able to add additional logic for summarization and override the configuration for specific documents based on any document characteristic. This allows the system user to control dynamically the various features for summarization based on need. Often this would be used on large documents that may hit the 1000-line limit in SAP. A user could with this BADI control when summarization is used based on the number of lines, the document type, etc.

For our example below, we show the code required within the BADI to turn off summarization on documents if there are less than 100 expense lines in the document:

METHOD /idt/adjust_tax_summation~adjust_tax_summation.

DATA : mv_line_count TYPE i,

mt_summarization_fields TYPE /idt/tab_idt_tax_sum_grp.

mt_summarization_fields = it_summarization_fields.

DESCRIBE TABLE it_fi_items LINES mv_line_count.

IF mv_line_count => 100.

"Let config be unchanged

ELSE.

CLEAR mt_summarization_fields.

ENDIF.

et_summarization_fields = mt_summarization_fields.

ENDMETHOD.

Special note: For LIV the number of lines will be one off of what is expected because LIV treats the

vendor line differently than FB60 does.

/IDT/BADI_ADJUST_PROXY

This BADI is used to add security to the proxy for HTTPS communication with Determination. The proxy will require a special username and password to be created in order for the additional security check to work as designed. See section above WS Security Considerations on the Proxy for a Hosted Environment. We have noted this BADI in this optional section, but it is currently part of the standard installation of the system. It is not optionally instantiated; however, it is not used if the WS security is not configured. A user may have other security protocol requirements based on their environment. Optional security programming may be done by the user's ABAP team within this BADI to address these needs.

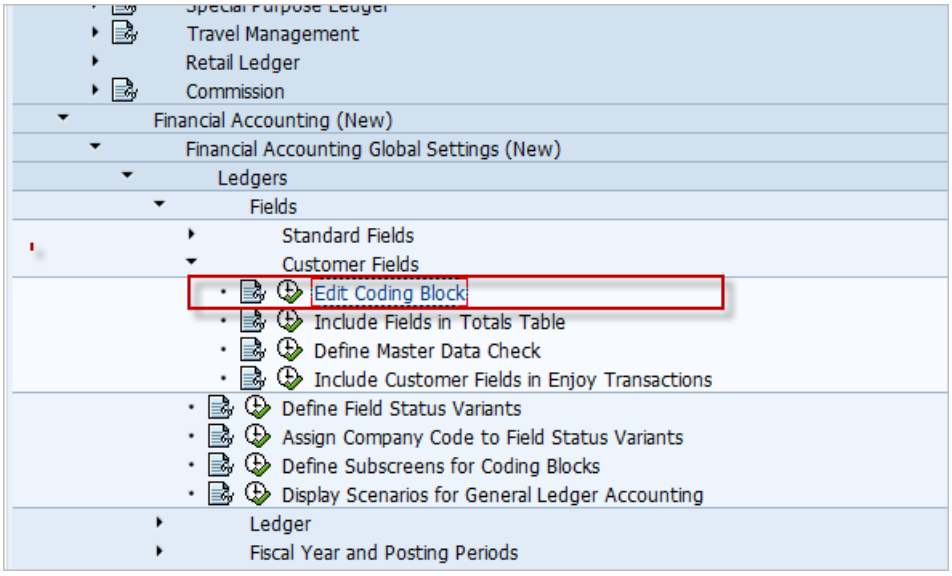
ADDING IDT ADDRESS FIELD TO INVOICE LINE-ITEM ENTRY SCREENS

Within the User Guide we discussed a feature “Changing Ship-To Address at Invoice” where a user could change the default ship to address at time of invoice on transactions FB60, FB65, FB70, FB75, FB01, and MIRO General Ledger tab. The following instructions show you how to add the IDT ADDRESS field to the line-item detail section of the invoice entry. This will save you a step on the FB** transaction rather than having to double click on the line-item to get to the more data screen. Instead the address field can be displayed as one of the columns in the line-item entry section of the transaction. For the LIV MIRO transaction this will be a requirement as the transaction does not use the more data tab for additional code block fields. The address field will have to be displayed in MIRO in the General Ledger tab for detail data entry.

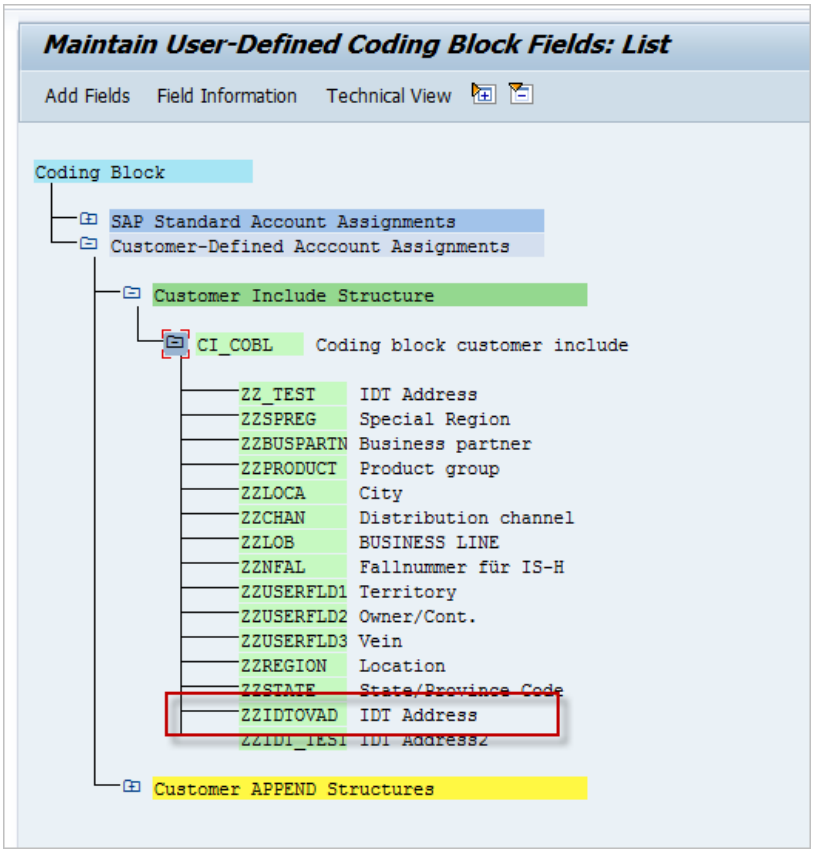
To add the address field to the MIRO PO Reference tab for use on invoice entry there is an entirely different process as this is part of MM rather than FI module and requires a different approach. The section below talks about this process and OSS notes used to add a tab to the line-item entry section of the MIRO transaction.

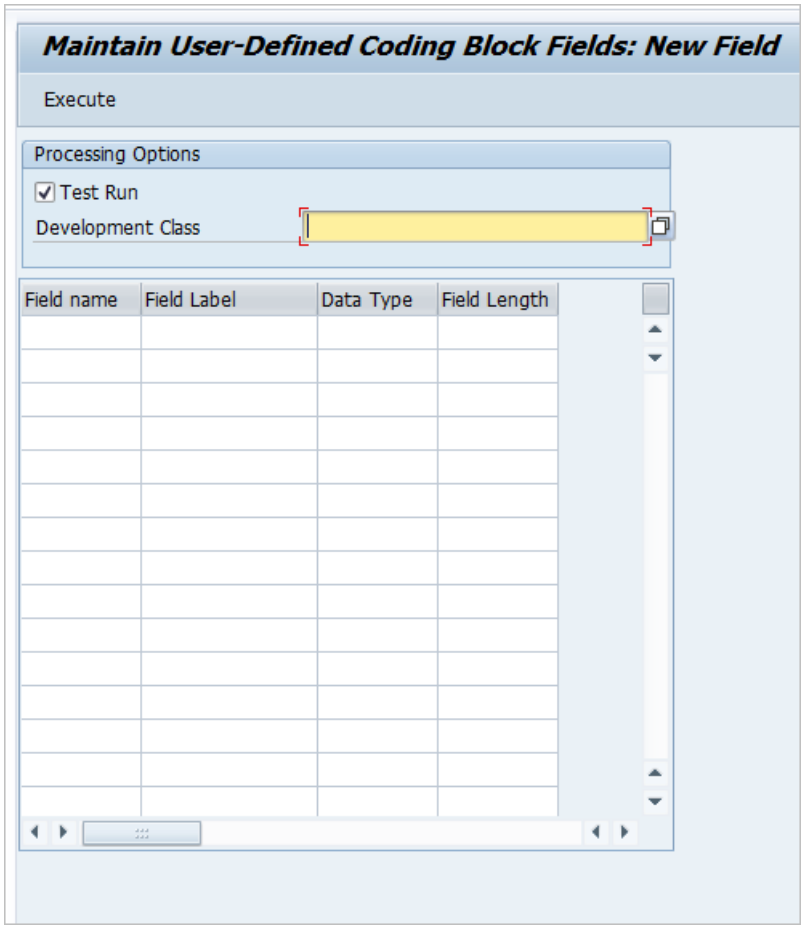
Note that this is not a simple process and will need to be done by an experience ABAP programmer that is familiar with adding custom fields to the code block and screen variants. There are a total of 5 customer fields that are provided by SAP within the coding block. If your company has already used these five then other decisions will need to be made as to your possible options including the use of modifications, substitution of an unused custom field, etc.

1. Add field to coding block: Follow steps in “Edit Coding Block” instructions. Use field name ZZIDTOVAD and data element ZZIDTOVAD. Use the expert mode per the instructions on Edit Coding Block.
   Transaction: SPRO navigate to > FINANCIAL ACCOUNTING (NEW) > LEDGERS > FIELDS > CUSTOMER FIELDS > EDIT CODING BLOCK



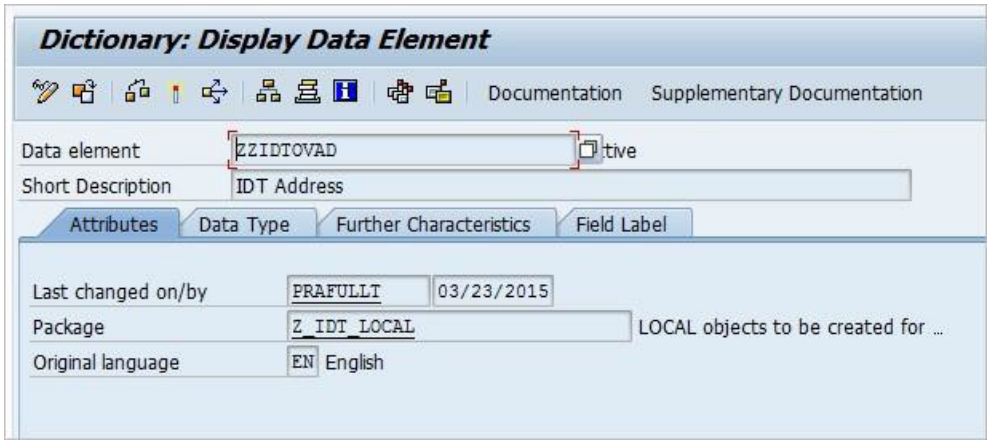
Expand menu to get to Customer Include Structure and then follow the expert mode instructions that are provided for this step. Step 2 and 3 of these instructions are already provided by our system transport and can be skipped.

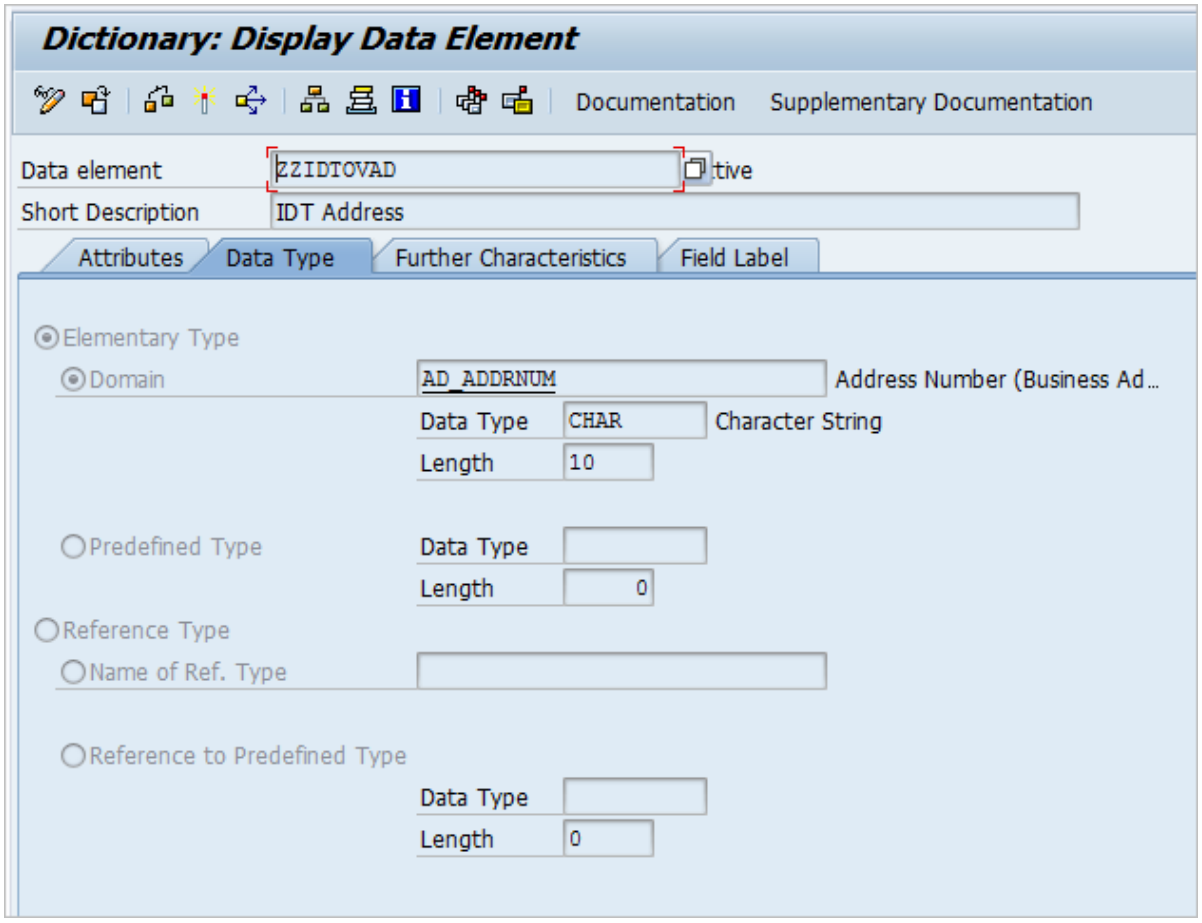


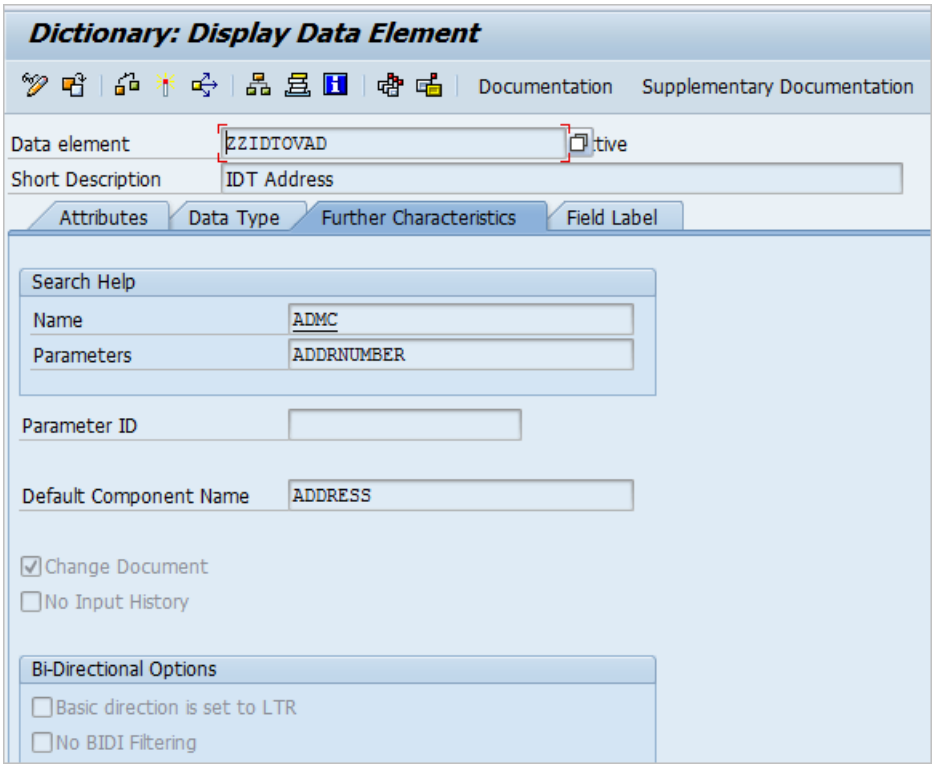


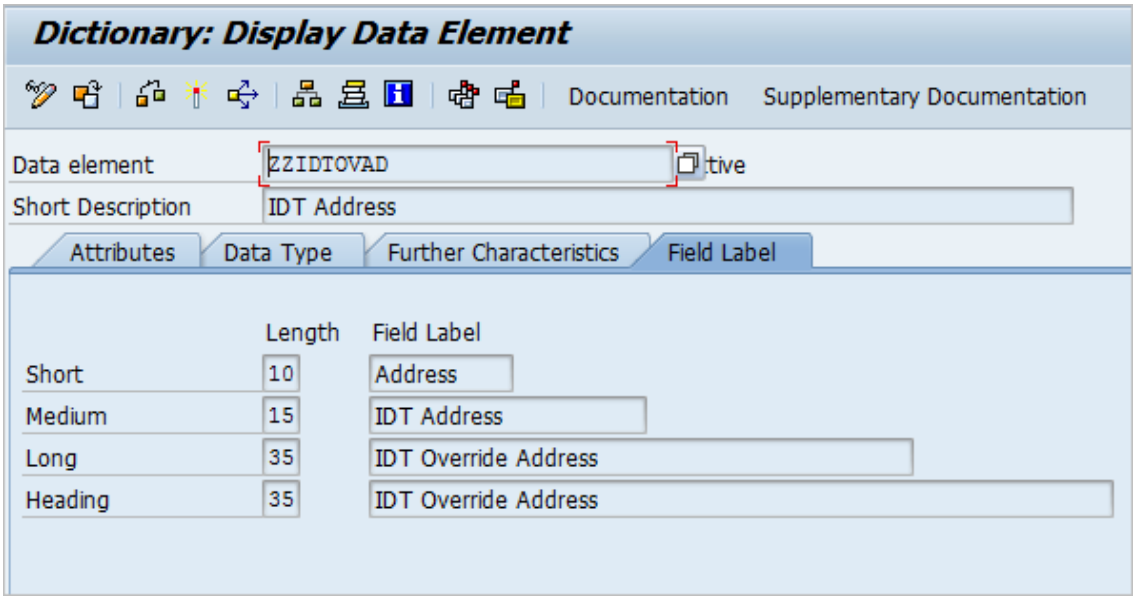
- Development class: (per your system)
- Field Name: ZZIDTOVAD
- Field Label: IDT Address
- Data Type: CHAR

- Field Length: 10



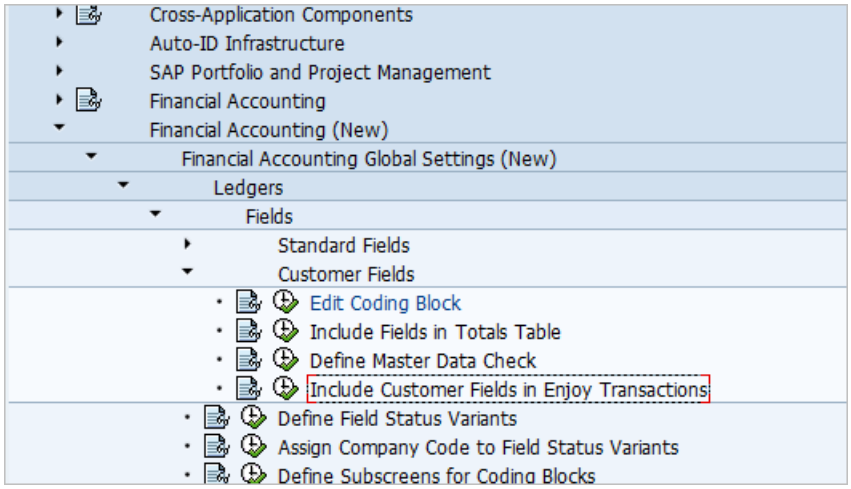




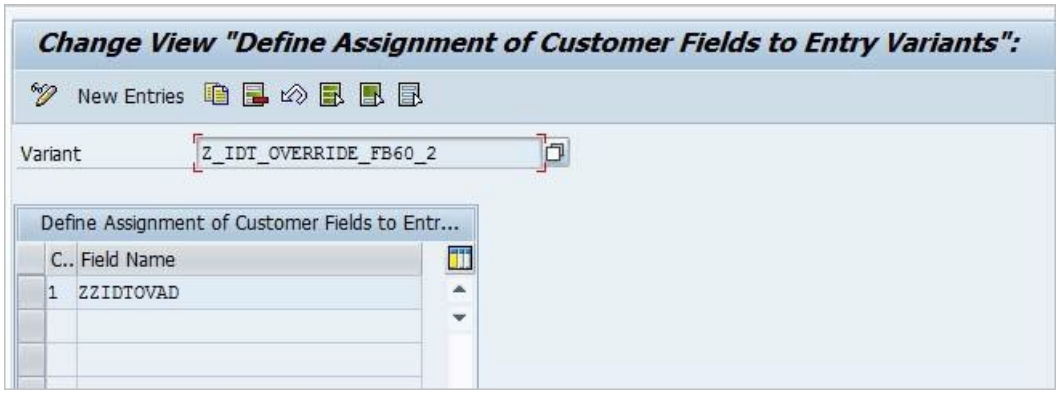


2. In “Include Customer Fields in Enjoy Transactions”, select your screen variant and assign field ZZIDTOVAD to a number 1 to 5. (This example links ZZIDTOVAD to ACGL_ITEM_GEN-GEN_CHAR2.) Your variant name will be different. Screen shots below are just an example. Note: HANA Coding Block (CI_COBL) - Update CDS Views

For HANA SAP systems, when Coding block (CI_COBL) is updated with custom fields (For Example (ZZIDTOVAD)) corresponding CDS_VIEW may need to be adjusted.
Transaction: SPRO navigate to **> FINANCIAL ACCOUNTING (NEW) > LEDGERS > FIELDS > CUSTOMER FIELDS >INCLUDE CUSTOMER FIELDS IN ENJOY TRANSACTIONS**

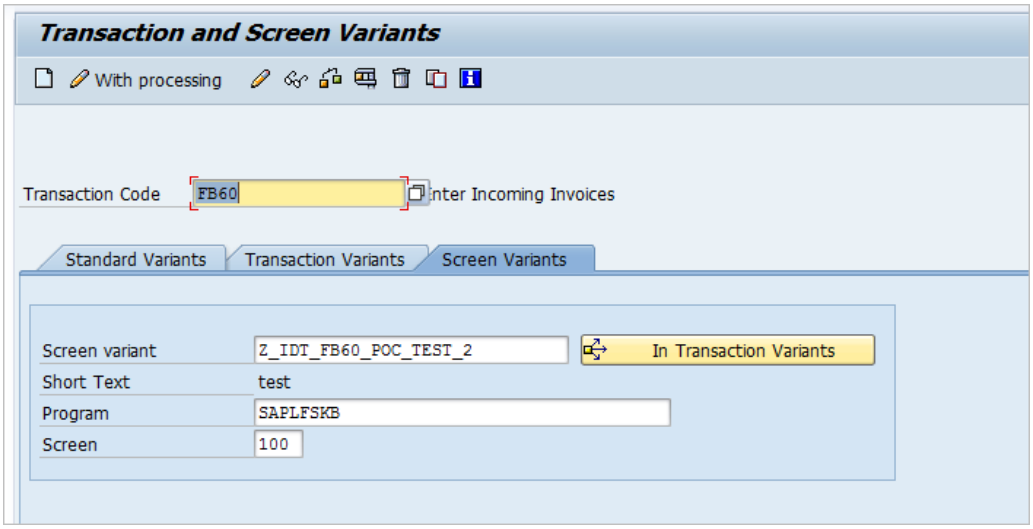


Your variant name will be different.

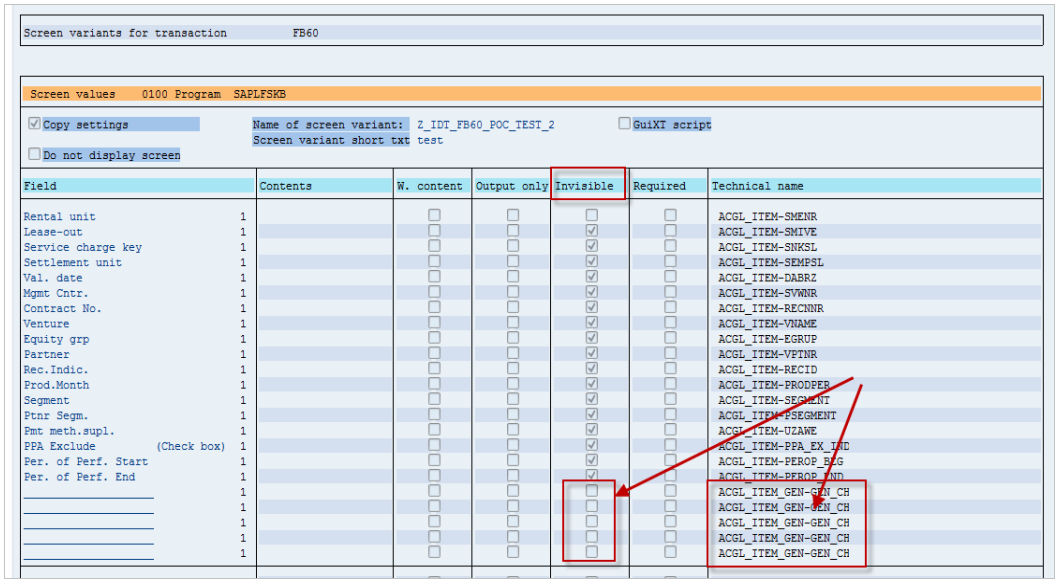


3. You may need to use transaction SHD0 to make the new field visible to the transaction.

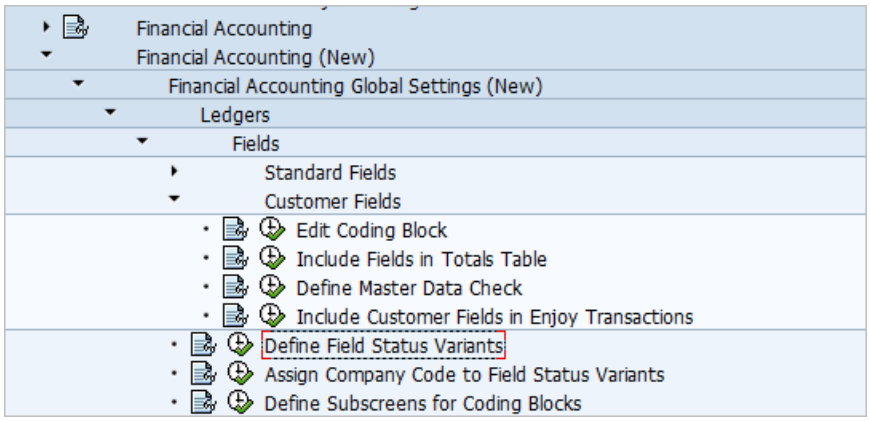
Once inside SHD0 on the screen variants tab you will want to make sure that the lines at the bottom for

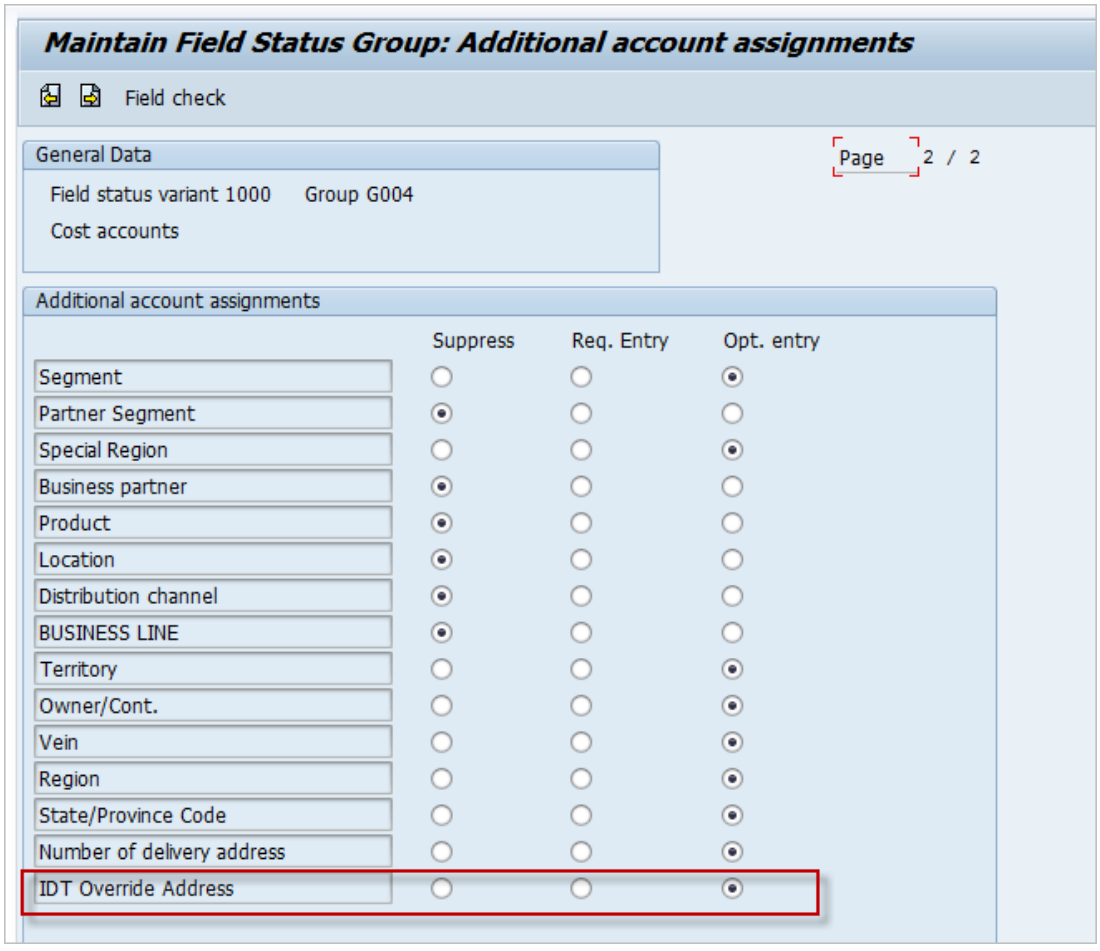


ACGL_ITEM_GEN_GEN_CH are not checked as invisible. See screen shot below:

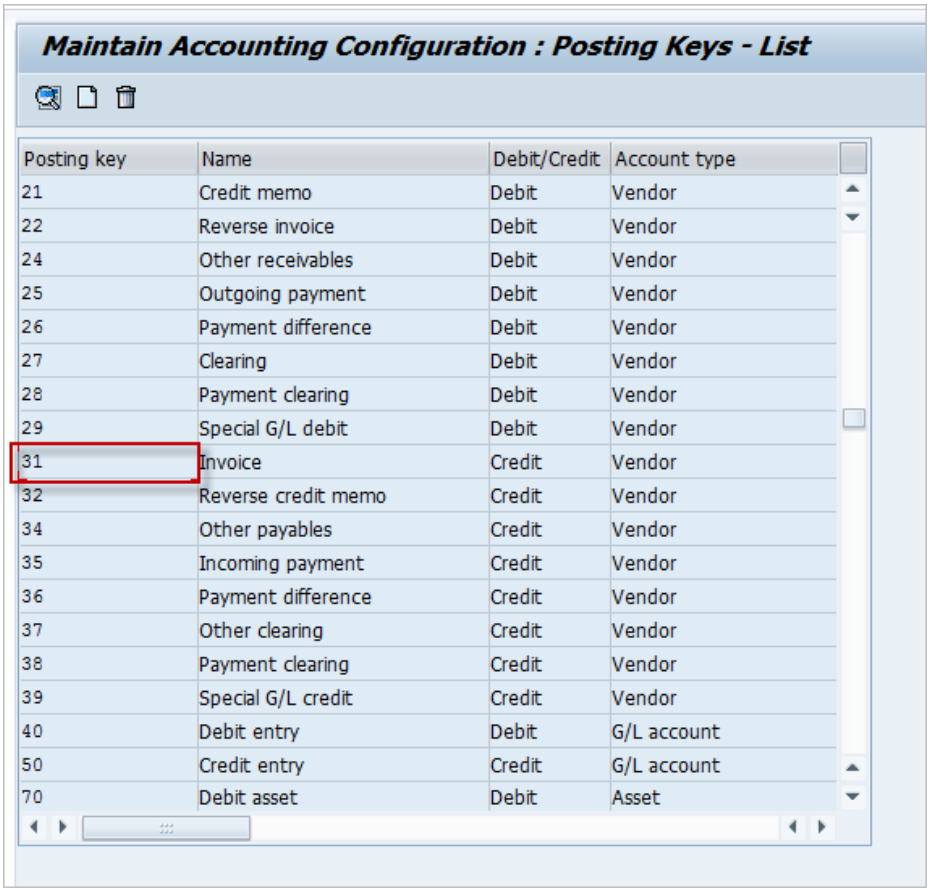


4. Use “Define Field Status Variants” to make “IDT Override Address” field editable for relevant “Field status” and “field status group”

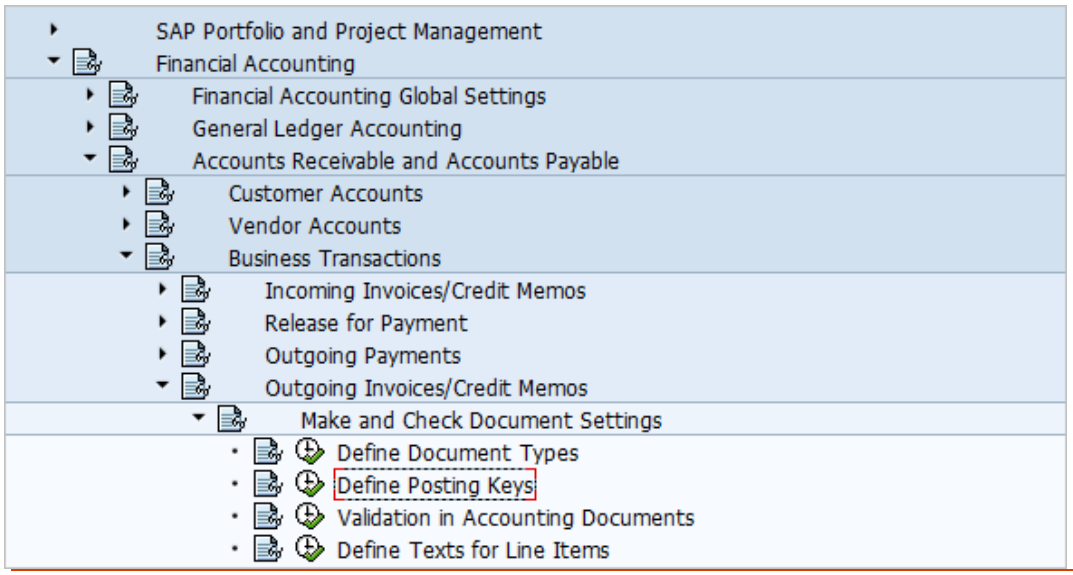


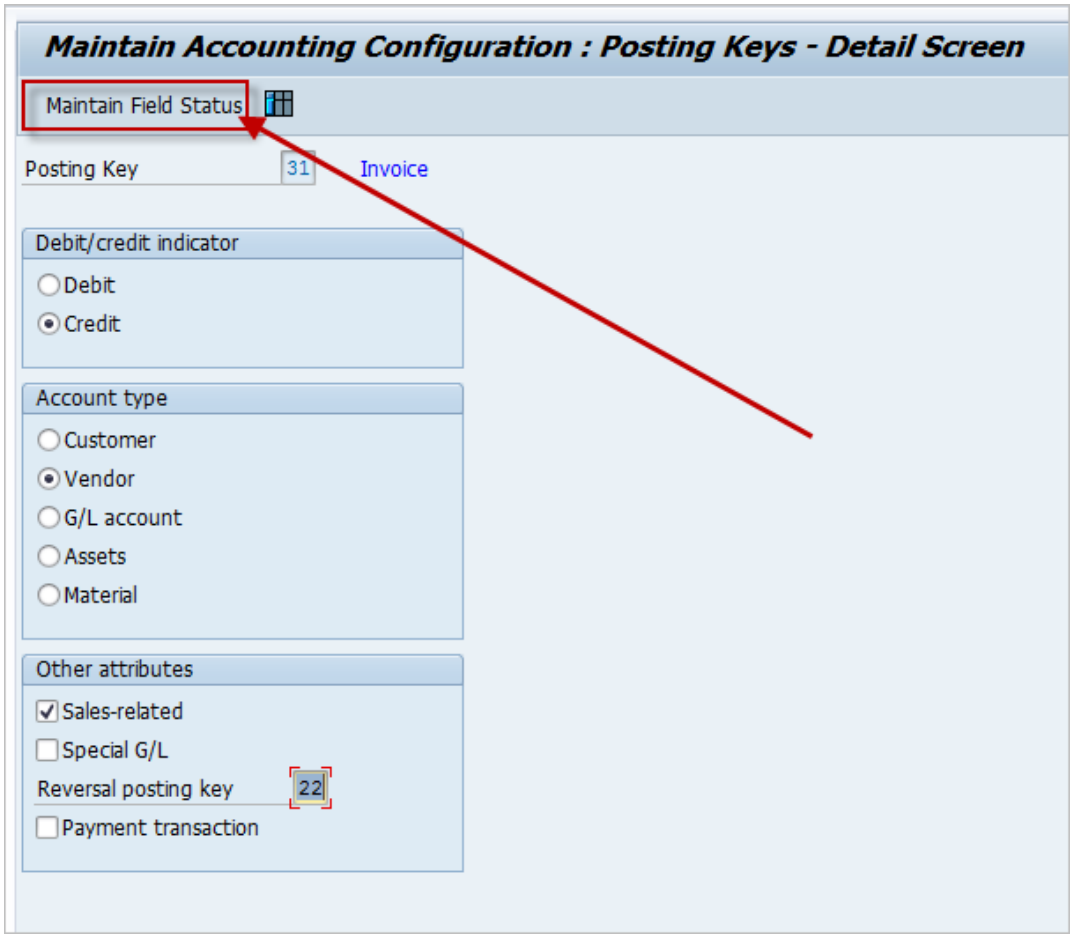


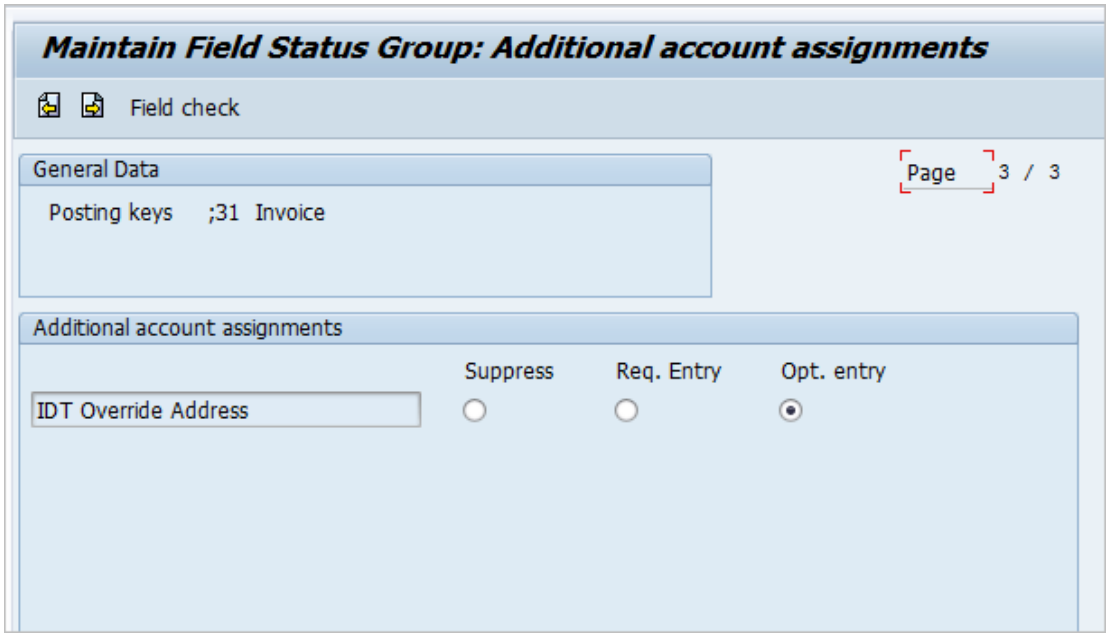
5. Use “Define Posting Keys” to make “IDT Override Address” field editable for relevant “Posting Keys”



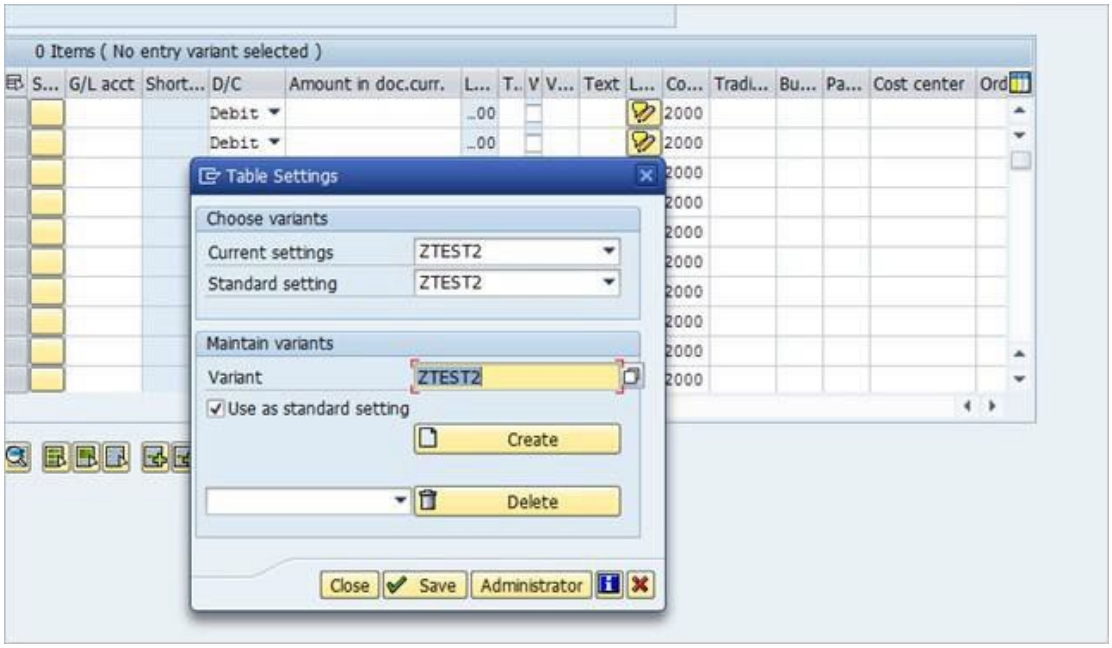
Posting key	Name	Debit/Credit	Account type
21	Credit memo	Debit	Vendor
22	Reverse invoice	Debit	Vendor
24	Other receivables	Debit	Vendor
25	Outgoing payment	Debit	Vendor
26	Payment difference	Debit	Vendor
27	Clearing	Debit	Vendor
28	Payment clearing	Debit	Vendor
29	Special G/L debit	Debit	Vendor
31	Invoice	Credit	Vendor
32	Reverse credit memo	Credit	Vendor
34	Other payables	Credit	Vendor
35	Incoming payment	Credit	Vendor
36	Payment difference	Credit	Vendor
37	Other clearing	Credit	Vendor
38	Payment clearing	Credit	Vendor
39	Special G/L credit	Credit	Vendor
40	Debit entry	Debit	G/L account
50	Credit entry	Credit	G/L account
70	Debit asset	Debit	Asset



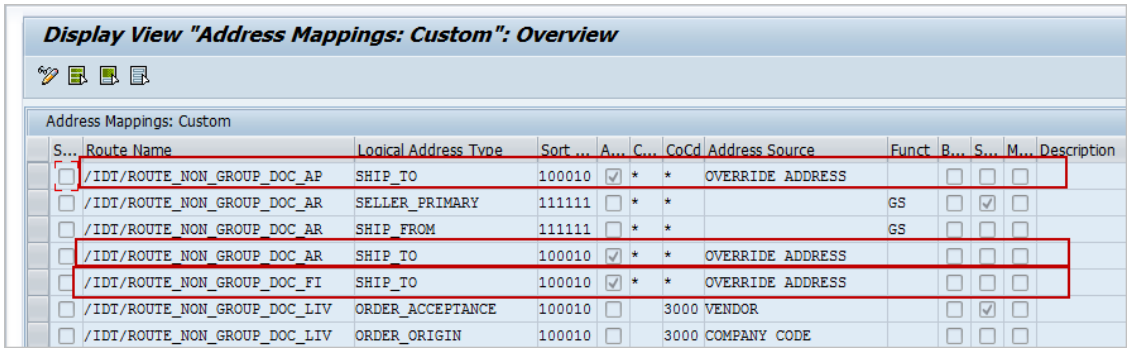




6. In transaction FB60, you may want to change the table control configuration to show the field better.



7. In transaction /IDT/ADDRESS_MAPPING, add a line like this. (If you used field name ZZIDTOVAD and data element ZZIDTOVAD then address source “OVERRIDE ADDRESS” should work automatically.

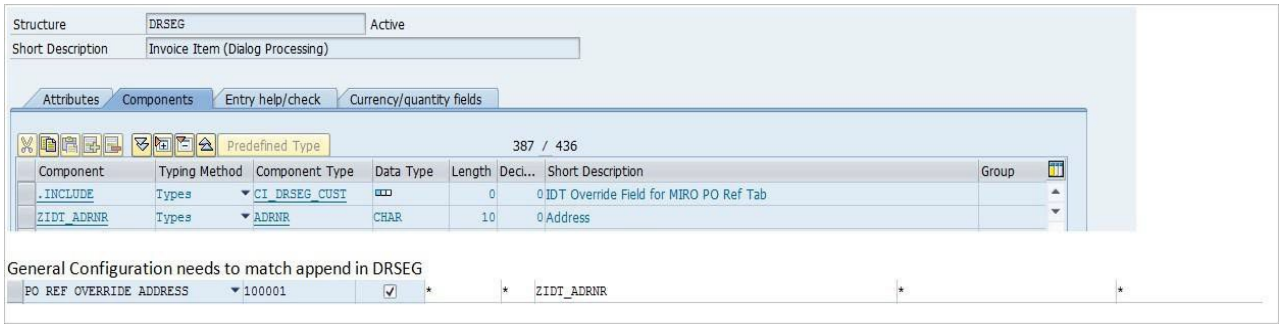


Note: that in this configuration set up we have addressed the set-up of a ship to address option to change the ship to at time of invoice. This same process can also be implemented with the ship from address if desired.

ADDING THE ZZIDTOVAD FIELD TO THE MIRO PO REFERENCE TAB

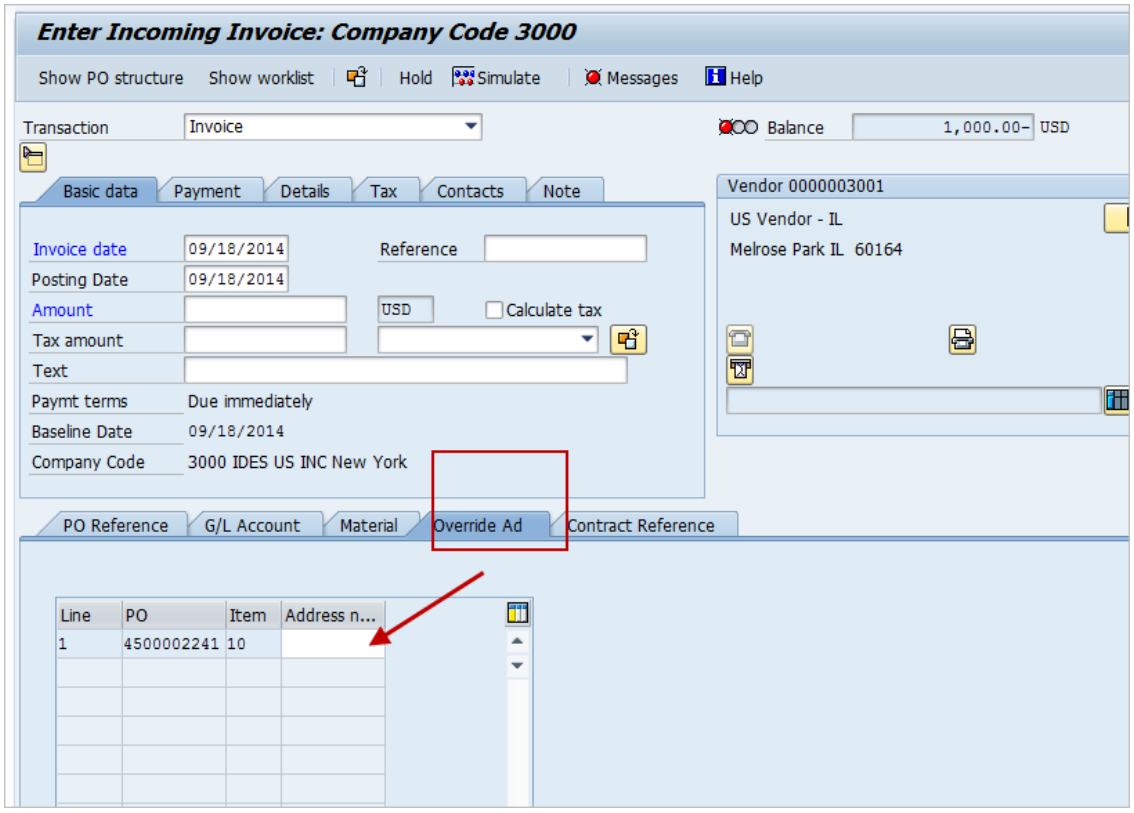
If you wish to use the IDT Address field within the MIRO invoice PO Reference tab to change the ship to address at time of invoice, there is an entirely different process than what is outlined above since the PO Reference tab on the MIRO document is tied to the MM processes rather than the G/L tab which is tied to FI module processes. The two tabs within MIRO work differently. System users have several options before them as to how to add the field to the PO Reference tab depending on their IT policies and programming procedures used. Instructions on how to do this would be cumbersome to document in entirety here. We recommend that you refer to SAP OSS note number 1156325 as this is the recommend process by SAP to add a new custom tab to the MIRO screen using BADI MRM_ITEM_CUSTFIELDS. The instructions also refer to several other LIV BADI that can be used and recommended procedures to implement.

You will have to also append the DRSEG table to add the new field to this table. The following two screens below will aid you in this process.



You will need to append the table with the new field name along with adding this line of configuration to the General Configuration table using transaction /IDT/GEN_CONFIG_VALS.

After following this recommend procedure by SAP you will have an additional tab on the LIV line-item entry screen that can be titled as desired. We have added the new custom tab in the example shown below which has a new tab labeled “Override Ad”. On this tab you can specify the line-item on the PO and add an alternate address number to override the ship to at time of payment.



Note: that in this configuration set up we have addressed the set-up of a ship to address option to change the ship to at time of invoice. This same process can also be implemented with the ship from address if desired.

ADDING ADDRESS FIELD TO UNPLANNED DELIVERY FUNCTION IN MIRO DETAILS TAB

As in this prior section we noted in the User Guide details about the use of a new address field within the Details Tab of the MIRO transaction for use with Unplanned Delivery Charges (UDC). This address field is optional and can be added by the system user via the instructions listed below.

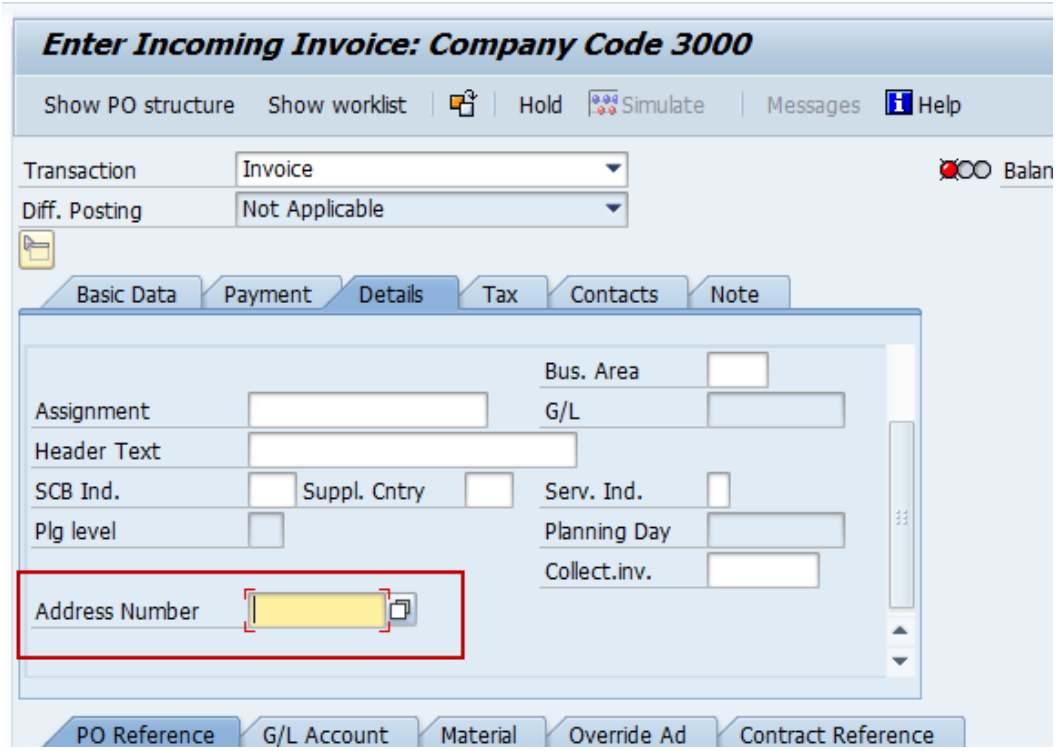
With this Integration, tax calculation is supported on UDC entered at the MIRO details tab. If UDC is entered at the details tab (header level), it can be created as a new line addition to the P.O. lines and G/L lines to a separate account number, or it can be split into PO lines by using standard configuration within SAP to set the function of G/L posting for UDC charges.

The system will apply the address of the header ship-to unless overridden by line-items ship-to addresses. If the system is configured to use a separate G/L account for posting of the unplanned delivery cost, then the user has the additional option of overriding the header level ship to address with a separate address number that is used in

a custom field within the details tab. Below instructions show how to add this optional field to the details tab for this purpose. If the user decides to configure the unplanned delivery charges based on the various line-items in the order, then this field will not be used and the line-item ship to addresses will prevail within the standard SAP allocation logic.

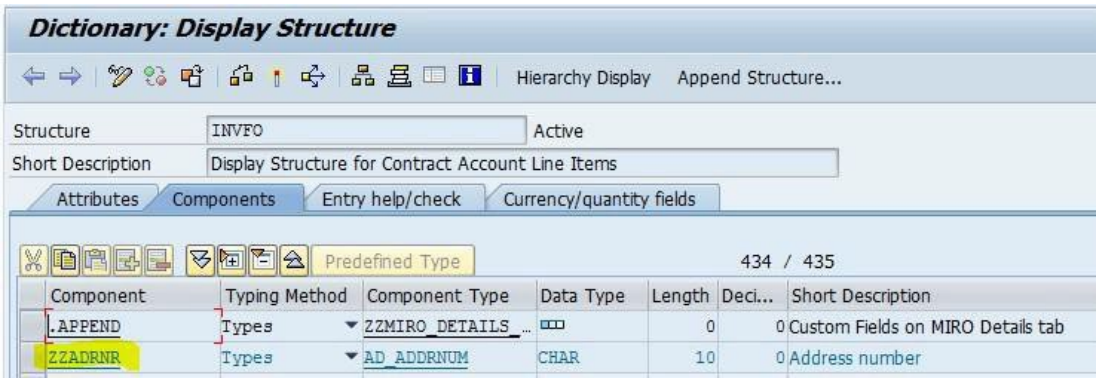
MIRO details tab needs to be enhanced to provide ship-to address number as a custom field for entry. See example below:

Note: that this is not a simple process and will need to be done by an experience ABAP programmer that is familiar with adding custom fields to the code block and screen variants.

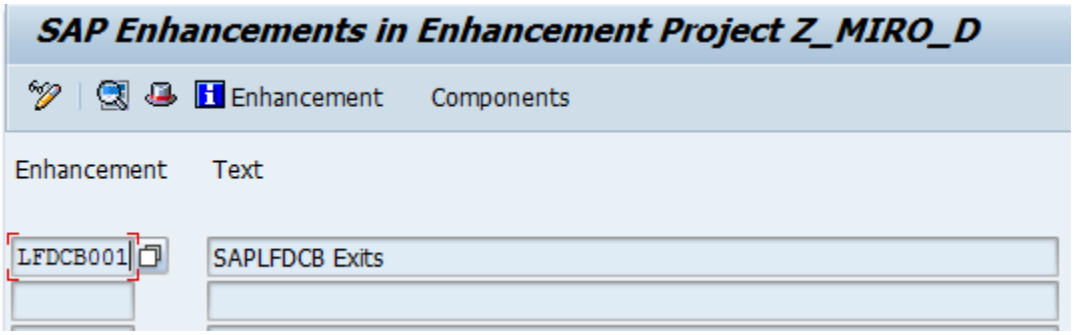


Notice that in this image the Address Number field including a search function is added at the bottom of the details tab screen. One must scroll down using the right scroll bar to the bottom of the tab to view and input an address number for this function. The field is not apparent at first view of the tab without scrolling to the bottom.

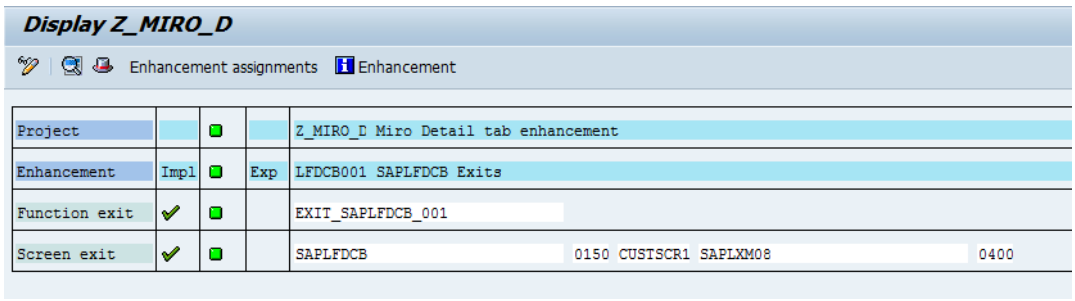
1. Enhance standard structures by adding the new field through append structure INVFO, RBKP, RBKP_V, ACMM_VENDOR_COMP



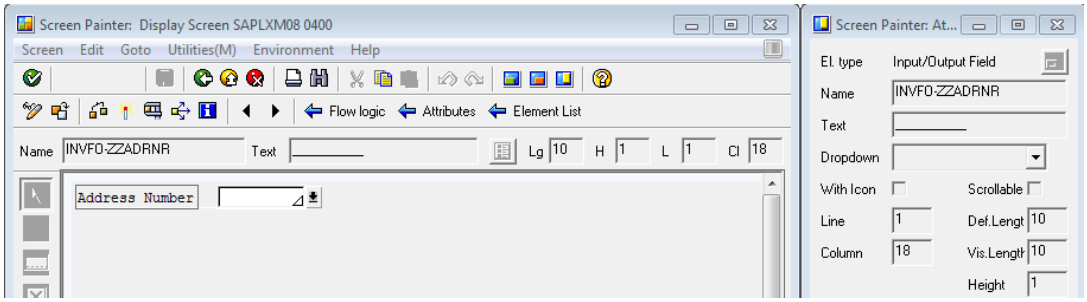
2. User exit LFDCB001 to enhance the "Details" tab screen to add new field for address.



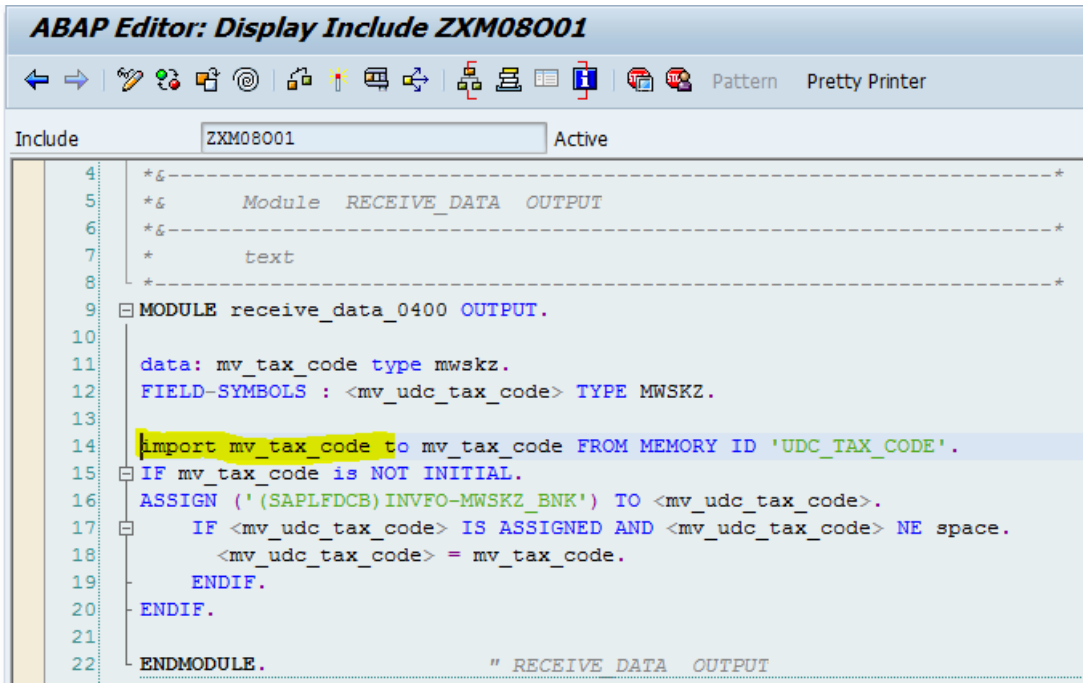
3. Create a subscreen SAPLXM08-0400.



4. Put address field on the screen.

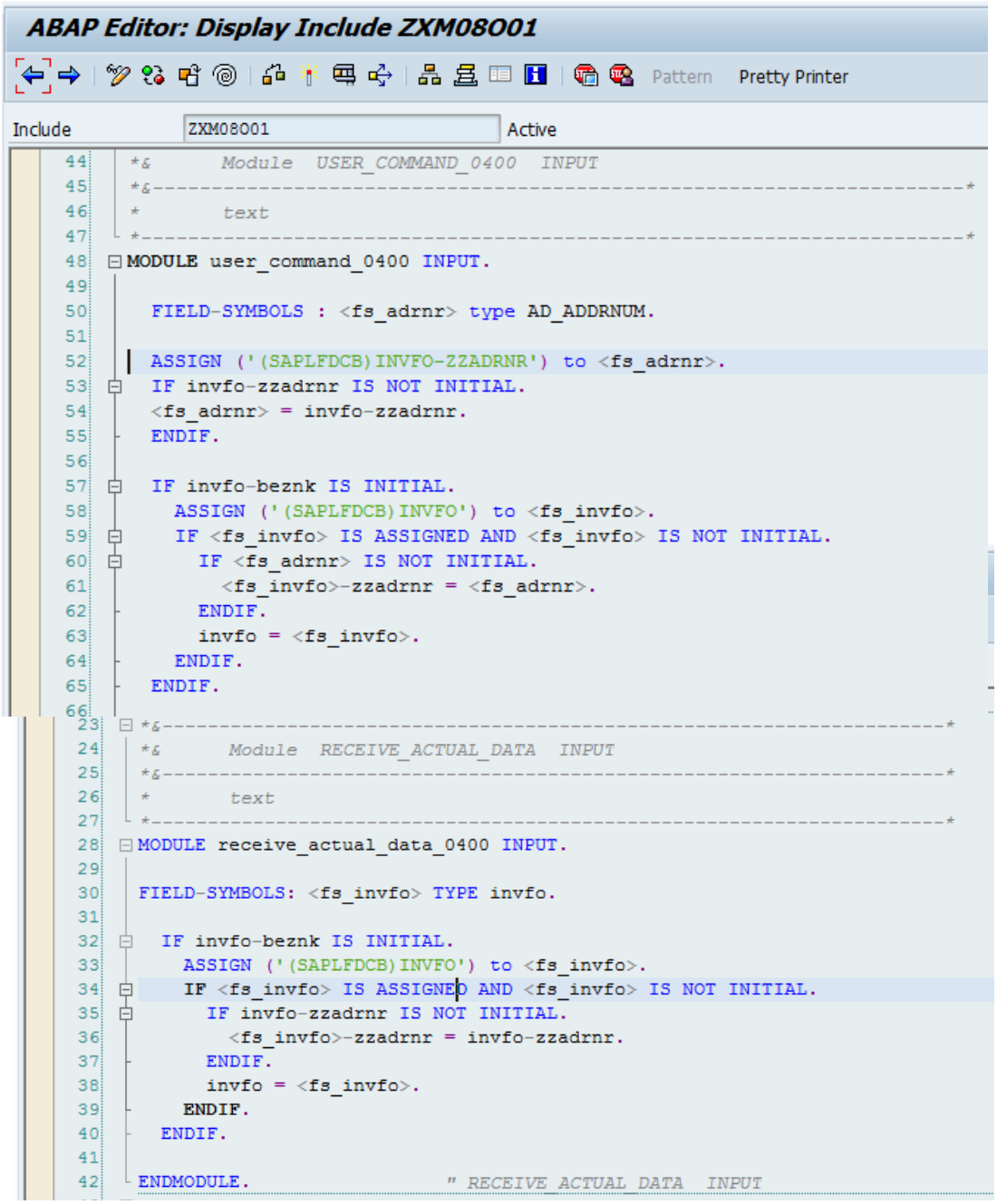


5. In the PBO of this screen below import statement is very useful as this is required to display target tax code on the MIRO details tab screen.



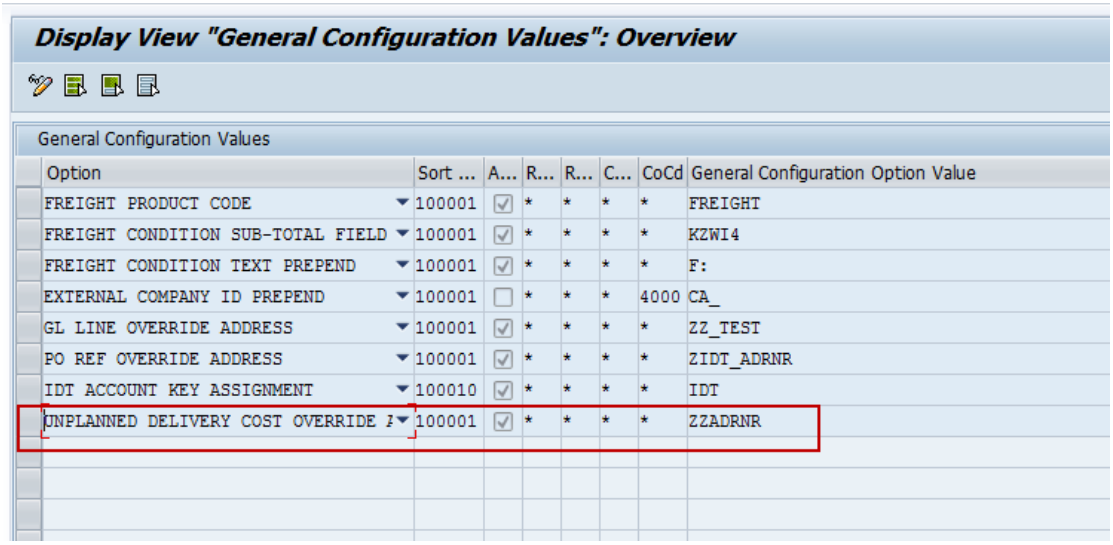
6. In the PAI method pass address number.

7. Put below code in user command module.

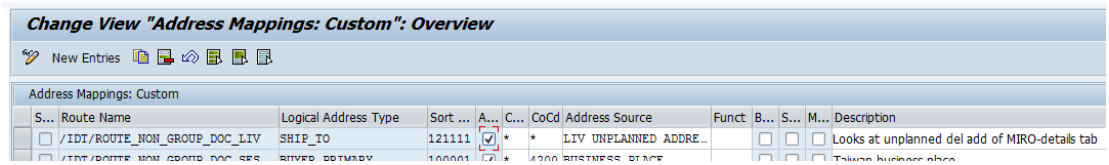


(Warning)Even if business does not need to have a custom address number on MIRO details tab, above enhancement is required to change tax code to target tax code on details tab. This is performed by the code mentioned in PBO module of above screen enhancement.

8. If custom address field is added through above screen enhancement, then the field name should be configured in the configuration table through transaction /IDT/GEN_CONFIG_VALS.



If business needs to have separate ship-to address for UDC then global next can support a ship to address to be populated by using below configuration in the custom view of the address mapping table.

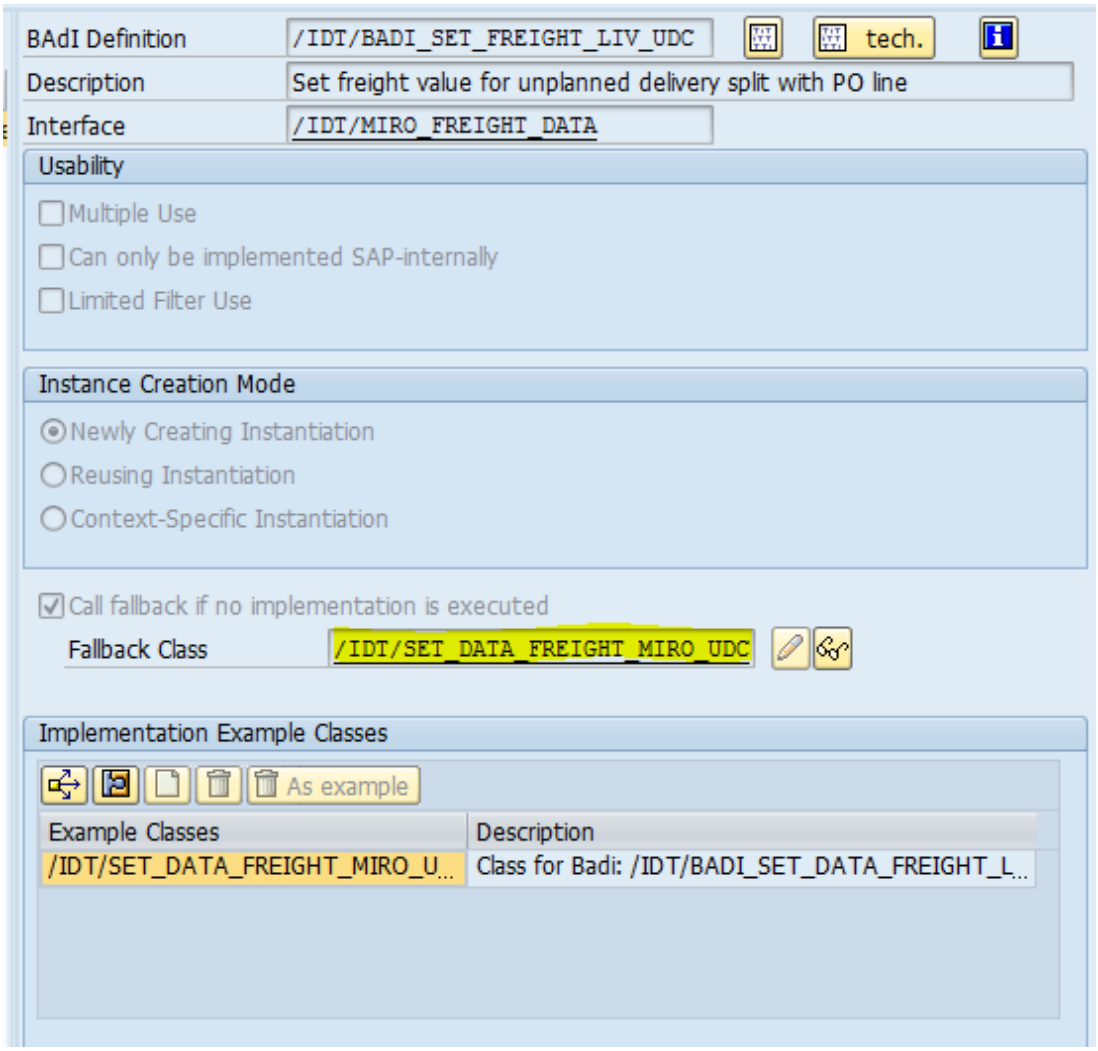


ADDING IDT ADDRESS FIELD TO SERVICE ENTRY SHEET LINE ITEM /IDT/BADI_SET_FREIGHT_LIV_UDC

The purpose of the BADI is to determine freight value only when it is entered at the MIRO details tab as an unplanned delivery charge and system configuration is set to option 0 or 1 which splits it up and includes it as part of the PO line-item amount. Every business may have different logic to determine freight value or ratio to its corresponding line-item. The method of the BADI is provided to have that logic which will override the default logic provided in the fallback class method. This BADI is not required to determine freight value if it comes from original PO line-item or by changing freight value in MIRO condition directly.

With the determined freight value, a new line for the Determination tax call will be created as freight can have different tax rules than its corresponding main line-item. This BADI uses interface /IDT/MIRO_FREIGHT_DATA and

its method SET_ATTRIBUTE_MIRO_FREIGHT. The method has input parameter as IS_SAP_ITEM which holds all required data for the PO e.g. EKKO, EKPO, DRSEG, BSEG….so on.



As part of the product, this BADI has a fallback class which will be executed if there is no implementation done from the customer. Method 'SET_ATTRIBUTE_MIRO_FREIGHT' uses below logic to extract the freight value which is entered at MIRO details tab (i.e. header level) and splits among PO line-items –

Freight_vaue = bseg-wrbtr – drseg-wrbtr.

Please note this is just an example and this logic can be overridden by creating a new implementation of this BADI in the customer's namespace.

ADDING MIRO MULTIPLE ACCOUNT ASSIGMENT ADDRESS OVERRIDES

When PO line-item is utilizing multiple account assignment functionality, Integration now creates related lines and tax blocks as per quantity/percentage distribution in the POs account assignment tab. The ship-to addresses could be different due to different cost centers or other cost object assignments. Some customers may elect to include MIRO functionality to override these addresses at time of invoice with an address number from the ADRC address table much like the options we have also provided for line level as well as G/L account tab address override logic.

Information and code sample below is provided so that your programmer can add MIRO enhancements for multiple account assignment override address.

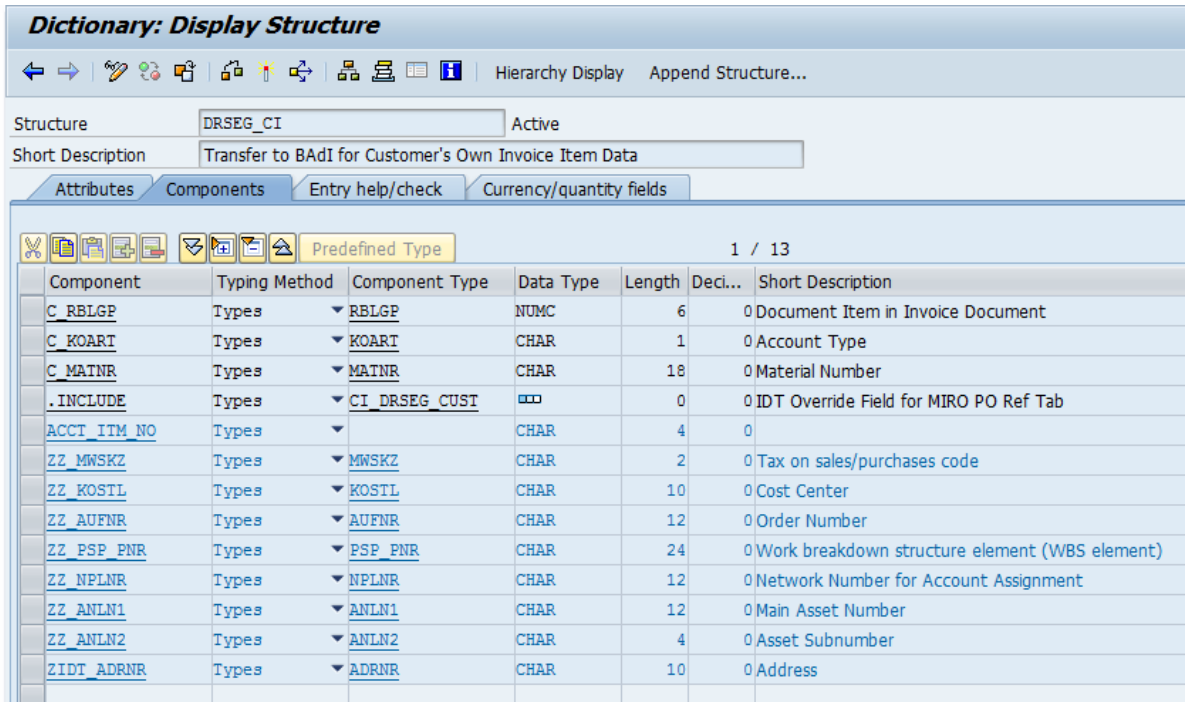
To display multiple account assignment address override fields in MIRO, you will need to perform the below noted tasks:

- Enhance/Create structures
- Implement BADI MRM_ITEM_CUSTFIELDS
- Create the sub-screen
- Write code in the PAI/PBO modules of the screen. (Sample code is attached.)

Enhance/ Create Structures

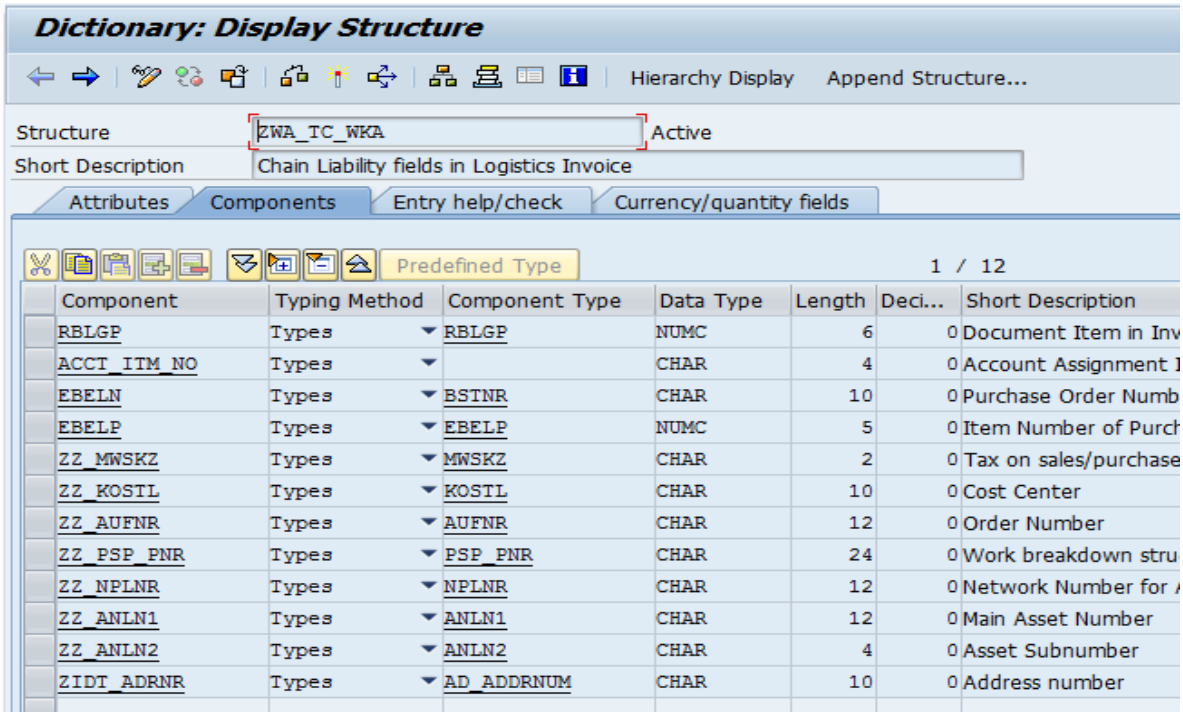
This is done to display required fields on MIRO account assignment tab.

1. DRSEG (DRSEG_CI).

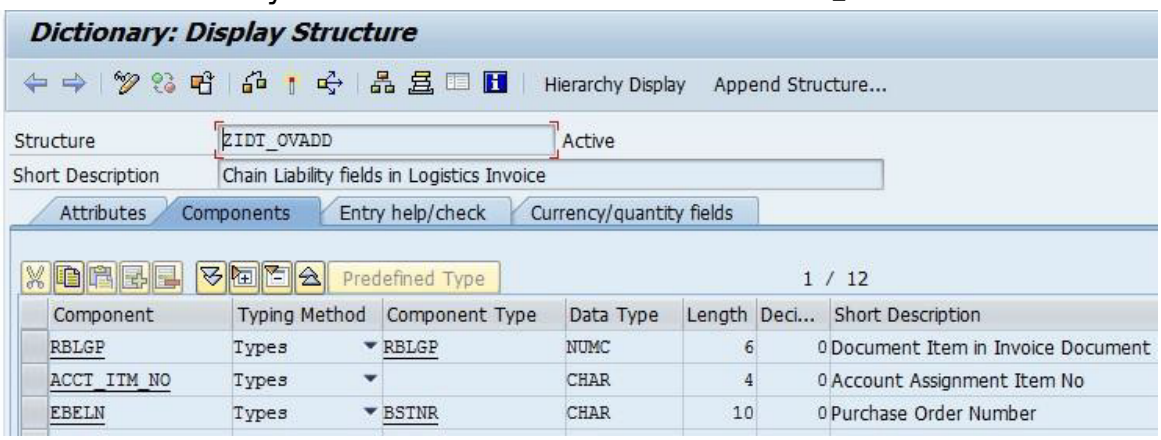


Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
C_RBLGP	Types	RBLGP	NUMC	6	0	Document Item in Invoice Document
C_KOART	Types	KOART	CHAR	1	0	Account Type
C_MATNR	Types	MATNR	CHAR	18	0	Material Number
.INCLUDE	Types	CI_DRSEG_CUST		0	0	IDT Override Field for MIRO PO Ref Tab
ACCT_ITM_NO	Types		CHAR	4	0	
ZZ_MWSKZ	Types	MWSKZ	CHAR	2	0	Tax on sales/purchases code
ZZ_KOSTL	Types	KOSTL	CHAR	10	0	Cost Center
ZZ_AUFNR	Types	AUFNR	CHAR	12	0	Order Number
ZZ_PSP_PNR	Types	PSP_PNR	CHAR	24	0	Work breakdown structure element (WBS element)
ZZ_NPLNR	Types	NPLNR	CHAR	12	0	Network Number for Account Assignment
ZZ_ANLN1	Types	ANLN1	CHAR	12	0	Main Asset Number
ZZ_ANLN2	Types	ANLN2	CHAR	4	0	Asset Subnumber
ZIDT_ADRNR	Types	ADRNR	CHAR	10	0	Address

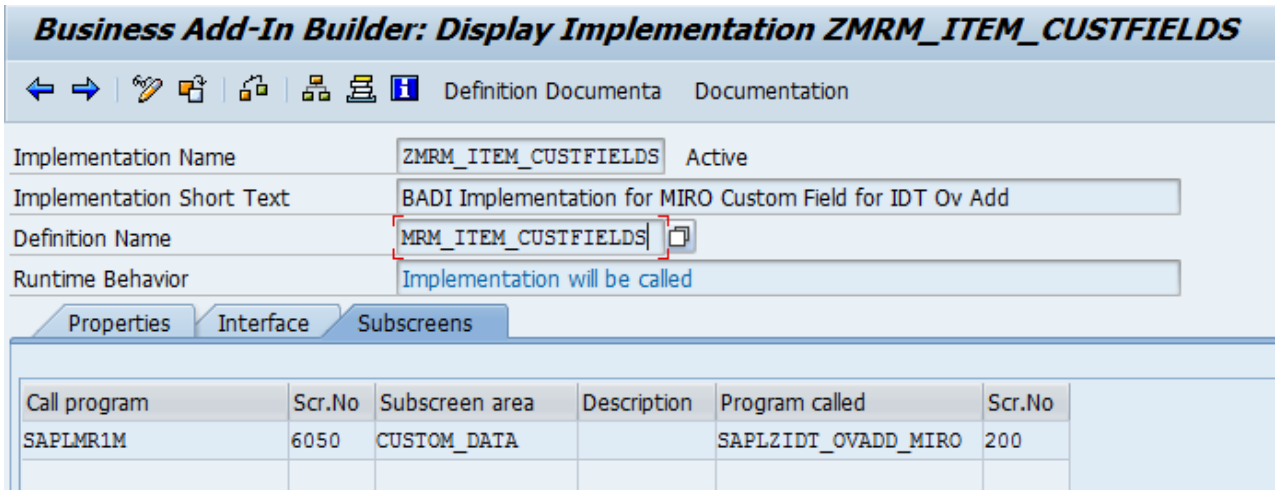
2. ZWA_TC_WKA – This structure has been used in screen enhancement code.



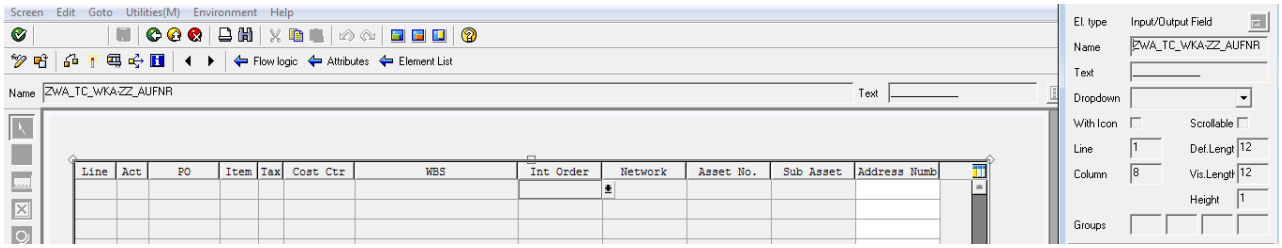
3. Create a very similar structure with the name - ZIDT_OVADD



Implement BADI MRM_ITEM_CUSTFIELDS



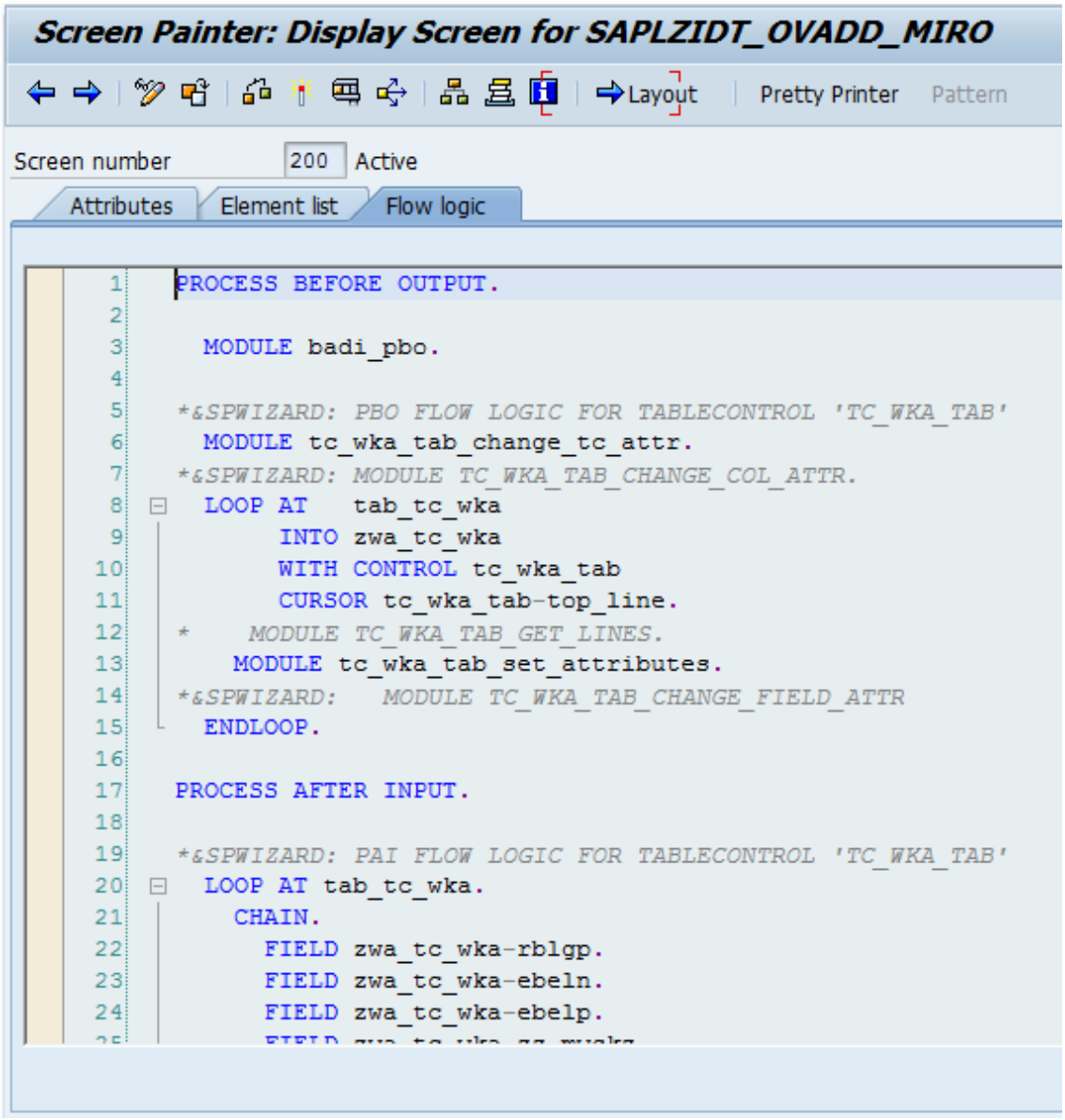
Screen 200



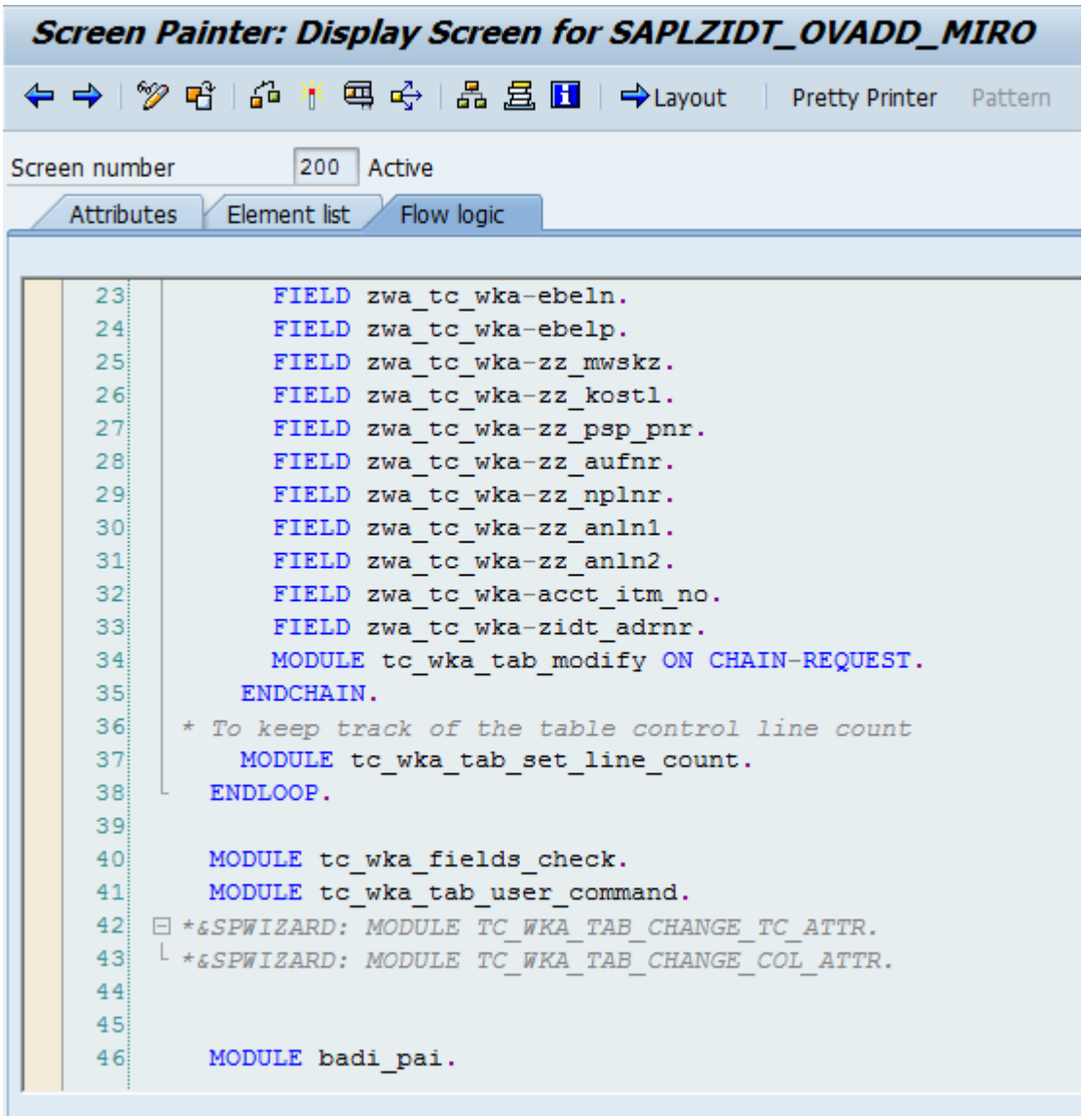
Create the sub-screen

Module pool code for this screen is attached below –

Screen 200 – Code



Flow Logic tab



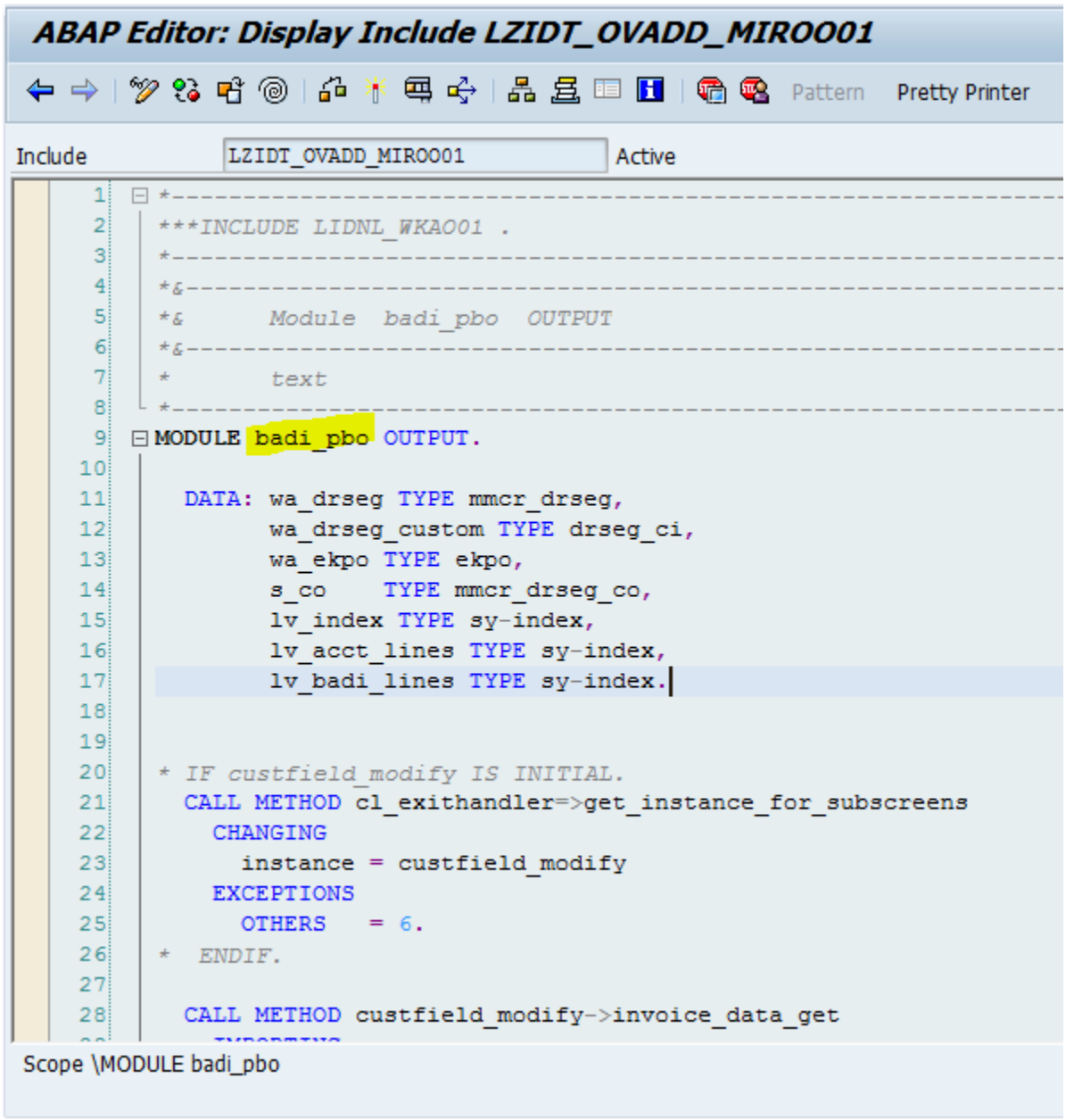
PROCESS BEFORE OUTPUT

```
MODULE BADI_PBO.

        *&SPWIZARD: PBO FLOW LOGIC FOR TABLECONTROL 'TC_WKA_TAB'
        MODULE TC_WKA_TAB_CHANGE_TC_ATTR.
        *&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_COL_ATTR.
        LOOP AT   TAB_TC_WKA
        INTO ZWA_TC_WKA
        WITH CONTROL TC_WKA_TAB
        CURSOR TC_WKA_TAB-TOP_LINE.
        *    MODULE TC_WKA_TAB_GET_LINES.
        MODULE TC_WKA_TAB_SET_ATTRIBUTES.
        *&SPWIZARD:    MODULE TC_WKA_TAB_CHANGE_FIELD_ATTR
        ENDLOOP.

        PROCESS AFTER INPUT.

        *&SPWIZARD: PAI FLOW LOGIC FOR TABLECONTROL 'TC_WKA_TAB'
        LOOP AT TAB_TC_WKA.
        CHAIN.
        FIELD ZWA_TC_WKA-RBLGP.
        FIELD ZWA_TC_WKA-EBELN.
        FIELD ZWA_TC_WKA-EBELP.
        FIELD ZWA_TC_WKA-ZZ_MWSKZ.
        FIELD ZWA_TC_WKA-ZZ_KOSTL.
        FIELD ZWA_TC_WKA-ZZ_PSP_PNR.
        FIELD ZWA_TC_WKA-ZZ_AUFNR.
        FIELD ZWA_TC_WKA-ZZ_NPLNR.
        FIELD ZWA_TC_WKA-ZZ_ANLN1.
        FIELD ZWA_TC_WKA-ZZ_ANLN2.
        FIELD ZWA_TC_WKA-ACCT_ITM_NO.
        FIELD ZWA_TC_WKA-ZIDT_ADRNR.
        MODULE TC_WKA_TAB_MODIFY ON CHAIN-REQUEST.
        ENDCHAIN.
        * TO KEEP TRACK OF THE TABLE CONTROL LINE COUNT
        MODULE TC_WKA_TAB_SET_LINE_COUNT.
        ENDLOOP.

        MODULE TC_WKA_FIELDS_CHECK.
        MODULE TC_WKA_TAB_USER_COMMAND.
        *&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_TC_ATTR.
        *&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_COL_ATTR.
```

White code in the PAI/PBO modules of the screen

MODULE BADI_PAI

PBO modules



Module BADI _PBO

```
MODULE BADI_PBO OUTPUT.

        DATA: WA_DRSEG TYPE MMCR_DRSEG,
        WA_DRSEG_CUSTOM TYPE  DRSEG_CI,
        WA_EKPO TYPE EKPO,
        S_CO    TYPE MMCR_DRSEG_CO,
        LV_INDEX TYPE SY-INDEX,
        LV_ACCT_LINES TYPE SY-INDEX,
        LV_BADI_LINES TYPE SY-INDEX.

        CALL METHOD CL_EXITHANDLER=>GET_INSTANCE_FOR_SUBSCREENS
        CHANGING
        INSTANCE = CUSTFIELD_MODIFY
        EXCEPTIONS
        OTHERS   = 6.

        CALL METHOD CUSTFIELD_MODIFY->INVOICE_DATA_GET
        IMPORTING
        E_TRANSACTION_TYPE = AKT_TYP
        ES_RBKPV           = RBKPV
        ET_DRSEG           = T_DRSEG
        ET_DRSEG_CUST      = T_DRSEG_CUSTOM.

        CLEAR: ZWA_TC_WKA, TAB_TC_WKA.
        REFRESH: TAB_TC_WKA.

        LOOP AT T_DRSEG INTO WA_DRSEG.
        CLEAR LV_INDEX.
        MOVE WA_DRSEG-RBLGP TO ZWA_TC_WKA-RBLGP.
        MOVE WA_DRSEG-EBELN TO ZWA_TC_WKA-EBELN.
        MOVE WA_DRSEG-EBELP TO ZWA_TC_WKA-EBELP.

        IF WA_DRSEG-CO IS INITIAL.

        MOVE WA_DRSEG-KOSTL      TO ZWA_TC_WKA-ZZ_KOSTL.
        MOVE WA_DRSEG-AUFNR      TO ZWA_TC_WKA-ZZ_AUFNR.
        MOVE WA_DRSEG-PS_PSP_PNR TO ZWA_TC_WKA-ZZ_PSP_PNR.
        MOVE WA_DRSEG-NPLNR      TO ZWA_TC_WKA-ZZ_NPLNR.
        MOVE WA_DRSEG-ANLN1      TO ZWA_TC_WKA-ZZ_ANLN1.
        MOVE WA_DRSEG-ANLN2      TO ZWA_TC_WKA-ZZ_ANLN2.

        CALL FUNCTION 'CONVERSION_EXIT_ABPSP_OUTPUT'
        EXPORTING
        INPUT  = ZWA_TC_WKA-ZZ_PSP_PNR
```

```
        IMPORTING
        OUTPUT = ZWA_TC_WKA-ZZ_PSP_PNR.

        IF WKA_START IS INITIAL.
        IF NOT WA_DRSEG-BELNR IS INITIAL.
        MOVE-CORRESPONDING WA_DRSEG TO ZWA_TC_WKA.
        ELSE.
        SELECT SINGLE *  FROM EKPO INTO WA_EKPO
        WHERE EBELN EQ ZWA_TC_WKA-EBELN
        AND EBELP EQ ZWA_TC_WKA-EBELP.
        IF SY-SUBRC IS INITIAL.
        MOVE-CORRESPONDING WA_EKPO TO ZWA_TC_WKA.
        ZWA_TC_WKA-RBLGP = WA_DRSEG-RBLGP.
        ENDIF.
        ENDIF.
        ELSE.
        READ TABLE T_DRSEG_CUSTOM INTO WA_DRSEG_CUSTOM
        WITH KEY C_RBLGP = WA_DRSEG-RBLGP
        C_KOART = WA_DRSEG-KOART.
        IF SY-SUBRC IS INITIAL AND WA_DRSEG_CUSTOM-ACCT_ITM_NO IS INITIAL.
        MOVE-CORRESPONDING WA_DRSEG_CUSTOM TO ZWA_TC_WKA.
        MOVE WA_DRSEG-ZIDT_ADRNR TO ZWA_TC_WKA-ZIDT_ADRNR.
        ELSE.
        IF AKT_TYP = 'A'.
        MOVE-CORRESPONDING WA_DRSEG TO ZWA_TC_WKA.
        ELSE.
        CLEAR ZWA_TC_WKA-ZIDT_ADRNR.
        ENDIF.
        ENDIF.
        ENDIF.
        MOVE WA_DRSEG-MWSKZ TO ZWA_TC_WKA-ZZ_MWSKZ.
        CLEAR ZWA_TC_WKA-ACCT_ITM_NO.
        APPEND ZWA_TC_WKA TO TAB_TC_WKA.
        ELSE.

        CLEAR LV_ACCT_LINES.
        DESCRIBE TABLE WA_DRSEG-CO LINES LV_ACCT_LINES.
        CLEAR LV_BADI_LINES.
        LOOP AT T_DRSEG_CUSTOM INTO WA_DRSEG_CUSTOM WHERE C_RBLGP = WA_DRSEG-
RBLGP.
        ADD 1 TO LV_BADI_LINES.
        ENDLOOP.
        CLEAR WA_DRSEG_CUSTOM.
        LOOP AT WA_DRSEG-CO INTO S_CO." WHERE SELKZ IS NOT INITIAL.
```

```
ADD 1 TO LV_INDEX.

IF WKA_START IS INITIAL.
IF NOT WA_DRSEG-BELNR IS INITIAL.
CLEAR S_CO.
READ TABLE WA_DRSEG-CO INTO S_CO INDEX LV_INDEX.
MOVE S_CO-ZIDT_ADRNR TO ZWA_TC_WKA-ZIDT_ADRNR.
ELSE.
SELECT SINGLE * FROM EKPO INTO WA_EKPO
WHERE EBELN EQ ZWA_TC_WKA-EBELN
AND EBELP EQ ZWA_TC_WKA-EBELP.
IF SY-SUBRC IS INITIAL.
MOVE-CORRESPONDING WA_EKPO TO ZWA_TC_WKA.
ZWA_TC_WKA-RBLGP = WA_DRSEG-RBLGP.
ENDIF.
ENDIF.
ELSE.
READ TABLE T_DRSEG_CUSTOM INTO WA_DRSEG_CUSTOM
WITH KEY C_RBLGP     = WA_DRSEG-RBLGP
C_KOART      = WA_DRSEG-KOART
ACCT_ITM_NO = LV_INDEX.
IF SY-SUBRC IS INITIAL AND ( LV_ACCT_LINES = LV_BADI_LINES ).
MOVE-CORRESPONDING WA_DRSEG_CUSTOM TO ZWA_TC_WKA.
ELSE.
IF AKT_TYP = 'A'.
ZWA_TC_WKA-ZIDT_ADRNR = S_CO-ZIDT_ADRNR.
ELSE.
CLEAR ZWA_TC_WKA-ZIDT_ADRNR.
ENDIF.
ENDIF.
ENDIF.

MOVE S_CO-KOSTL      TO ZWA_TC_WKA-ZZ_KOSTL.
MOVE LV_INDEX        TO ZWA_TC_WKA-ACCT_ITM_NO.
MOVE S_CO-AUFNR      TO ZWA_TC_WKA-ZZ_AUFNR.
MOVE S_CO-PS_PSP_PNR TO ZWA_TC_WKA-ZZ_PSP_PNR.
MOVE S_CO-ANLN1      TO ZWA_TC_WKA-ZZ_ANLN1.
MOVE S_CO-ANLN2      TO ZWA_TC_WKA-ZZ_ANLN2.
MOVE S_CO-NPLNR      TO ZWA_TC_WKA-ZZ_NPLNR.

CALL FUNCTION 'CONVERSION_EXIT_ABPSP_OUTPUT'
EXPORTING
INPUT  = ZWA_TC_WKA-ZZ_PSP_PNR
IMPORTING
```

```
    OUTPUT = ZWA_TC_WKA-ZZ_PSP_PNR.

    MOVE WA_DRSEG-MWSKZ TO ZWA_TC_WKA-ZZ_MWSKZ.

    APPEND ZWA_TC_WKA TO TAB_TC_WKA.

    ENDLOOP.
    ENDIF.
    ENDLOOP.
    WKA_START = 'X'.

    ENDMODULE.                    " BADI PBO  OUTPUT
```

Module TC_WKA_TAB_CHANGE_TC_ATTRIBUTES

```
MODULE TC_WKA_TAB_CHANGE_TC_ATTR OUTPUT.

        DESCRIBE TABLE TAB_TC_WKA LINES TC_WKA_TAB-LINES.

        IF AKT_TYP = C_TRTYP_A.

        LOOP AT TC_WKA_TAB-COLS INTO COL.
        IF COL-SCREEN-NAME EQ 'ZWA_TC_WKA-ZIDT_ADRNR'.
        COL-SCREEN-INPUT = 0.
        MODIFY TC_WKA_TAB-COLS FROM COL.
        ENDIF.
        ENDLOOP.
        ENDIF.

        ENDMODULE.                    "TC_WKA_TAB_CHANGE_TC_ATTR OUTPUT
```

Module TC_WKA_TAB_SET_ATTRIBUTES

```
MODULE TC_WKA_TAB_SET_ATTRIBUTES OUTPUT.

        REFRESH CONTROL 'TC_WKA_TAB' FROM SCREEN '200'.  "DEFAULT

        ENDMODULE.                 " TC_WKA_TAB_SET_ATTRIBUTES  OUTPUT
```

Module TC_WKA_TAB_MODIFY

```
MODULE TC_WKA_TAB_MODIFY INPUT.
        MODIFY TAB_TC_WKA
        FROM ZWA_TC_WKA
        INDEX TC_WKA_TAB-CURRENT_LINE.
        ENDMODULE.
```

Module TC_WKA_TAB_SET_LINE_COUNT

```
MODULE TC_WKA_TAB_SET_LINE_COUNT INPUT.
        TC_WKA_TAB-LINES = SY-LOOPC.
        ENDMODULE.                 " TC_WKA_TAB_SET_LINE_COUNT  INPUT
```

Module TC_WKA_FIELDS_CHECK

```
MODULE TC_WKA_FIELDS_CHECK INPUT.

        LOOP AT TAB_TC_WKA INTO ZWA_TC_WKA.
        IF ZWA_TC_WKA-ZIDT_ADRNR IS NOT INITIAL.
        DATA V_ADRNR TYPE ADRC.
        SELECT SINGLE * FROM ADRC INTO V_ADRNR
        WHERE ADDRNUMBER = ZWA_TC_WKA-ZIDT_ADRNR.

        IF SY-SUBRC <> 0.
        MESSAGE 'INVALID ADDRESS NUMBER' TYPE 'E'.
        ENDIF.
        ENDIF.
        ENDLOOP.

        ENDMODULE.                 " TC_WKA_FIELDS_CHECK  INPUT
```

Module TC_WKA_TAB_USER_COMAND

```
MODULE TC_WKA_TAB_USER_COMMAND INPUT.
        OK_CODE = SY-UCOMM.
        PERFORM USER_OK_TC USING    'TC_WKA_TAB'
        'TAB_TC_WKA'
        ' '
        CHANGING OK_CODE.
        SY-UCOMM = OK_CODE.
        ENDMODULE.
```

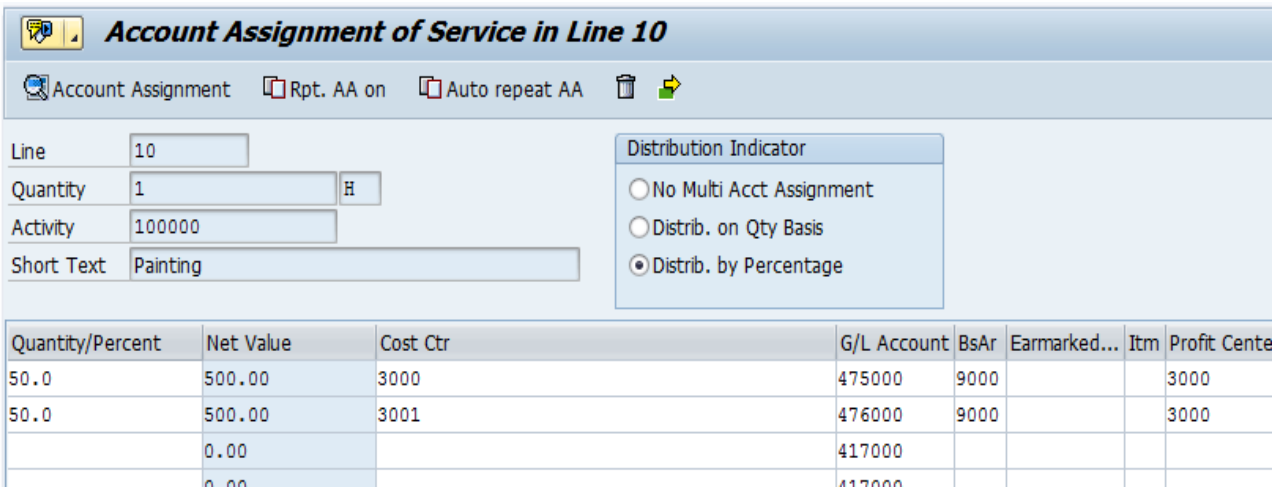
Module BADI_PAI

```
MODULE BADI_PAI INPUT.

        DATA: H_SORT TYPE BOOLE-BOOLE,
        LV_INDEX1 TYPE SY-INDEX.

        IF NOT ( AKT_TYP = C_TRTYP_A ).

        CLEAR: T_DRSEG_CUSTOM.
        REFRESH T_DRSEG_CUSTOM.



        LOOP AT T_DRSEG INTO WA_DRSEG.
        CLEAR LV_INDEX1.
        IF WA_DRSEG-CO IS INITIAL.

        CLEAR WA_DRSEG_CUSTOM.
        MOVE-CORRESPONDING WA_DRSEG TO WA_DRSEG_CUSTOM.
        READ TABLE TAB_TC_WKA INTO ZWA_TC_WKA
        WITH KEY RBLGP = WA_DRSEG-RBLGP.
        IF SY-SUBRC EQ 0.
        MOVE-CORRESPONDING ZWA_TC_WKA TO WA_DRSEG_CUSTOM.
        WA_DRSEG_CUSTOM-C_RBLGP = WA_DRSEG-RBLGP.
        ENDIF.
        CLEAR WA_DRSEG_CUSTOM-ACCT_ITM_NO.
        APPEND WA_DRSEG_CUSTOM TO T_DRSEG_CUSTOM.
        ELSE.
        LOOP AT WA_DRSEG-CO INTO S_CO." WHERE SELKZ IS NOT INITIAL.
        ADD 1 TO LV_INDEX1.
        CLEAR WA_DRSEG_CUSTOM.
        MOVE-CORRESPONDING WA_DRSEG TO WA_DRSEG_CUSTOM.
        READ TABLE TAB_TC_WKA INTO ZWA_TC_WKA WITH KEY RBLGP = WA_DRSEG-RBLGP
ACCT_ITM_NO = LV_INDEX1.
        IF SY-SUBRC EQ 0. "AND ZWAA_TC_WKA-RBLGP = WA_DRSEG-RBLGP.
        MOVE-CORRESPONDING ZWA_TC_WKA TO WA_DRSEG_CUSTOM.
        WA_DRSEG_CUSTOM-C_RBLGP = WA_DRSEG-RBLGP.
        ENDIF.
        APPEND WA_DRSEG_CUSTOM TO T_DRSEG_CUSTOM.
        ENDLOOP.

        ENDIF.

        ENDLOOP.
```

```
CALL METHOD CUSTFIELD_MODIFY->CUSTOM_DATA_TRANSFER
EXPORTING
I_SORT        = H_SORT
IT_DRSEG_CUST = T_DRSEG_CUSTOM
I_CHANGE      = 'X'.

ENDIF.

ENDMODULE.                    " BADI_PAI  INPUT
```

SUPPORT OF MULTIPLE ACCOUNT ASSIGNMENT (MAA) IN ML81N

The hook mentioned below is applicable to those customers who will be using MAA in ML81N and will have taxes calculated on cost object. This hook is introduced because standard SAP distributes taxes as per the percentage given in MAA in account assignment of services window.



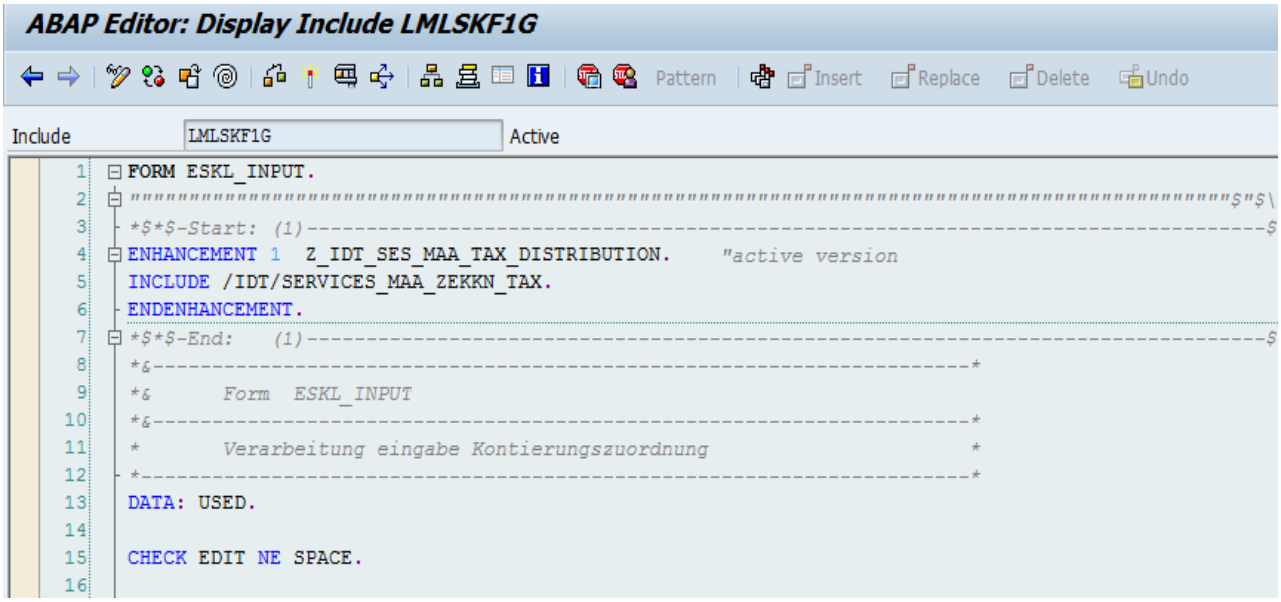
But in such cases both the lines will have different tax amounts due to different cost centers, but standard SAP distributes total tax amount for that service with the percentage provided in above window which would be wrong. Hence below implicit enhancement is created to distribute the taxes correctly.

Include - LMLSKF1G

Form name - ESKL_INPUT

Create enhancement at the beginning of this routine and include the program –

INCLUDE /IDT/SERVICES_MAA_ZEKKN_TAX.



SUPPORT OF OVERRIDE ADDRESS-ADDRESS NUMBER AT SERVICE LEVEL IN PO AND ML81N

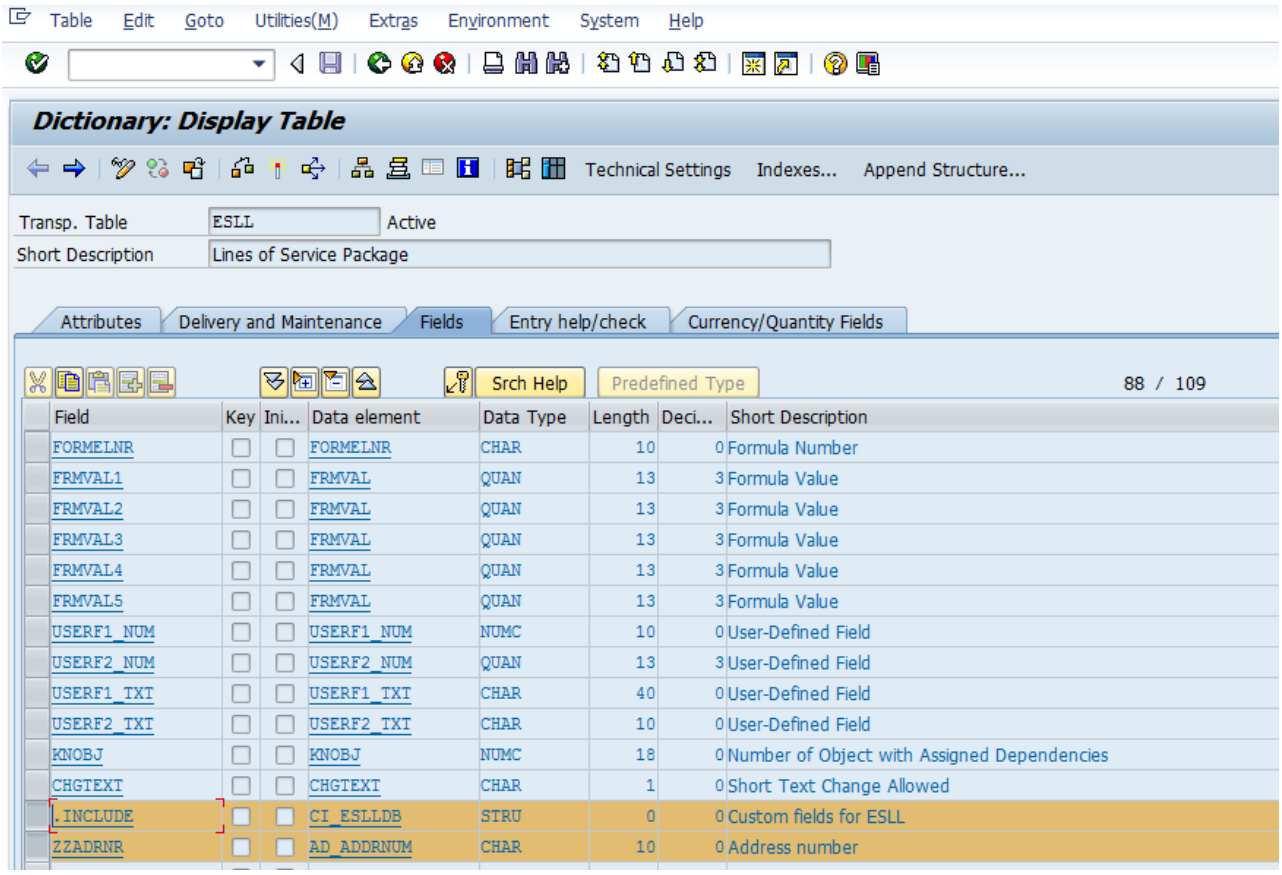
When PO line-item is utilizing Service Entry Sheet functionality with multiple account assignment, Integration now creates related lines and tax blocks as per quantity/percentage distribution in the POs SES account assignment tab. The ship-to addresses could be different due to different cost centers or other cost object assignments.

Some customers may elect to include ME21N and ML81N functionality to override these addresses at time of entry with an address number from the ADRC address table much like the options we have also provided for line level as well as G/L account tab address override logic.

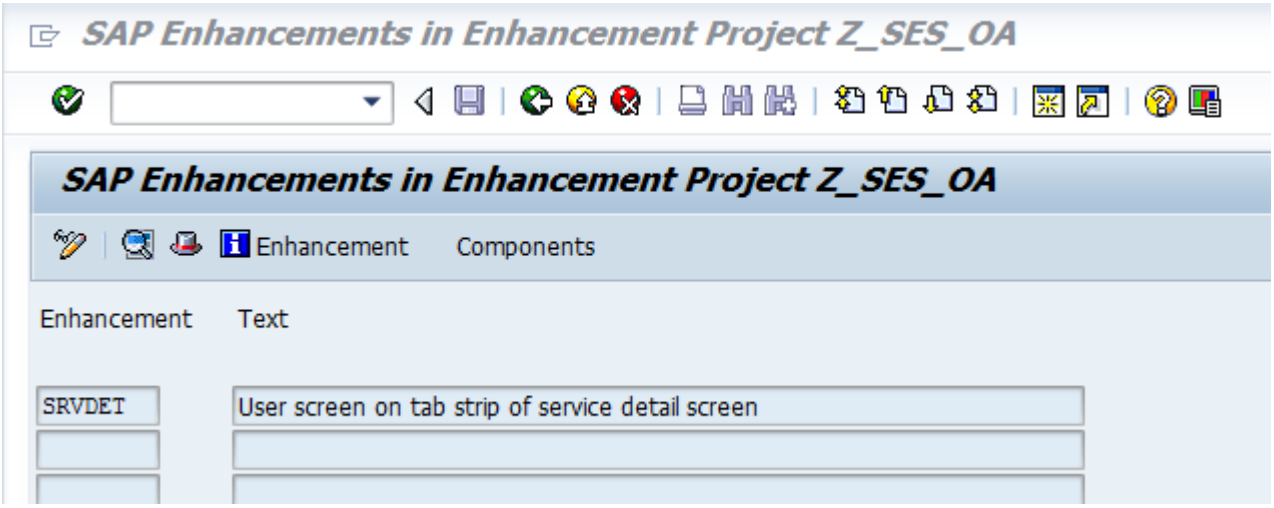
The new tab on the screen will be common for both PO and SES transactions. Override address entered on the PO will get carried over to the SES when the same PO is used as a reference. Users will have the option to change this address if required and the new address will be used based on the mapping.

Sample Code to be implemented for Adding the Override Address in ME21N and ML81N is shown for reference.

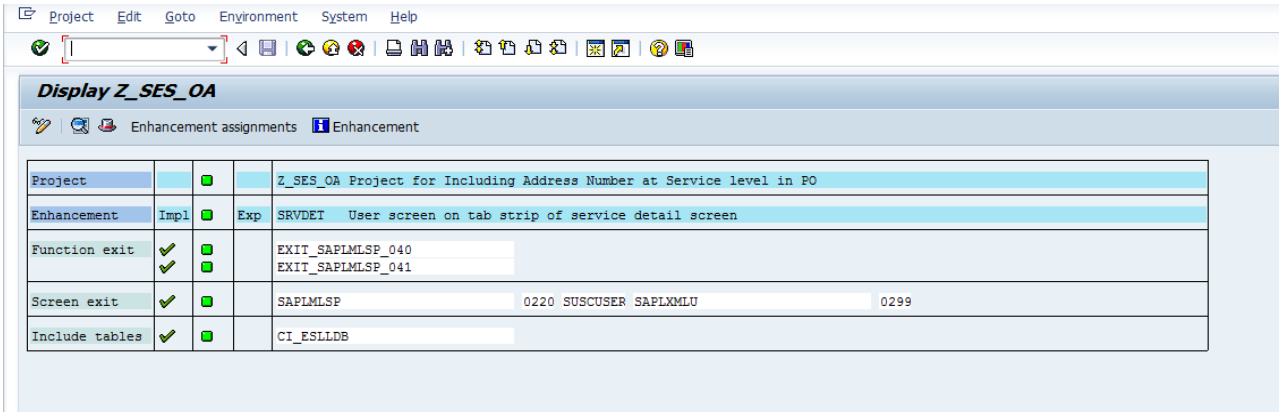
1. Add the new address number field ZZADRNR to the append structure CI_ESLLDB that is part of ESLL table (Example shown below for reference)

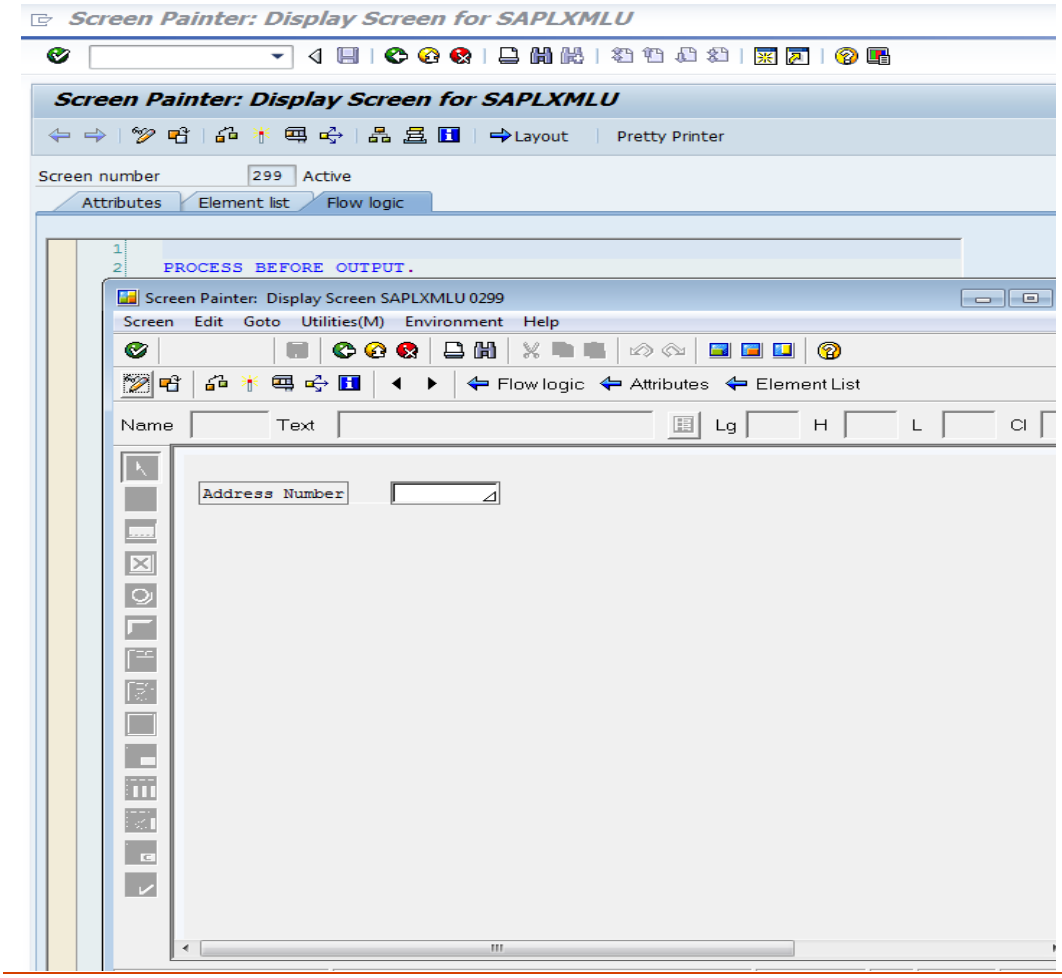


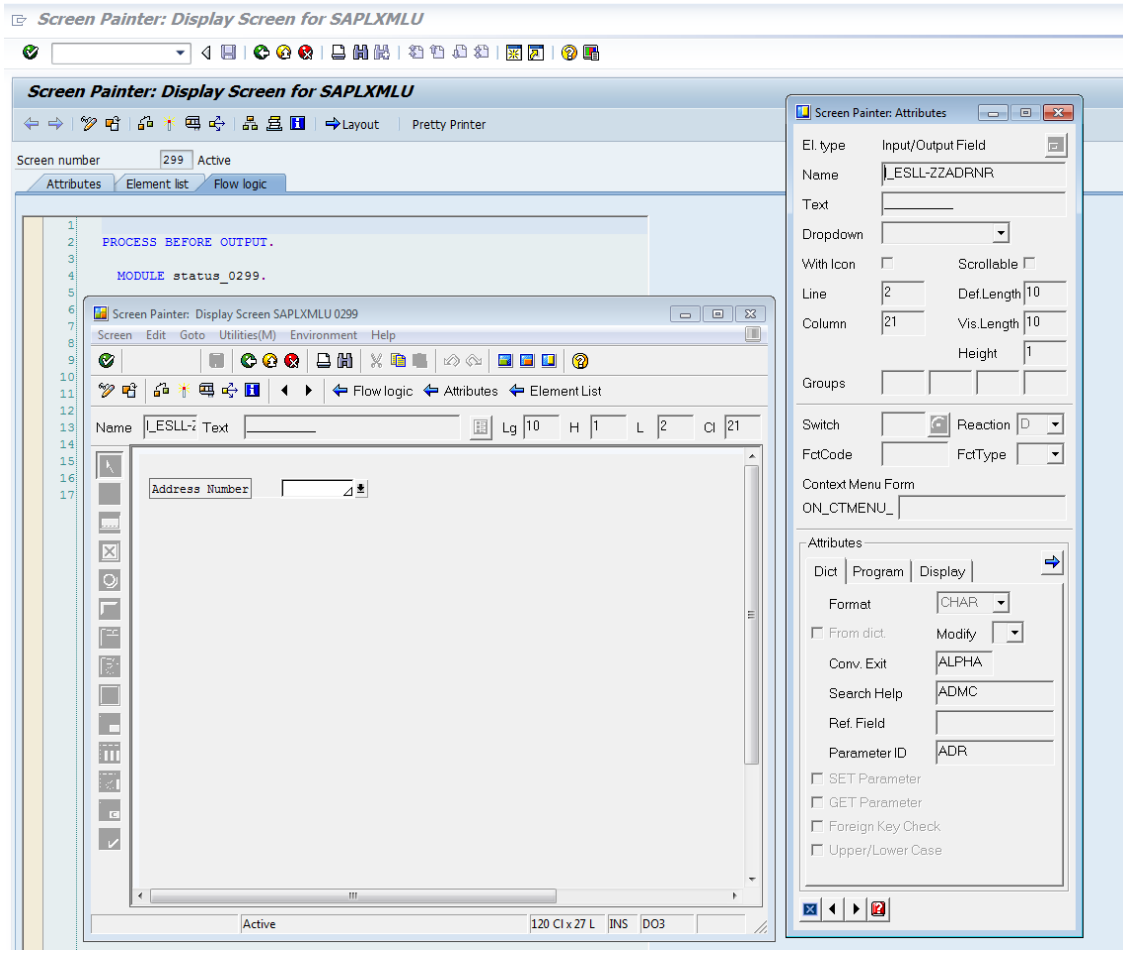
2. Create a new Project in CMOD and include the enhancement SRVDET for getting this to screen at Service level.



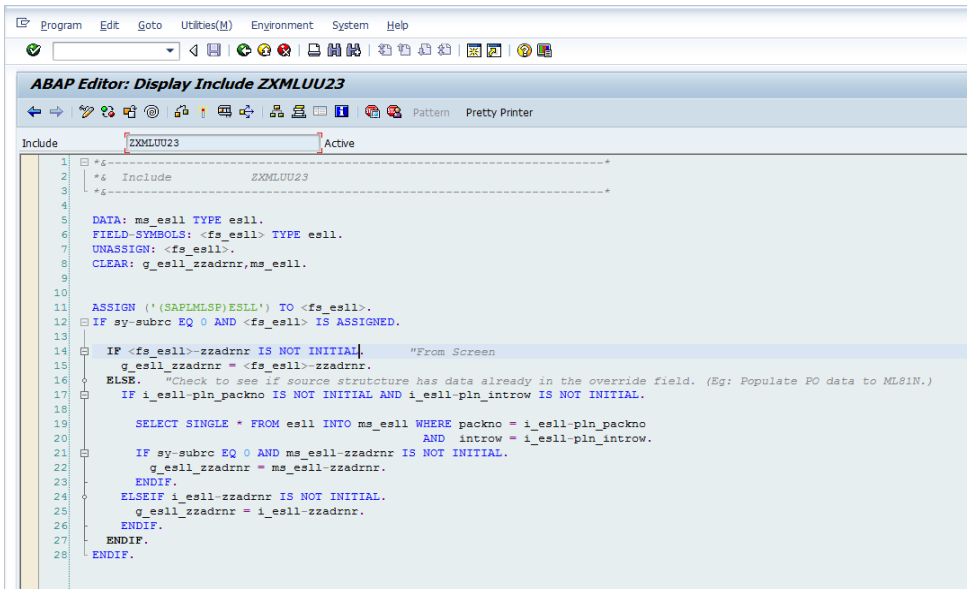
3. Implement the following exits within the enhancement. This includes the screen changes to add the custom field for override address. Sample code changes are also provided below to show how this works as a reference. This can be implemented in different ways too according to customer needs.



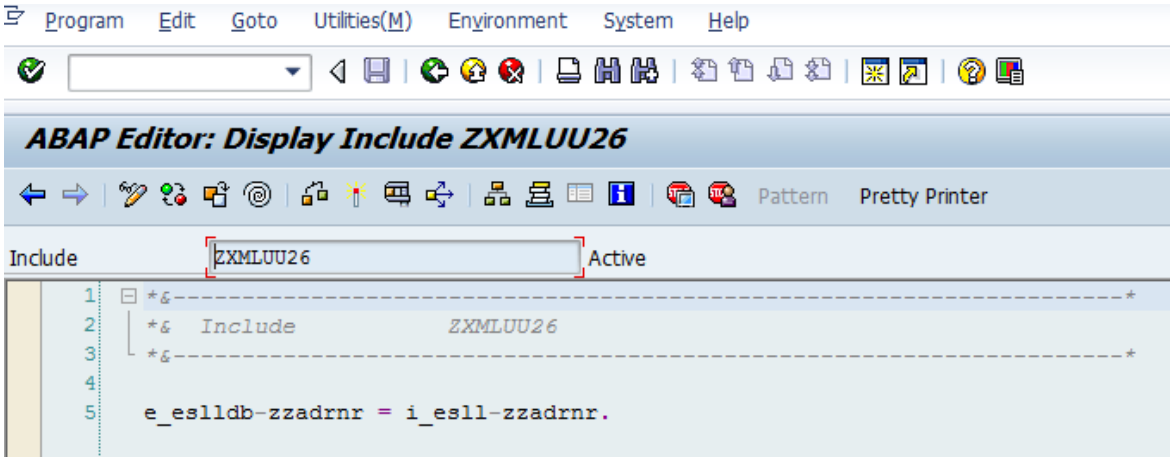




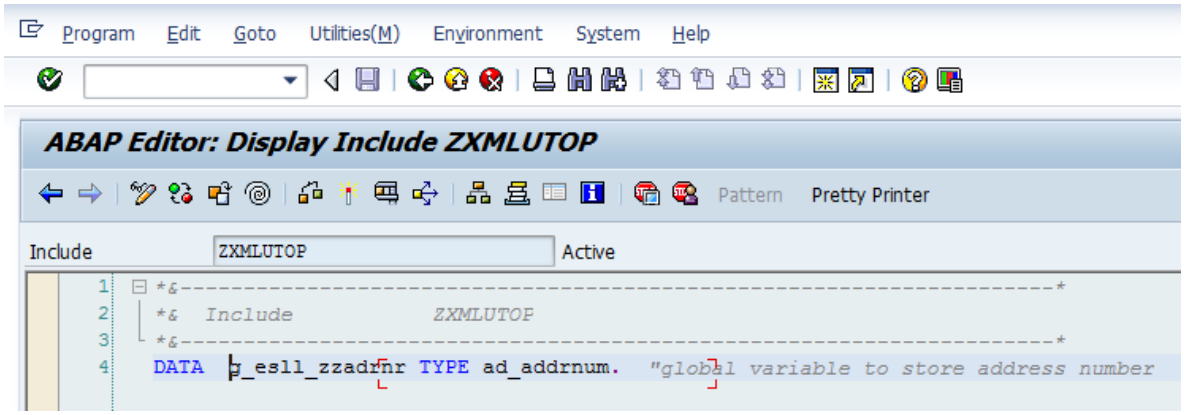
Sample code inside EXIT_SAPLMLSP_040 in include ZXMLUU23



Sample code inside EXIT_SAPLMLSP_041 on include ZXMLUU26

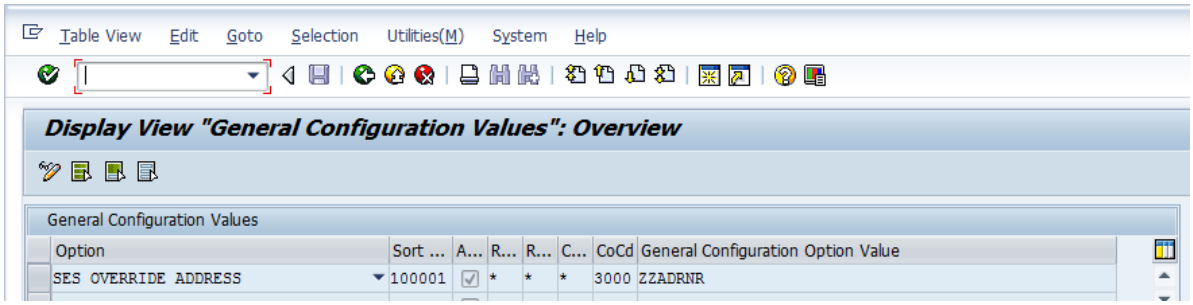


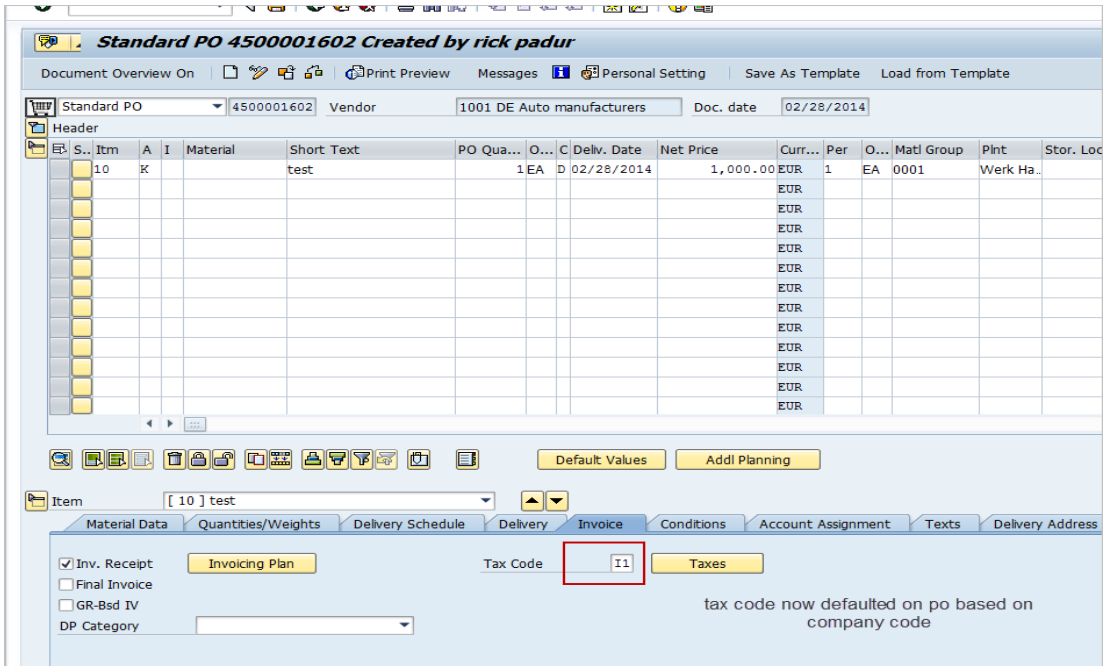
Code inside include ZXMLUTOP for declaring the global variable.



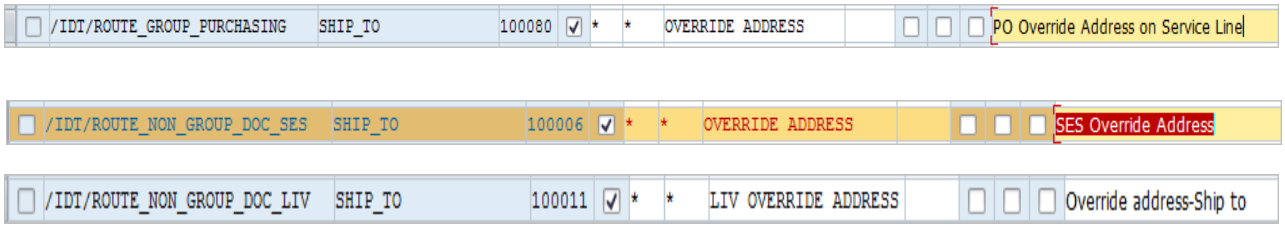
4. Create a new tab at the Service Level Detail screen (E.g.: Tab Named Other) and add the address number to the screen.

5. An entry for the new override address field for SES also needs to be added to the /IDT/D_GENERAL table as shown below.





6. The override address can be used as part of the mapping as shown below in the example.



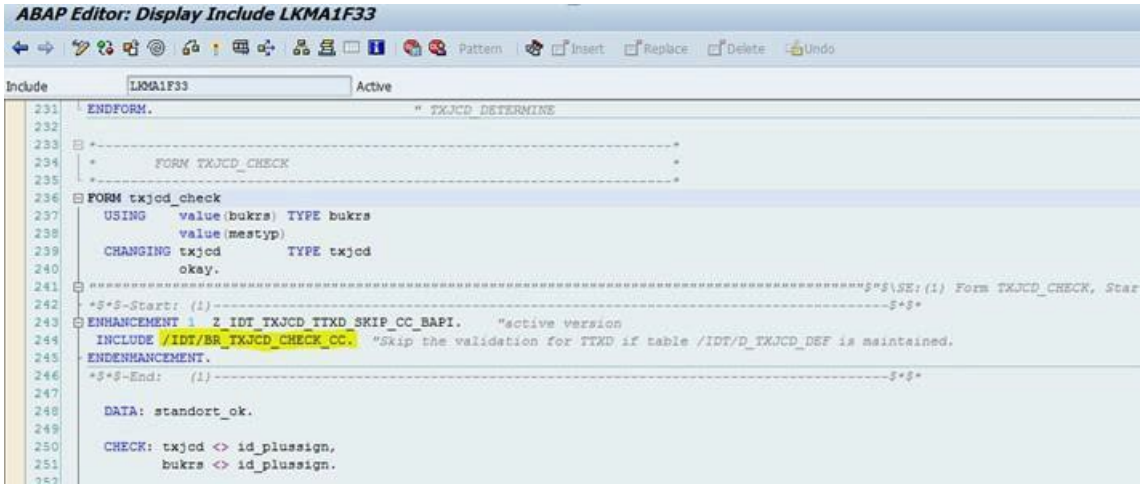
There is standard IDT code inside the class /IDT/SET_ADDRESS_OVERRIDE_ADDR within the methods SET_ADDRESS_ITM_PURCHASING and SET_ADDRESS_ITM_SES to pass this to the used for address in tax calculations.

BRAZIL: TAX JURISDICTION CODE FOR COST CENTER CREATED THROUGH BAPI

This is an optional enhancement point that a user can elect to implement to insert functionality to update the tax jurisdiction code for cost center addresses when cost centers are created using this BAPI. Manual entry of cost centers for Brazil have already been addressed with an include statement as part of the jurisdiction code usage for the Nota Fiscal, however if user elects to use this BAPI the current modification is not called and needs to also be inserted within this BAPI for use in batch entry. See notes below on this optional include:

A separate implicit enhancement is required in the include LKMA1F33 to handle the BAPI call K_COSTCTR_BAPI_CREATEMULTIPLE.

This looks at TTXD table for an entry and if it doesn't find one it will look at /IDT/D_TXJCD_DEF table. If an entry is available, then it skips the TXJCD validation.

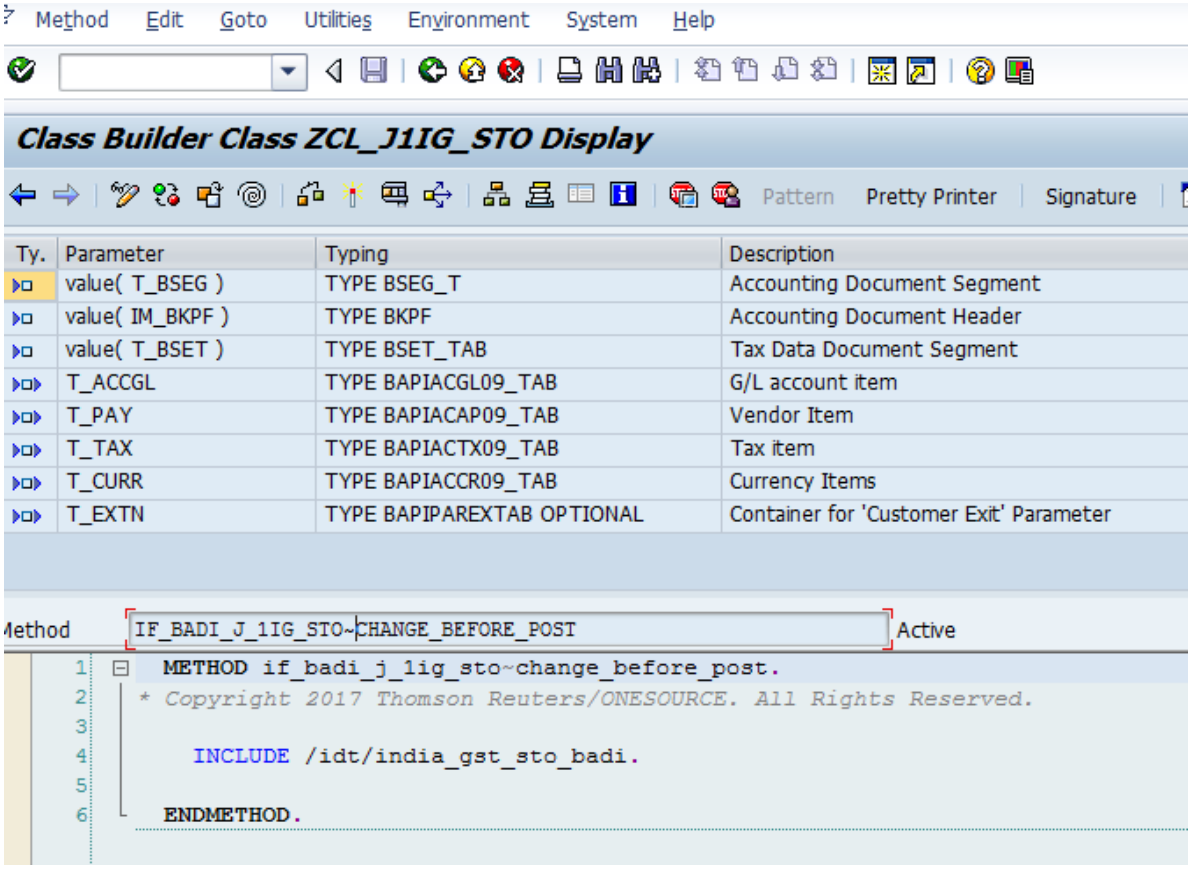


INCLUDE /IDT/BR_TXJCD_CHECK_CC.

INDIA OPTIONAL HOOKS/INCLUDE STATEMENTS FOR TRANSACTION J_1IG_INV

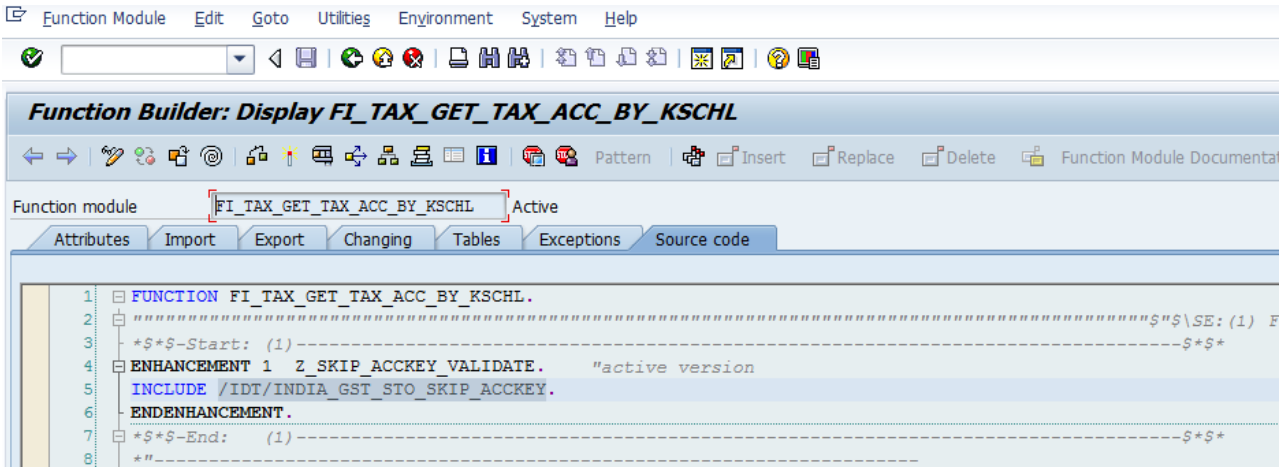
The following hook information must be added to your system if you have elected to utilize transaction J_1IG_INV for processing of Stock Transport Orders for India. This transaction scenario is further described in the India section of the Special Function Configuration Guide.

Hook 1 – Include /IDT/INDIA_GST_STO_BADI needs to be added in the method CHANGE_BEFORE_POST as shown in the screen shot below.



Then the BAPI BAPI_ACC_DOCUMENT_POST gets called immediately after this. We need to create an implicit enhancement as BAPI does a validation for Account Key.

Hook 2- The include program /IDT/INDIA_GST_STO_SKIP_ACCKEY needs to be added at the start of the function module FI_TAX_GET_TAX_ACC_BY_KSCHL to skip this validation. This is required only for India GST STO.



BADI ENHANCEMENTS FOR SETTLEMENT MANAGEMENT

Note: The BADI List mentioned for Settlement Management are valid from S/4 HANA 1809 onwards.

/IDT/BADI_WLF_SET_ROUTE_NAME

The purpose of this BADI is to determine the Settlement Route based on customer criteria.

The method SET_ROUTE_NAME of the BADI is provided to have custom logic which will override the default logic provided in the fallback class method.

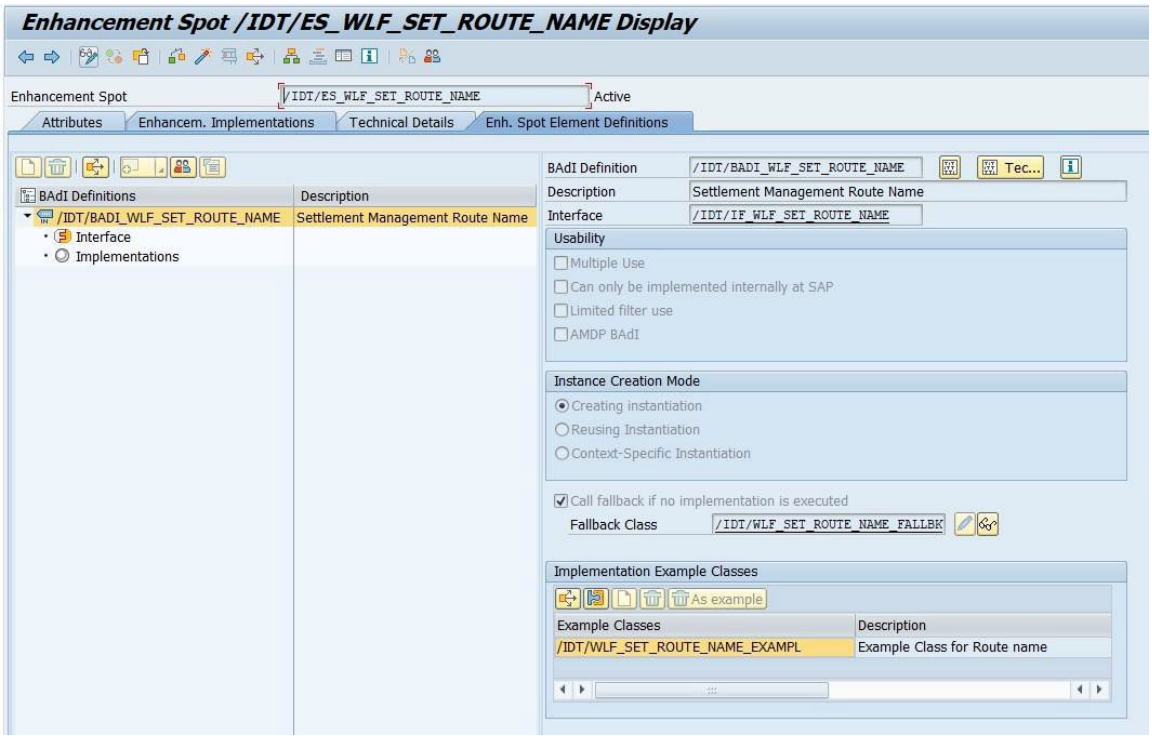
This BADI has a fallback class which will be executed if there is no implementation done from the customer. The default logic in the fallback class picks the route based on entries maintained using Transaction code

/IDT/SELECT_ROUTE_V for either of the two valid routes /IDT/ROUTE_GROUP_BILLING_SM_SD and

/IDT/ROUTE_GROUP_BILLING_SM_MM.

This logic can be overridden by creating a new implementation of this BADI in the customer's namespace.

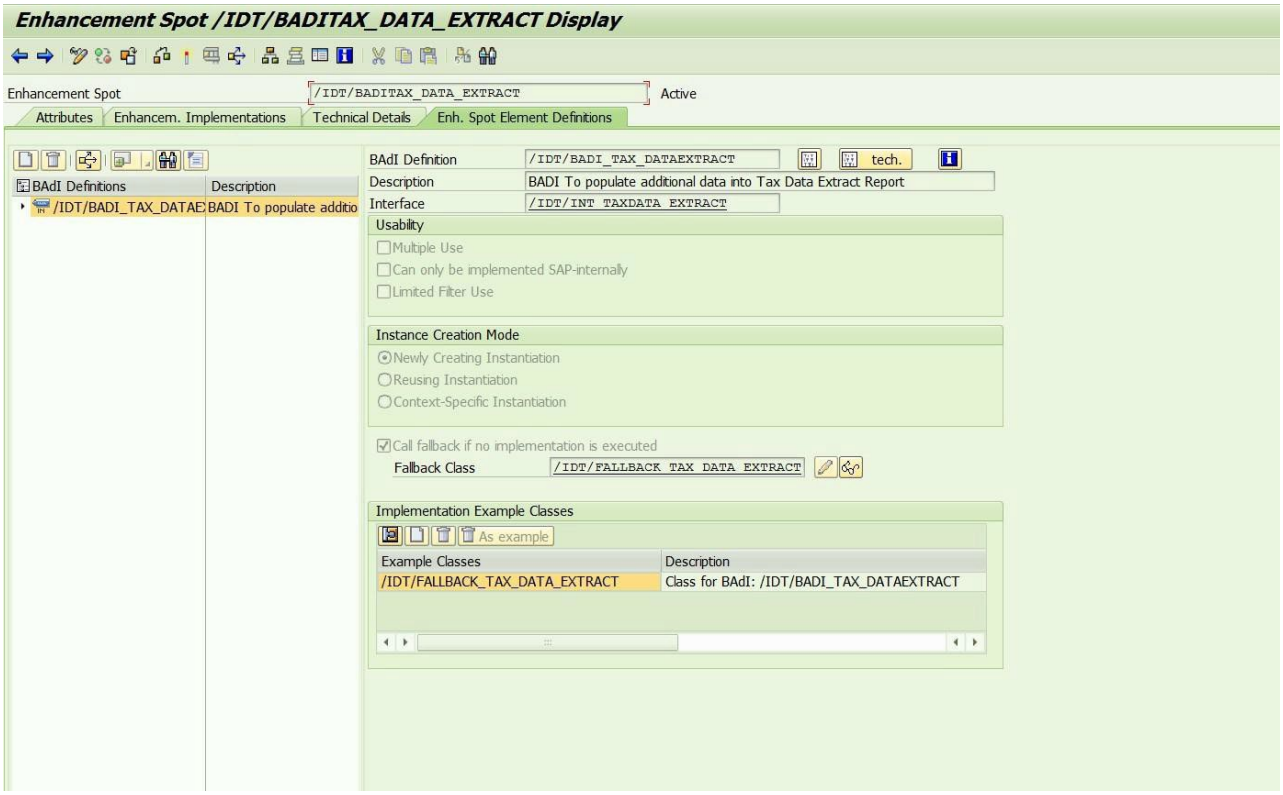
Example Class /IDT/WLF_SET_ROUTE_NAME_EXAMPL has commented code as an example that shows how to return the Settlement Management route names.



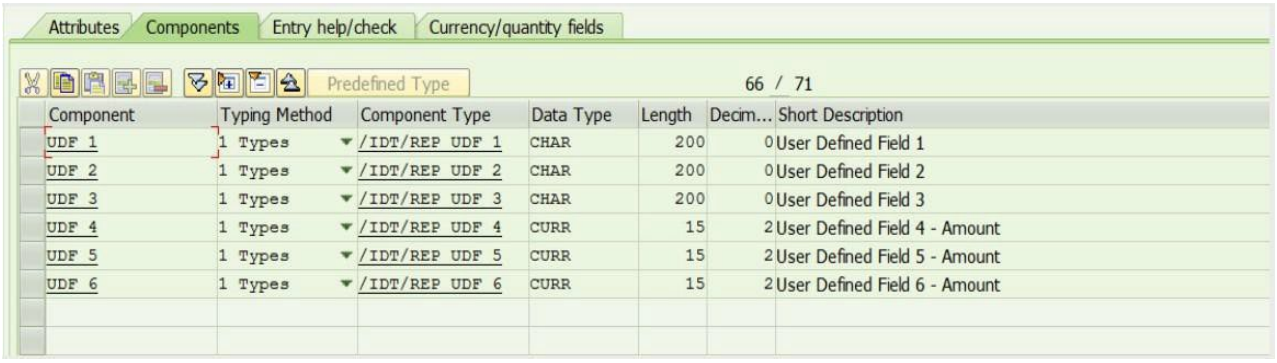
TAX DATA DOWNLOAD REPORT: OPTIONAL BADI FOR ADDING SELECTION FIELDS TO LAYOUT

A tax data download report is provided to users for analysis and download of tax data from their transactions. The report selects transaction data from the Tax Data Table. Users can also download the selected data for use in downstream compliance reporting needs. The list of data fields available in the report has been improved giving a larger list of fields from the tables. If the user has a need for an additional data element that is not listed in the field selection list, the user can write a program using the provided BADI to add the needed information to the report. This is an optional BADI and is not required for installation or running of the download report. It is used to add or modify data for the report layout.

/IDT/BADI_TAX_DATAEXTRACT



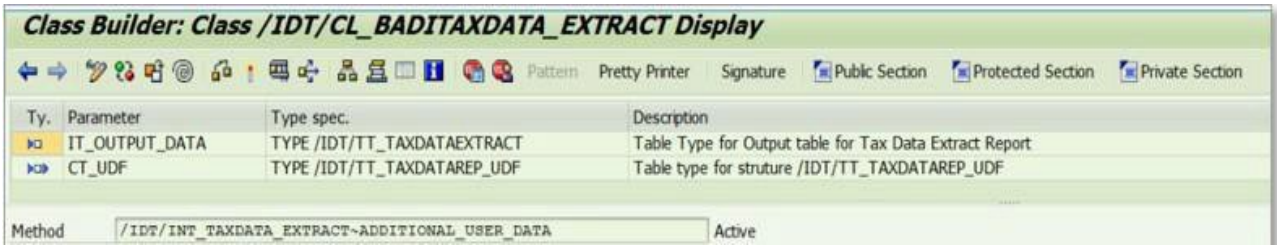
The BADI takes advantage of 6 fields that the user can use to program added output for the report. The fields and their layout are shown below. There are three for 200-character based fields and three for 15-character amount/currency-based fields.



Component	Typing Method	Component Type	Data Type	Length	Decim...	Short Description
UDF_1	1 Types	/IDT/REP_UDF_1	CHAR	200	0	User Defined Field 1
UDF_2	1 Types	/IDT/REP_UDF_2	CHAR	200	0	User Defined Field 2
UDF_3	1 Types	/IDT/REP_UDF_3	CHAR	200	0	User Defined Field 3
UDF_4	1 Types	/IDT/REP_UDF_4	CURR	15	2	User Defined Field 4 - Amount
UDF_5	1 Types	/IDT/REP_UDF_5	CURR	15	2	User Defined Field 5 - Amount
UDF_6	1 Types	/IDT/REP_UDF_6	CURR	15	2	User Defined Field 6 - Amount

In the below sample code we have taken the UDF4 field and within the BADI written code to reduce the multiple display of the Gross amount field to only display the gross amount once per line-item for each line-item on the document so that the gross amount is not overstated for some reporting needs. This is just an example of one of the many options that you can use this feature within the report layout.

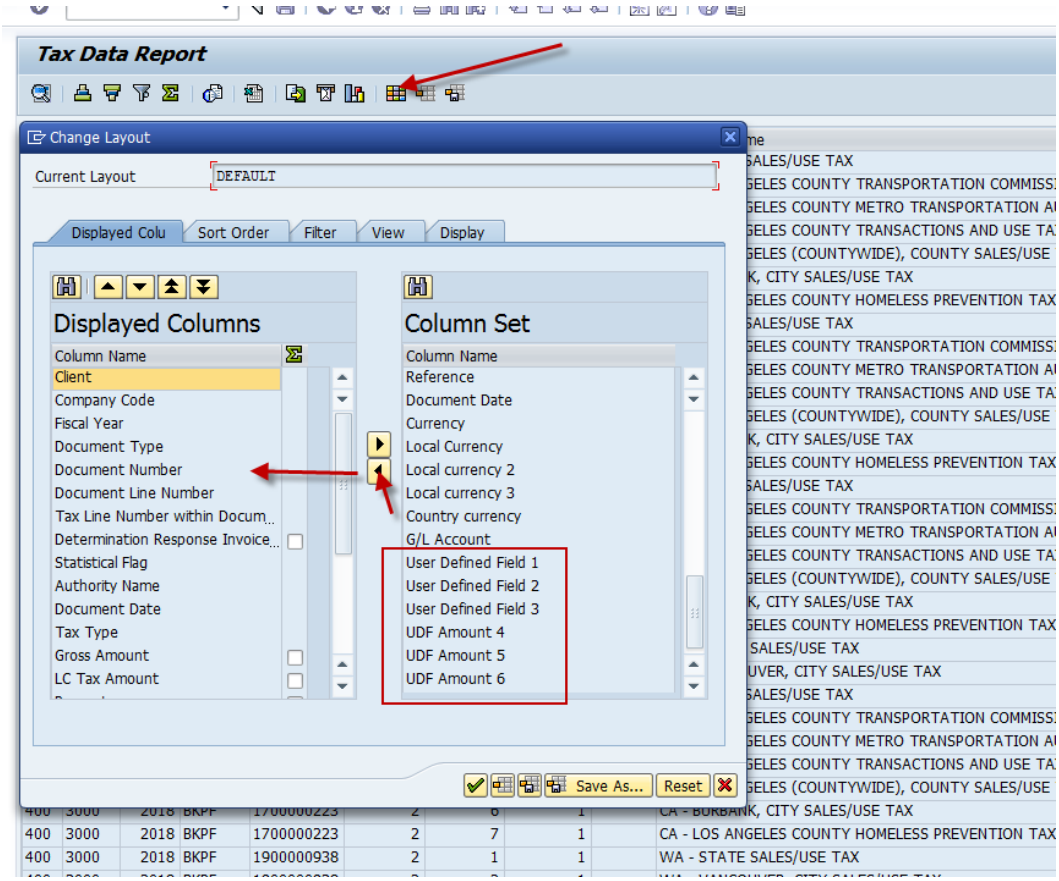
Note that using this BADI does not create this field within the Tax_Data Table but instead saves this temporarily in memory for the display of the report. Another option for users is to add additional fields to the Tax Data Table directly so they can be added to the selection options for the report.



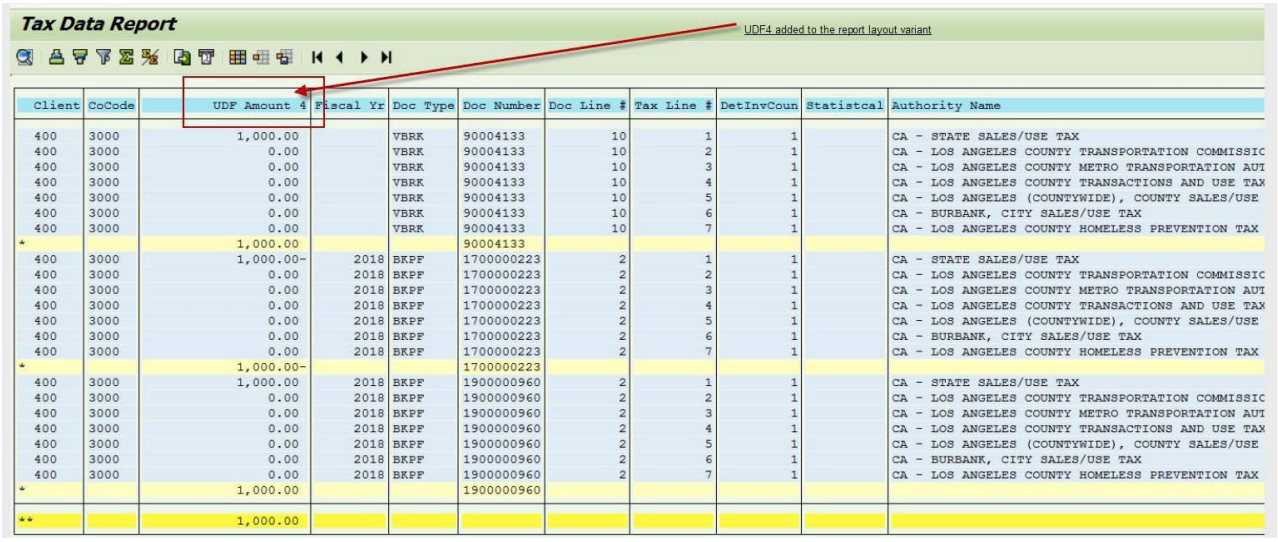

```
METHOD /idt/int_taxdata_extract~additional_user_data.
*This code snippet is for testing the implementation of BADI for Tax Data Report.
*Code to be excluded from Product Release Cycle.

  " Internal Table/Work Area Structure Declarations
  DATA : it_output_data_copy  TYPE         /idt/tt_taxdataextract,
         ms_output_data       LIKE LINE OF it_output_data.
  DATA   ms_udf               LIKE LINE OF ct_udf.

  " Make a copy of the final output entries.
  it_output_data_copy = it_output_data.

  " Delete adjacent duplicates from this set of entries by comparing the document company code document line numbers.
  DELETE ADJACENT DUPLICATES FROM it_output_data_copy COMPARING company_code document doc_line_number.

  " Build CT_UDF , internal table with User Defined Fields.
  LOOP AT it_output_data_copy INTO ms_output_data.

    ms_udf-belnr           = ms_output_data-document.
    ms_udf-bukrs           = ms_output_data-company_code.
    ms_udf-gjahr           = ms_output_data-fiscal_year.
    ms_udf-doc_line_number = ms_output_data-doc_line_number.
    ms_udf-doc_tax_number  = ms_output_data-doc_tax_number.
    ms_udf-document_type   = ms_output_data-document_type.
    ms_udf-inv_count       = ms_output_data-inv_count.
    ms_udf-statistical     = ms_output_data-statistical.

    ms_udf-udf_4           = ms_output_data-gross_amount.  " Move gross_amount to User Defined Field for Amount (UDF_4) field.

    APPEND ms_udf TO ct_udf.
    CLEAR:ms_output_data,ms_udf.

  ENDLOOP.

ENDMETHOD.
```

```
METHOD /idt/int_taxdata_extract~additional_user_data.
       *This code snippet is for testing the implementation of BADI for Tax Data
Report.
       *Code to be excluded from Product Release Cycle.

       " Internal Table/Work Area Structure Declarations
       DATA : it_output_data_copy TYPE          /idt/tt_taxdataextract,
       ms_output_data       LIKE LINE OF it_output_data.
       DATA    ms_udf               LIKE LINE OF ct_udf.

       " Make a copy of the final output entries.
       it_output_data_copy = it_output_data.

       " Delete adjacent duplicates from this set of entries by comparing the
document company code document line numbers.
       DELETE ADJACENT DUPLICATES FROM it_output_data_copy COMPARING company_code
document doc_line_number.

       " Build CT_UDF , internal table with User Defined Fields.
       LOOP AT it_output_data_copy INTO ms_output_data.

       ms_udf-belnr            = ms_output_data-document.
       ms_udf-bukrs            = ms_output_data-company_code.
       ms_udf-gjahr            = ms_output_data-fiscal_year.
       ms_udf-doc_line_number  = ms_output_data-doc_line_number.
       ms_udf-doc_tax_number   = ms_output_data-doc_tax_number.
       ms_udf-document_type    = ms_output_data-document_type.
       ms_udf-inv_count        = ms_output_data-inv_count.
       ms_udf-statistical      = ms_output_data-statistical.
       ms_udf-udf_4            = ms_output_data-gross_amount.  " Move gross_amount
to User Defined Field for Amount (UDF_4) field.

       APPEND ms_udf TO ct_udf.
       CLEAR:ms_output_data,ms_udf.

       ENDLOOP.

       ENDMETHOD.
```

Once this code is added to the BADI then the UDF4 field can then be selected in the list of fields for the report layout and used in the report.



Above screen showing the selection of fields for the layout and the 6 custom fields from the BADI



Client	CoCode	UDF Amount 4	Fiscal Yr	Doc Type	Doc Number	Doc Line #	Tax Line #	DetInvCoun	Statistcal	Authority Name
400	3000	1,000.00		VBRK	90004133	10	1	1		CA - STATE SALES/USE TAX
400	3000	0.00		VBRK	90004133	10	2	1		CA - LOS ANGELES COUNTY TRANSPORTATION COMMISSIC
400	3000	0.00		VBRK	90004133	10	3	1		CA - LOS ANGELES COUNTY METRO TRANSPORTATION AUT
400	3000	0.00		VBRK	90004133	10	4	1		CA - LOS ANGELES COUNTY TRANSACTIONS AND USE TAX
400	3000	0.00		VBRK	90004133	10	5	1		CA - LOS ANGELES (COUNTYWIDE), COUNTY SALES/USE
400	3000	0.00		VBRK	90004133	10	6	1		CA - BURBANK, CITY SALES/USE TAX
400	3000	0.00		VBRK	90004133	10	7	1		CA - LOS ANGELES COUNTY HOMELESS PREVENTION TAX
*		1,000.00			90004133					
400	3000	1,000.00-	2018	BKPF	1700000223	2	1	1		CA - STATE SALES/USE TAX
400	3000	0.00	2018	BKPF	1700000223	2	2	1		CA - LOS ANGELES COUNTY TRANSPORTATION COMMISSIC
400	3000	0.00	2018	BKPF	1700000223	2	3	1		CA - LOS ANGELES COUNTY METRO TRANSPORTATION AUT
400	3000	0.00	2018	BKPF	1700000223	2	4	1		CA - LOS ANGELES COUNTY TRANSACTIONS AND USE TAX
400	3000	0.00	2018	BKPF	1700000223	2	5	1		CA - LOS ANGELES (COUNTYWIDE), COUNTY SALES/USE
400	3000	0.00	2018	BKPF	1700000223	2	6	1		CA - BURBANK, CITY SALES/USE TAX
400	3000	0.00	2018	BKPF	1700000223	2	7	1		CA - LOS ANGELES COUNTY HOMELESS PREVENTION TAX
*		1,000.00-			1700000223					
400	3000	1,000.00	2018	BKPF	1900000960	2	1	1		CA - STATE SALES/USE TAX
400	3000	0.00	2018	BKPF	1900000960	2	2	1		CA - LOS ANGELES COUNTY TRANSPORTATION COMMISSIC
400	3000	0.00	2018	BKPF	1900000960	2	3	1		CA - LOS ANGELES COUNTY METRO TRANSPORTATION AUT
400	3000	0.00	2018	BKPF	1900000960	2	4	1		CA - LOS ANGELES COUNTY TRANSACTIONS AND USE TAX
400	3000	0.00	2018	BKPF	1900000960	2	5	1		CA - LOS ANGELES (COUNTYWIDE), COUNTY SALES/USE
400	3000	0.00	2018	BKPF	1900000960	2	6	1		CA - BURBANK, CITY SALES/USE TAX
400	3000	0.00	2018	BKPF	1900000960	2	7	1		CA - LOS ANGELES COUNTY HOMELESS PREVENTION TAX
*		1,000.00			1900000960					
**		1,000.00								

This display of the report shows the field that was created with the BADI added to the report layout This BADI is included in the Integration code transports but is blank for your use as you wish.

Note: Please see the User Guide section on the Tax Data Download Report for more information on the running of this report and special tips to consider when using the selection screen for filtering to exclude items from the report.

OPTIONAL BAPI FUNCTIONS SUPPORTED

There are several BAPIs that do not function when invoked properly when used in conjunction with this Integration for a few scenarios. These BAPIs work with the jurisdiction-based interface but do not work with this Integration and will need additional support for them to work for various functions. We have identified a list of them as currently requested from our customers and have developed the requested adaptations as listed below. This list may grow as future releases of the Integration address further requests.

For some of the BAPI issues, the problem identifies a need to add a tax call to the transaction prior to posting the transaction, and the standard BAPI call does not work as needed. Our Professional Services group has assisted customers to add a tax call within the BAPI. Such is the case with BAPI_ACC_DOCUMENT_POST. For this and any scenario where the tax data is already assumed to be present in the data table being used, we have elected to not include a tax calculation modification to the standard BAPI, and have elected to provide a separate tax calculation function module that must be run by the user prior to invoking the SAP standard post BAPI. By making this a separate utility in our name space the standard SAP BAPI is left unchanged, and the separate calculation process avoids duplicity of code. The function module for the tax call and the list of supported BAPIs are listed below.

Several other BAPIs have been brought up as possibly needing support due to unusual “corner cases” or select and specific country requirements. These were listed here as well and might involve a special circumstance where assistance was needed by our support team.

BAPI_ACC_DOCUMENT _CHECK

This BAPI is used to check the document and make sure that it has what it needs to post to the G/L. It should not need any further modification for it to work with our Integration

BAPI_ACC_DOCCUMENT_POST

This BAPI for posting FI accounting documents does make a call but also has a complex logic that checks certain aspects and thus causes some needed modification for it to work completely with our Integration. Our Professional Services group has assisted some customers with a long list of needed changes and adjustments. The Engineering team has created a different option to create a separate tax calculation utility to avoid some of the complexities that are required to use the standard tax call within the BAPI. The utility can be called separately outside of the BAPI or within the BAPI with the use of an include statement.

NEW UTILITY FOR CALCULATION OF TAX

The utility is /IDT/TAX_CALC_UTILITY_BAPI

For some application needs, the user may require the document that is to be posted to make a tax call and calculate the taxes prior to the BAPI posting the document. Our Professional services team has been able to assist many customers with this process with the addition of several program modification and user exits within the BAPI itself. In our attempt to productize this need we have elected to make this process a separate utility that can be run either prior to the BAPI being called or with an include statement within the BAPI so it can be utilized and calculated within the BAPI. Using a separate utility along with the use of the configuration tables to provide a flexible model that can adapt for many separate uses for a variety of needs.

BAPI_INVOICERECEIPT_CHECK

This BAPI is used to check the document and make sure that it has what it needs to post to the G/L. It should not need any further modification for it to work with our Integration

BAPI_INVOICERECEIPT_POST

This BAPI also has a tax call within but does not have some of the internal checks that are causing issues with our Integration. It has been tested and works as is without the use of the utility. There may be some issues that depend on corner case scenarios so users are encouraged to contact Customer Support if you are encountering an issue with this BAPI so that we can investigate and possibly assist in fixing the issue.

BAPI_INCOMINGINVOICE_PARK

This BAPI is making a call for a tax calculation and has been a reported issue in the past. It is likely that the issues were related to other program fixes that were needed in the code regarding the processing of parked document. Since these issues were fixed in our prior release this BAPI has been tested and confirmed to be working for our scenarios.

BAPI_INCOMINGINVOICE_CREATE

The issues with this BAPI are unlikely to be able to be fixed as SAP has been touching and changing it with every recent EHP release at the time the issues were presented. The issues crop up depending on which version the customer is on and is not something that we can easily productize. Some of the issues were identified as:

- Support of unplanned delivery costs which have since been addressed by other changes to code,
- Usage with entry of credit memos
- Use of Open Text VIM add on as well as KOFAX
- Invoices with PO's having split account assignment lines
- Normal PO Invoices: - With PO Lines and GL Only Lines

Since first reported, many of these problems may have been fixed with further additions to other programs or modules. If you are encountering an issue with the BAPI please contact TR support service to investigate and possibly fix the issue.

BAPI_INCOMINGINVOICE_CREATE1

This BAPI has been tested in our system and does work and is making a tax call. Prior issue regarding this BAPI was the result of other code functionality that has since been fixed for unplanned delivery charges. If you are encountering an issue with this BAPI, please contact TR support service to investigate and possibly fix the issue.

BAPI K_COSTCTR_BAPI_CREATEMULTIPLE

This BAPI ran into an issue when trying to add a large group of cost centers to Brazil. Because the Brazil Enablement included special logic for including a jurisdiction code that is only used for population on the Nota Fiscal, this BAPI needs an optional include statement added to use it for entry of Brazil cost centers. See section above for notes on the Brazil optional user exit that will need to be added if you are creating Brazil cost centers.

BAPI_CUSTOMER_CONTRACT_CHANGE

This BAPI has been tested in our system and does work and is making a tax call. Prior issue regarding this BAPI was the result of other code functionality that has since been fixed. If you are encountering an issue with the BAPI, please contact TR support service to investigate and possibly fix the issue.

/ARBA/BAPI_INVOICE_CREATE

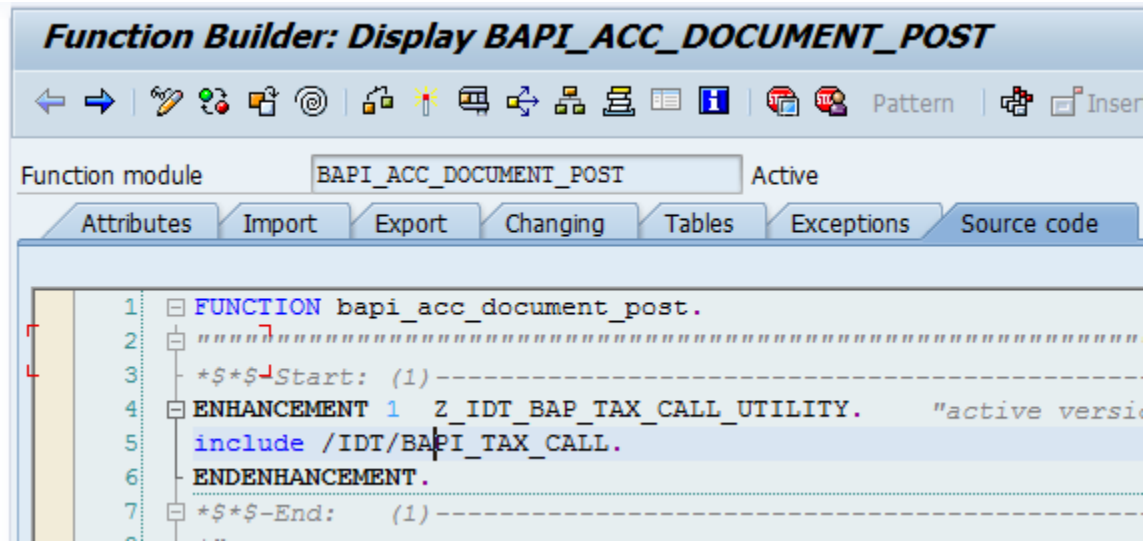
This is the BAPI used by the Ariba Integration to complete the G/L document post function from the OK to PAY files. Additional information on this BAPI can be found in the Ariba Integration Configuration Guide.

OPTIONAL BAPI INCLUDE TO BE ADDED

There is an optional user exit/ include statement that can be placed at the beginning of the BAPI program is the situation calls for the tax calculation utility to be called within the Post BAPI rather than a run of the utility prior to the BAPI being called. This is also outlined in the Installation and Programmers Guide as part of the install of the product. A copy of that include is also shown below:

BAPI_ACC_DOCUMENT_POST

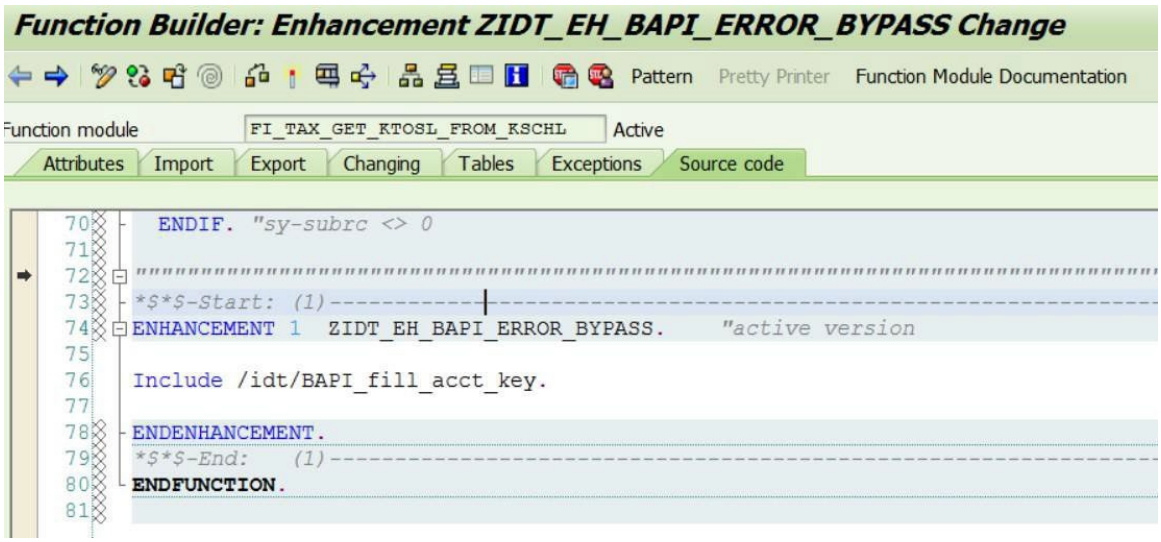
Include /IDT/BAPI_TAX_CALL



An Include is also required in the BAPI within this function to avoid an error with the assigned account key for the line-item. See Installation and Programmers Guide for further instructions.

Function module FI_TAX_GET_KTOSL_FROM_KSCHL

Include /IDT/BAPI_FILL_ACCT_KEY



Note: Special Note: As part of our testing processes we ran across several issues with the BAPI_ACC_DOCUMENT_POST that required the application of OSS notes. We recommend that you review and apply OSS notes 2074542, 1672710, 2424220, 2083799, 2117203, 1444708, 1631466,

1719134.

REFERENCES

This section lists all customer facing Journeys, Routes, Bases and Tables with an explanation of their purpose and use.

LIST OF JOURNEYS

Most Journeys are used in the field mapping process to assist in determining for which business process what fields should be used for taxability determination. Some Journeys are used internally for unique treatment of a process, like Freight and Plants Abroad for example, these are mostly likely not used in the field mapping.

JOURNEY	DESCRIPTION
/IDT/JOURNEY_HEADER_REQUEST	This Journey manages header data going from SAP to Determination. In field mappings this Journey passes data to link specific header SAP fields to corresponding invoice level XML elements of Determination.
/IDT/JOURNEY_HEADER_REQUEST_GM	This Journey manages header data going from SAP to Determination for the Goods Movement product transactions. In field mappings this Journey passes data to link specific header SAP fields to corresponding invoice level XML elements of Determination for materials movement transactions in MM.
/IDT/JOURNEY_HEADER_REQ_BR_GM	This Journey manages the header data going from SAP to Determination for the Brazil material transaction MBOA for receiving inbound transaction on Intra Co STO process.
/IDT/JOURNEY_GET_CONDITION_DTL	This journey is used internally and gets the relevant tax data from the KONV table and uses that for getting the taxes on the NF document when a PGI is done on the delivery document. Brazil functions only.
/IDT/JOURNEY_ITEM_REQUEST	This Journey manages header and item data going from SAP to Determination. In field mappings this Journey passes data to link specific header and item SAP fields to corresponding line level XML elements of Determination.

JOURNEY	DESCRIPTION
/IDT/JOURNEY_ITEM_REQ_BR_GM	This Journey manages the item data going from SAP to Determination for the Brazil material transaction MBOA for receiving inbound transaction on Intra Co STO process.
/IDT/JOURNEY_ITEM_REQUEST_GM	This Journey manages header and item data going from SAP to Determination for the Goods Movement product transactions. In field mappings this Journey passes data to link specific header and item SAP fields to corresponding line level XML elements of Determination for materials movement transactions in MM.
/IDT/JOURNEY_NG_ITEM_REQUEST	This Journey manages header and item data going from non- group transactions of SAP to Determination. In field mappings this Journey passes data to link specific header and item SAP fields to corresponding line level XML elements of Determination.
/IDT/JOURNEY_NG_ITEM_SERV_ENTR	This Journey manages item level data going from SAP to Determination for the specific data in Service Entry Sheets within MM PO process. In field mappings this Journey passes data to line specific line-item SAP fields to corresponding line level XML elements of Determination for Service Entry Sheet charges.
/IDT/JOURNEY_MULTI_ACC_SERVICE	This journey manages the multi account assignment as it is used on a service entry sheet line-item.
/IDT/JOURNEY_TAX_TAB_RESPONSE	This Journey manages data coming from Determination to SAP. It collects the tax calculation results from the XML and will use the configurable field mappings to link specific Determination fields to corresponding fields in table /IDT/D_ TAX_DATA for later use in downstream processes such as invoice printing and reporting.

JOURNEY	DESCRIPTION
/IDT/JOURNEY_STANDARD_RESPONSE	This Journey manages data coming from Determination to SAP. It collects the tax calculation results from the XML and distributes them in condition value formulas where needed. It will use the configurable field mappings to link specific Determination fields to corresponding SAP fields in table KONV. NOTE: This mapping is provided for backwards compatibility and should not be used.
/IDT/JOURNEY_AUDIT_UPD_DB_BILL	This Journey manages the update to the Determination audit database for Billing documents. At time of SAP finishing posting to the General Ledger account a call will be made to Determination for persisting the tax liability in audit. The process will use the latest calculation done and sets the IS_AUDITED flag to TRUE, the FISCAL_DATE to the posting date of the Billing document, the INVOICE_NUMBER and UNIQUE_INVOICE_NUMBER and the GROSS_AMOUNT in company code currency. It also may set the IS_CREDIT flag to TRUE/FALSE to negate amounts and make the audit reports add up correctly. NOTE: In some cases, a certain value isn't known till the document has been posted to the G/L, i.e. the legal document number required in some countries. This Journey can be used to write the value to audit, but it would not be recommended to use such a value for taxability rules as the value would not be available during calculations.
/IDT/JOURNEY_AUDIT_UPD_DB_SETL	This Journey manages the update to the Determination audit database for both Settlement Management sales and purchase billing documents. This journey is identical to /IDT/JOURNEY_AUDIT_UPD_DB_BILL for Settlement Management Audit documents.
/IDT/JOURNEY_AUDIT_UPD_DB_GL	This Journey manages the update to the Determination audit update database for G/L documents when there is no billing document. It is using the route /IDT/ROUTE_UPDATE_AUDIT_DB and is used to add information to the update to audit table and report for noting that status of an entry that may not have been posted yet to the audit database.

JOURNEY	DESCRIPTION
/IDT/JOURNEY_AUDIT_SAVE_FRM_GL	This Journey manages the update to the Determination audit database for G/L (LIV/FI) documents. At time of SAP finishing posting to the General Ledger account a call will be made to Determination for persisting the tax liability in audit. The process will use the latest calculation done and sets the IS_ AUDITED flag to TRUE, the FISCAL_DATE to the posting date of the G/L document, the INVOICE_NUMBER and UNIQUE_INVOICE_NUMBER and the GROSS_AMOUNT in company code currency. It also may set the IS_CREDIT flag to TRUE/FALSE to negate amounts and make the audit reports add up correctly. NOTE: In some cases, a certain value isn't known till the document has been posted to the G/L, i.e. the legal document number required in some countries. This Journey can be used to write the value to audit, but it would not be recommended to use such a value for taxability rules as the value would not be available during calculations.
/IDT/JOURNEY_AUDIT_SAVE_TAX_UP	This journey manages the update to the audit database and will be used for all manual tax scenarios (calculate tax = " ") and down payments. To update the audit database, this will use the last tax calculation make changes to a few fields like override amount and then send that to audit.
/IDT/JOURNEY_AUDIT_RESPONSE	This Journey manages the response of the update to the Audit Database. It determines if an update to the Audit Database is successful from the response message and passes a flag with that information to be stored in table /IDT/D_AUDIT_STA.
/IDT/JOURNEY_AUDIT_SAVE	This Journey saves the data of the last tax calculation call at the time of saving the invoice document in table /IDT/D_ AUDIT_REC for later use in the audit update call, cancellations, and other processes. This journey assumes calculate tax = X
/IDT/JOURNEY_BRAZIL_SD_ADJUST	This Journey adjusts the Brazil SD document and process when use to set the application area from TX to V. This was done to allow the system to utilize the standard Nota Fiscal mapping logic in SD for Brazil

JOURNEY	DESCRIPTION
/IDT/JOURNEY_CHECK_AUDIT_MESS	This Journey checks the audit message and adjusts the call to audit. It does a double check to make sure the call is for a final invoice.
/IDT/JOURNEY_GM_RESPONSE	This Journey manages the response of the update to the Audit Database for Goods Movement transactions in the Goods Movement product.
/IDT/JOURNEY_US_SPECIAL_LOGIC2	This Journey manages the AP logic for countries like US, and PR by switching the company role for the Vendor Charged Tax and offsetting the tax lines for Self-Accrual taxes. NOTE: Customers most likely will not use this in the field mappings; the default is delivered by Thomson Reuters.
/IDT/JOURNEY_FREIGHT	This Journey manages the Freight logic based on the configurable Freight condition sub-total. It will create a shadow line as a related line to the product line. See the Configuration Guide for more details. NOTE: This journey is not used in the field mappings.
/IDT/JOURNEY_FREIGHT_LIV	This Journey manages the Freight logic based on the configurable Freight condition sub-total specifically for functions that are different within the MIRO transaction. It will create a shadow line as a related line to the product line.
/IDT/JOURNEY_PLANTS_ABROAD	This Journey manages the logic for Plants Abroad based on the billing types maintained in table /IDT/D_PLNTS_ABD. For these billing types a Seller and Buyer call is made for the one SD Invoice. Billing type WIA has been added as a default. NOTE: This journey is not used in the field mappings.
/IDT/JOURNEY_FB05_RESPONSE	This Journey is to handle the complexity of the transactions that can have cash discounts.
/IDT/JOURNEY_NG_ITEM_FB05	This Journey manages header and item data going from non- group cash discount transactions of SAP to Determination. In field mappings this Journey passes data to link specific header and item SAP fields to corresponding line level XML elements of Determination.

JOURNEY	DESCRIPTION
/IDT/JOURNEY_FB05_COMPANY_ROLE	This Journey manages the company role for the FB05 transaction logic.
/IDT/JOURNEY_NG_ITEM_DOWN_PAYM	This Journey is to handle the complexity of the down payment transactions.
/IDT/JOURNEY_NG_ITEM_SERV_ENTR	This Journey is to handle the complexity of the transactions with service entry sheets.
/IDT/JOURNEY_SELF_ASSESSMENT	This Journey is used to handle the special fields needed in tandem with the table transaction /IDT/OFFSET_CONFIG to handle the offsetting side entry needed for self-assessment transactions.
/IDT/JOURNEY_UTILITY_RESPONSE	A Journey that allows a user-exit to read tax result data before the tax data table has been updated. This is not used in any standard function but can be used by a customer for custom user exit data that needs to be retrieved from the response prior to posting to the tax data table.
/IDT/JOURNEY_MULTI_ACC_ASSIGN	This journey is used to handle the complexity of the logic needed for multiple account assignment within a single PO line-item.
/IDT/JOURNEY_INDIA_STO_INB	This journey is used to handle the complexity of the logic needed for the J_1IG_INV inbound invoice for STO transaction.
/IDT/JOURNEY_ADDRESSES	This journey is used to handle the complexity of the logic needed for collecting address data.
/IDT/JOURNEY_ITEM_REQ_DT_RC	This Journey manages item data going from SAP to Determination for Deferred taxes Reverse charge scenarios.

LIST OF ROUTES

Routes can be split into two categories: Group and Non-Group. Group Routes are based on transactions which use pricing procedures (SD) or calculation schemas (PO), where Non-Group Routes are based on transactions which use tax procedures (LIV/FI). Routes can be used in the field mapping if desired.

ROUTE	DESCRIPTION
/IDT/ROUTE_GROUP_SALES	This Route handles the complexity of calculating tax unique to SD sales transactions.
/IDT/ROUTE_GROUP_BILLING_GEN	This Route handles the complexity of calculating tax unique to SD billing transactions.
/IDT/ROUTE_GROUP_DELIVERY	This Route handles the complexity of calculating tax unique to delivery transactions. This was added to support Brazil STO process for Nota Fiscal based on the delivery document.
/IDT/ROUTE_GROUP_BILLING_PA	This Route handles the complexity of calculating tax unique to SD Plants Abroad billing transactions. Table /IDT/D_PLNTS_ABD needs to be maintained with the relevant Billing Types for Plants Abroad, billing type WIA has been added as a default.
/IDT/ROUTE_GROUP_PURCHASING	This Route handles the complexity of calculating tax unique to purchasing transactions using a calculation schema.
/IDT/ROUTE_NON_GROUP_DOC_DNF	This Route handles the complexity of calculating tax unique to the delivery documents for Brazil Nota Fiscal.
/IDT/ROUTE_NON_GROUP_DOC_LIV	This Route handles the complexity of calculating tax unique to LIV transactions.
/IDT/ROUTE_NON_GROUP_DOC_PUR	This Route is in support of the TAXES button and copies the taxes from the conditions tab and replicates the data in the taxes button so that both conditions tab and taxes button are the same.
/IDT/ROUTE_NON_GROUP_DOC_AP	This Route handles the complexity of calculating tax unique to FI AP transactions.
/IDT/ROUTE_NON_GROUP_DOC_AR	This Route handles the complexity of calculating tax unique to FI AR transactions.
/IDT/ROUTE_NON_GROUP_DOC_FI	This Route handles the complexity of calculating tax unique to FI processes where there is no Vendor or Customer in the transaction.

ROUTE	DESCRIPTION
/IDT/ROUTE_NON_GROUP_DOC_GM_BR	This Route handles the complexity of calculation tax unique to Brazil STO material receiving transaction using MBOA transaction for inbound process Nota Fiscal data
/IDT/ROUTE_UPDATE_AUDIT_DB	This Route handles the complexity of updating the Audit Database. It initiates asynchronous update process (V2) after a G/L document posting has been successfully done in SAP and will trigger the audit call.
/IDT/ROUTE_NON_GROUP_DOC_DT	This Route is to handle the complexity of transactions with deferred tax for domestic scenarios.
/IDT/ROUTE_NON_GROUP_DOC_DT_RC	This Route is to handle the complexity of transactions with deferred tax for Reverse charge scenarios.
/IDT/ROUTE_NON_GROUP_DOC_FB5	This Route is to handle the complexity of the transactions that can have cash discounts.
/IDT/ROUTE_NON_GROUP_DOC_A_GL	This Route is to handle the logic for audit updated based on the amounts in the G/L document in SAP.
/IDT/ROUTE_NON_GROUP_DOC_DP	This Route is to handle the complexity of transaction with Down payments.
/IDT/ROUTE_NON_GROUP_DOC_SES	This Route is to handle the complexity of transactions with service entry sheets.
/IDT/ROUTE_NON_GROUP_DOC_INSTO	This Route is to handle the complexity of the J_1IG_INV transaction for intra-company STO process for the inbound invoice.
/IDT/ROUTE_GROUP_BILLING_SM_SD	This Route is to handle the calculation of taxes for Settlement Management Sales Rebates process.
/IDT/ROUTE_GROUP_BILLING_SM_MM	This Route is to handle the calculation of taxes for Settlement Management Purchase Rebates process.

LIST OF BASES

Bases represent a source or target in the field mapping of a tax request and response. They either represent an entity in SAP or a Determination XML structure such as Batch, Invoice, Line, or Tax. Some complex XML structures like User Attributes, Quantities, Registrations, Currency Conversions, etc. require special processing described at the end of this section. Not all the sources are available for all Journeys.

SOURCE BASES

Base	Description	Journey supported
SAP_HEADER	Fields from the following SAP document header tables: BKPF, EKKO,ESSR, KNA1, LFA1, T001, T001Z, T180, TVAK, TVAP, TVFK, TVKO, TVTA, VBAK, VBKD, VBUK, VBRK, KOMK, CALC_HDR*,J_ 1BBRANCH, J_1IMOVEND, J_ 1IMOCUST, J_1IMOCOMP, MKPF, MSEG, T156, DM07M, VM07M	/IDT/JOURNEY_HEADER_REQUEST /IDT/JOURNEY_HEADER_REQ_BR_GM /IDT/JOURNEY_FB05_COMPANY_ROLE
SAP_HEADER	BKPF (FI/LIV) VBRK (SD) T001 for both processes	/IDT/JOURNEY_AUDIT_UPD_DB_GL /IDT/JOURNEY_AUDIT_UPD_DB_BILL /IDT/JOURNEY_AUDIT_UPD_DB_SETL
SAP_HEADER	MKPF, BKPF, KOMK, KOMP, MSEG, EKKO, LFA1, CALC_HDR	/IDT/JOURNEY_HEADER_REQ_BR_GM
SAP_ITEM	Fields from the following SAP document item tables: BSEG, CSKS, DRSEG, EINA, EINE, EKKNU, EKPO, KOMP, KOMV_ INDEX, MAKT, MARA, MARC, MARD, MBEW, MT06E, MVKE, PRICE_ COND, T001W, T001Z ,TVAP, VBAP, VBAPF, VBRP, VBUP, CALC_ ITEM*,J_1BBRANCH, J_1IMOVEND, J_1IMOCUST, J_1IMOCOMP, MKPF, MSEG, T156, DM07M, VM07M Header fields: Any field listed in the SAP_HEADER section above, use prefix HDR-> when mapping a header table at line level	/IDT/JOURNEY_ITEM_REQUEST /IDT/JOURNEY_NG_ITEM_REQUEST /IDT/JOURNEY_ITEM_REQ_BR_GM /IDT/JOURNEY_NG_ITEM_DOWN_ PAYM /IDT/JOURNEY_NG_ITEM_FB05 /IDT/JOURNEY_NG_ITEM_SERV_ENTR

Base	Description	Journey supported
SAP_ITEM	BSE (FI/LIV) VBRP (SD) WBRP (SD,MM)	/IDT/JOURNEY_AUDIT_UPD_DB_GL /IDT/JOURNEY_AUDIT_UPD_DB_BILL /IDT/JOURNEY_AUDIT_UPD_DB_SETL
SAP_ITEM	MKPF, MSEG, EKPO, KOMP, LFA1, MBEW, J_1BBRANCH, LIKP	/IDT/JOURNEY_ITEM_REQ_BR_GM
SAP_FIELD	SYST GC_XS_FALSE – translates an SAP check box value to a XSD true/false value GC_VERSION – represents the XS	All
DET_TAX	All fields in the Tax level of the Determination response (OUTDATA) message	/IDT/JOURNEY_STANDARD_ RESPONSE /IDT/JOURNEY_TAX_TAB_RESPONSE /IDT/JOURNEY_FB05_RESPONSE
CONSTANT	Any constant value	All

TARGET BASES

Base	Description	Journey supported
DET_BATCH	All fields in Batch level of Determination request (INDATA) message	/IDT/JOURNEY_HEADER_REQUEST /IDT/JOURNEY_FB05_COMPANY_ ROLE /IDT/JOURNEY_HEADER_REQ_BR_ GM
DET_INVOICE	All fields in Invoice level of Determination request (INDATA) message	/IDT/JOURNEY_HEADER_REQUEST /IDT/JOURNEY_FB05_COMPANY_ ROLE /IDT/JOURNEY_HEADER_REQ_BR_ GM /IDT/JOURNEY_AUDIT_UPD_DB_BILL /IDT/JOURNEY_AUDIT_UPD_DB_GL

Base	Description	Journey supported
DET_LINE	All fields in the Item level of the Determination request (INDATA) message	/IDT/JOURNEY_ITEM_REQUEST /IDT/JOURNEY_NG_ITEM_REQUEST /IDT/JOURNEY_ITEM_REQ_BR_GM /IDT/JOURNEY_NG_ITEM_DOWN_ PAYM /IDT/JOURNEY_NG_ITEM_FB05 /IDT/JOURNEY_NG_ITEM_SERV_ ENTR
SAP_TAX_LINE	All fields in table /IDT/D_TAX_DATA, including custom appended fields	/IDT/JOURNEY_TAX_TAB_RESPONSE /IDT/JOURNEY_FB05_RESPONSE
SAP_TAX_LINE	All fields of the SAP pricing lines as defined in structure KOMV_INDEX NOTE: used for backwards compatibility only	/IDT/JOURNEY_STANDARD_ RESPONSE

SPECIAL PURPOSE OPERANDS TABLE

Operand	Description	Journey supported
HDR->	Pointer use in Item level request mapping to indicate that the field used is at header level, i.e. HDR->T001W-WERKS would indicate the plant from the header table to be mapped at the item level.	/IDT/JOURNEY_ITEM_REQUEST, /IDT/JOURNEY_NG_ITEM_REQUEST, /IDT/JOURNEY_FB05_RESPONSE
ANCESTOR->	Pointer used in response mapping to indicate that the field used is at a higher level in the structure, i.e. ANCESTOR->ANCESTOR->CALLING_ SYSTEM_NUMBER would be used to map from Invoice level field in the tax data level. NOTE: At this time, we don't support mapping from the BATCH level of the Determination XML, all fields are available on the INVOICE level too.	/IDT/JOURNEY_TAX_TAB_RESPONSE /IDT/JOURNEY_STANDARD_RESPONSE /IDT/JOURNEY_FB05_RESPONSE
ITEMS->	Pointer used to determine at the header level a field that is stored at the line-item level in order to pass that to the request.	/IDT/JOURNEY_HEADER_REQUEST

XSD TABLES

Table	Description	Journey supported
USER_ELEMENT	A custom field in the XSD comprised of the XML element USER_ELEMENT and field ATTRIBUTE#, where # is any number between 1-40 at Invoice or Item level. NOTE: Attributes 41-50 are reserved by Thomson Reuters.	/IDT/JOURNEY_HEADER_REQUEST, /IDT/JOURNEY_ITEM_REQUEST, /IDT/JOURNEY_NG_ITEM_REQUEST, /IDT/JOURNEY_STANDARD_ RESPONSE, /IDT/JOURNEY_TAX_ TAB_RESPONSE
REGISTRATIONS	A structure of registration numbers for different roles; SELLER_ROLE, BUYER_ ROLE, MIDDLEMAN_ROLE.	/IDT/JOURNEY_HEADER_ REQUEST, /IDT/JOURNEY_ITEM_ REQUEST, /IDT/JOURNEY_NG_ ITEM_REQUEST,
QUANTITIES	A complex structure to represent a list of quantities and unit of measures.	/IDT/JOURNEY_ITEM_REQUEST, /IDT/JOURNEY_NG_ITEM_ REQUEST,
CURRENCY_CONVERSION	Represents a result set of one or two currency conversion steps in the TAX block of the tax response.	/IDT/JOURNEY_STANDARD_ RESPONSE, /IDT/JOURNEY_TAX_TAB_ RESPONSE

SPECIAL PURPOSE OPERANDS IN TABLES

Operand	Description
NAME	A field value within a table like NAME=ATTRIBUTE1
ROW	A field value to point to a specific place in a table i.e. ROW=3 would point to the third row
ADD	An action, i.e. to add a row to the table
CREATE_IF_NOT_EXIST	A control flag to only add the mapping if none already exists

LIST OF DELIVERED TABLES

For reference below table lists all Thomson Reuters delivered SAP tables.

Configuration Tables

Most of the configuration tables are accessible via the User Menu provided with a few exceptions which are noted.

TABLE	DESCRIPTION
/IDT/C_DET_TAX_T	Determination Tax Code
/IDT/C_FI_CONTRL	FI Control Process: Custom
/IDT/C_LOG_CONFI	Log Configuration: Custom
/IDT/C_PROXIES	Configuration for Proxy Call
/IDT/C_ROLE_OVER	AP Company Role Override and Special Logic
/IDT/C_TAX_REL	Tax Code Relevancy
/IDT/C_TAX_TYPE	Determine condition types for taxes
/IDT/D_ADDR_SURC	Non partner function address sources
/IDT/D_ADDR_TYPE	Logical address types for ONESOURCE Tax Determination
/IDT/D_ADDRESSES	Match address sources to logical addresses types
/IDT/D_AUTO_JRNY	Automatically processed Journeys for each Route
/IDT/D_BASE_MAP	Base Maps
/IDT/D_CASH_DISC	Match SAP tax code to Determination Tax Code for Cash Discounts at time of payment calculations. Note this table is not in standard menu yet. Use SM30 transaction to maintain at this point.
/IDT/D_CG_ASSIGN	Country group assignment
/IDT/D_COUNTRY_G	Country groups
/IDT/D_DET_TAX_T	Determination Tax Type
/IDT/D_FI_CONTRL	FI process control configuration
/IDT/D_FIELD_MAP	Configurable field mappings
/IDT/D_GENERAL	General configuration option values table
/IDT/D_GRP_BUKRS	List company codes that should use non-grouped tax calc
/IDT/D_IS_EXEMPT	Configure exempt transactions

TABLE	DESCRIPTION
/IDT/D_LOG_CONFI	Log configuration
/IDT/D_NEG_TTYP	Tax scenarios to negate the tax values
/IDT/D_PART_SA	Partial Self-Assessment table
/IDT/D_PLNTS_ABD	Billing documents types that signify a plants abroad process
/IDT/D_PROXIES	Configuration for Proxy Call
/IDT/D_ROLE_OVER	AP company role override
/IDT/D_ROUTE_GRP	Route - Route Group index
/IDT/D_ROUTES	Configuration to switch on Routes
/IDT/D_SEL_ROUTE	Select route for SD and purchasing
/IDT/V_TAX_FILTE	Tax Filters
/IDT/D_TAX_REL	Tax Code Relevancy
/IDT/D_TAX_TYPE	Determine condition types for taxes
/IDT/D_TXJCD_DEF	Tax Jurisdiction Code Definitions for Country
/IDT/D_VERSION	IDT Integration Version
/IDT/D_WSDL	WSDL Data of the Proxy Structure
/IDT/GM_MVMTTYPE	Maintain Internal Goods Movement Types
/IDT/S_FI_CONTRL	FI Control Process: Standard
/IDT/S_LOG_CONFI	Log Configuration: Standard
/IDT/D_TX_SUM_GP	IDT Tax Summarization Table
/IDT/V_ADDR_SURC	SAP Non-Partner Function Address Sources: Custom
/IDT/V_ADDR_TYPE	Determination Address Types: Standard
/IDT/V_ADDRESSES	Address Mappings: Custom
/IDT/V_ADR_PRIVA	Address Mappings: Standard
/IDT/V_AJ_PRIVAT	Auto Processed Journeys for Route: Standard

TABLE	DESCRIPTION
/IDT/V_AUTO_JRNY	Auto Processed Journeys for Route: Custom
/IDT/V_BASE_MAP	Base Mappings: Standard
/IDT/V_CG_ASSIGN	Country Group Assign Standard
/IDT/V_CG_ASGN_N	Country Group Assignment View
/IDT/V_COUNTRY_G	Country Group Standard
/IDT/V_FM_PRIVAT	Field Mappings: Standard
/IDT/V_FIELD_MAP	Field Mappings: Custom
/IDT/V_GENERAL	General Configuration Values
/IDT/V_GM_MVTYPE	Goods Movement types
/IDT/V_NEG_TTYP	Negate Tax Types: Standard
/IDT/V_NEG_TTYPE	Negate Tax Types: Custom
/IDT/V_PLNTS_ABD	Plants Abroad Billing Type
/IDT/V_RG_PRIVAT	Route Groups: Standard
/IDT/V_ROUTE_GRP	Route Groups: Custom
/IDT/V_ROUTES	Configuration to Switch on Routes
/IDT/D_PRODUCTS	List of possible products
/IDT/V_SEL_ROUTE	Select Route for SD and Purchasing Configuration
/IDT/V_TAX_SUM_C	IDT Tax Summarization Configuration View
/IDT/V_VERSION	IDT Integration Version
/IDT/D_WS	WS Security Configuration for Proxy
/IDT/D_DYN_PROG	Dynamic Program Index
/IDT/D_ACT_CNTRL	Actual Data Pop-Up Control Configuration Tables
/IDT/D_DL_CONFIG	Developer Log Configuration: Custom
/IDT/D_STRUCTS	Data Dictionary objects for Journey source and target structure. This is a behind the scenes table that is not made available through the main menu and is pre-populated with the transport of the system configuration.

TABLE	DESCRIPTION
/IDT/D_SR_AD_MAP	Configurable Source address mappings
/IDT/D_PROXYGRPS	Used to establish the three options for a proxy group code, blank, CL, OP

Transaction Tables

Transaction tables can hold a considerable amount of data depending on your business processes and system configurations. You should monitor growth of these tables and manage them as part of your archiving and/or purging process.

ROUTE	DESCRIPTION
/IDT/D_AUDIT_REC	Record of last Determination calculation request messages for document for later use in the audit update call, cancellations, and other processes. Archiving should be tied to your document retention process.
/IDT/D_AUDIT_STA	Status of messages for audit DB. Archiving can be managed as part of month end close process.
/IDT/D_LOG	Log for tax calculations based on log configuration settings. Archiving can be managed as part of month end close process.
/IDT/D_DEV_LOG	Indirect tax interface developers log
/IDT/D_TAX_DATA	Table that holds additional Tax Data information for a given document in support of invoice printing and downstream processes. This table is tied to the transaction document and should be managed for archiving based on the leading documents archiving.
/IDT/D_INDIA_STO	Table holds additional tax data information from the outbound invoice on STO process and is temporarily stored and used for the inbound invoice using transaction code J_1IG_INV.

RESERVED ATTRIBUTES

Thomson Reuters reserved attributes 41-50 of the Invoice and Line fields for internal use. The following table is a list of the standard attributes that are already mapped. Customers can't make use of Attributes 41-50.

ATTRIBUTE	USED FOR
INVOICE.USER_ELEMENT.ATTRIBUTE41	TAX CATEGORY OF 0 OR 1
INVOICE.USER_ELEMENT.ATTRIBUTE50	PLANTS ABROAD - CONSTANT "PA"
LINE.USER_ELEMENT.ATTRIBUTE42	G/L ACCOUNT NUMBER
LINE.USER_ELEMENT.ATTRIBUTE43	COST CENTER
LINE.USER_ELEMENT.ATTRIBUTE44	CREDIT/DEBIT FLAG (S, H)/ GM - Used in Goods Movement Program Also used at invoice level to note the use of both I1 and U1 driver tax codes on a document for FB08 proper document reversals.

ATTRIBUTE	USED FOR
LINE.USER_ELEMENT.ATTRIBUTE45	ROUTE NAME
LINE.USER_ELEMENT.ATTRIBUTE46	SAP TAX CODE for Item and NG_Item Journey/ TAX CODE OF ORIGINAL DOCUMENT FOR CASH DISCOUNTS AT TIME OF PAYMENT CALCULATION
LINE.USER_ELEMENT.ATTRIBUTE47	US-specific configuration that applies to the line.
LINE.USER_ELEMENT.ATTRIBUTE49	Used for (Sequential Number of Account Assignment) for item request journey for SES - /IDT/JOURNEY_NG_ITEM_SERV_ENTR.
LINE.USER_ELEMENT.ATTRIBUTE50	DIVISION/ BUSINESS AREA (depending on module)

List of Transaction Codes

Below is a list of the transaction codes added to the system for the IDT integration tables and processes. This list can be used by security personnel in order to assign transactions to roles for security authorizations

TRANSACTION CODE	USED FOR
/IDT/SELECT_ROUTE_V	Select Route for SD and Purchasing View Only
/IDT/AUTO_JOURNEYS_V	Automatically Processed Journeys View Only
/IDT/ROUTE_GROUP_V	Route Groups View Only
/IDT/ADDRESS_TYPES_V	Address Types View Only
/IDT/ADDRESS_SOURC_V	Address Sources View Only
/IDT/LOG_CONFIG_V	Log Configuration View Only
/IDT/NEG_TAX_TYPE_V	Negate Direction of Tax Types View Only
/IDT/FI_CONTROL_V	FI Process Control Configuration View Only
/IDT/COUNTRY_G_V	Country Groups View
/IDT/CG_ASSIGN_V	Country Group Assignment View
/IDT/BASE_MAPPING_V	Base Mappings View Only
/IDT/FIELD_MAPPING_V	Field Mappings View Only
/IDT/ADDRESS_MAP_V	Address Mapping View Only
/IDT/PROXY_CONFIG	ONESOURCE Proxy Configuration
/IDT/LOG_NUMBR_RANGE	Log Number Range
/IDT/WS	Web Service Security Configuration

TRANSACTION CODE	USED FOR
/IDT/LOG_CONFIG	Configure Logs
/IDT/ROUTE_CONFIG	Route Configuration
/IDT/DETER_COND_TYPE	Determine Condition Type for Taxes
/IDT/ROUTE_GROUP	Route Groups
/IDT/NEG_TAX_TYPE	Negate Direction of Tax Types
/IDT/FI_CONTROL	FI Process Control Configuration
/IDT/FIELD_MAPPINGS	Field Mappings
/IDT/ADDRESS_MAPPING	Address Mapping
/IDT/COUNTRY_GROUPS	Country Groups
/IDT/CNTRY_GRP_ASSIG	Country Group Assignment
/IDT/TAX_SUM_CONFIG	Configuration for Tax Summarization
/IDT/EXEMPT_SETTINGS	Tax Exemption Settings
/IDT/GEN_CONFIG_VALS	General Configuration Values
/IDT/DET_TAX_CODE	SAP Tax Code/Det Tax Code Index
/IDT/CASH_DISCOUNT	Cash Discount/Det Tax Code Index
/IDT/US_LOGIC	US Specific Logic
/IDT/OFFSET_CONFIG	Offset tax line configuration
/IDT/PLANTS_ABROAD	Plants Abroad Billing Types
/IDT/TAX_CODE_REL	Tax Code Relevancy
/IDT/AUTO_JOURNEYS	Automatically Processed Journeys
/IDT/ADDRESS_SOURCES	Address Sources
/IDT/LOG	Log Reader
/IDT/LOG_ARCHIVE	Log Archival – Delete/Export to ZIP File (used for both foreground and background processing)
/IDT/AUDIT_DATABASE	Audit Database Transactions Update used for foreground processing
/IDT/RECON_EXTRACT	ERP Reconciliation Extract Report

TRANSACTION CODE	USED FOR
/IDT/TAX_DATA_RPT	Tax Data Report
/IDT/TAX_FILTERS	Tax Filters Configuration
/IDT/VERSION	Global Next Product Version
/IDT/WS	Web Service Security Configuration

Reconciliation Extract Program References

Reconciliation Extract Mapped from SAP for ERP Reconciliation Extract Report

This shows the mapping from SAP to ONESOURCE Indirect Tax Reconciliation Extract Output.

OUTPUT FIELD NAME	SAP FIELD NAME	TABLE-FIELDNAME
External Company ID	SAP company code	BKPF-BUKRS
Host System	SAP system name (e.g. CO3)	SY-SYSID
Calling System	SAP client number	SY-MANDT
Company Role	Buyer (B) / Seller (S)	
ERP Transaction ID	Reference key of SAP Accounting document	BKPF-AWKEY
Document Number	Accounting document number	FI: BKPF-BELNR LIV: RBKP-BELNR SD: VBRK-VBELN
Document Type	Accounting document type	BKPF-BLART
Document Description	Accounting document header text	BKPF-BKTXT
Document partner number	Customer number (AR transactions) /	BSEG-KUNNR / BSEG-LIFNR
Vendor number (AP transactions)	BSEG-KUNNR /	KNA1-NAME1 / LFA1-NAME1
ERP Period	Fiscal Period	BKPF-MONAT

OUTPUT FIELD NAME	SAP FIELD NAME	TABLE-FIELDNAME
Document Date	Document date in format : DD-MON-YYYY, E.g. 12-Aug-2012	BKPF-BLDAT
Fiscal Date	Posting Date Date Format : DD-MON-YYYY	BKPF-BUDAT
Gross Amount	Tax Base Amount in Local (Company code) Currency	BSET-HWBAS
Tax Amount	Tax Amount in Local (Company code) Currency	BSET-HWSTE
Document currency	Document currency key	BKPF-WAERS
Optional UDF 1	Custom field based on custom BAPI implementation	N/A
Optional UDF 2	Custom field based on custom BAPI implementation	N/A
Optional UDF 3	Custom field based on custom BAPI implementation	N/A
Optional UDF 4	Custom field based on custom BAPI implementation	N/A
Optional UDF 5	Custom field based on custom BAPI implementation	N/A

Extract Selection Screen Field Definition

The table below shows in detail each of the selection screens available at runtime of the Reconciliation Extract:

DESCRIPTION	FIELD NAME	TYPE	OPTIONAL/REQUIRED	DEFAULT VALUE
Company Code	BKPF-BUKRS	Parameter	R	N/A
Fiscal Year	BKPF-GJAHR	Parameter	R	N/A

DESCRIPTION	FIELD NAME	TYPE	OPTIONAL/REQUIRED	DEFAULT VALUE
Posting Date	BKPF-BUDAT	Select options	O	N/A
Posting Date	BKPF-MONAT	Select options	O	N/A

DESCRIPTION	FIELD NAME	TYPE	OPTIONAL/REQUIRED	DEFAULT VALUE
Tax Code	BSEG-MWSKZ	Select options	O	N/A
Select Zero/Exempt tax records	N/A	Check Box	O	X
Application / Local server path	N/A	Radio Buttons	O	Application server
Application Server	RLGRAP-FILENAME	Parameter	O	TVARV variable value
Presentation Server	RLGRAP-FILENAME	Parameter	O	User parameter value
Company Code Prepend	N/A	Parameter	O	N/A
Package Size	N/A	Parameter	R	10000

ONESOURCE Indirect Tax Transport Object for ERP Reconciliation Extract Report

This section lists the technical objects delivered with the Reconciliation Extract Report transport provided:

OBJECT NAME	OBJECT TYPE	DESCRIPTION OF OBJECT
/IDT/	Name space	Container for all objects delivered by Thomson Reuters
/IDT/RECON_EXTRACT	Package	Package containing all report development objects
/IDT/RECEXT	Transaction code	Transaction code to run the Reconciliation extract report

OBJECT NAME	OBJECT TYPE	DESCRIPTION OF OBJECT
/IDT/RECON_EXTRACT	Report program	Main program
/IDT/RECON_EXTRACT_TOP	Include	Include contains global types and data declaration
/IDT/RECON_EXTRACT_SEL	Include	Selection screen is defined under this include
/IDT/RECON_EXTRACT_FORM	Include	Include contains form routines
/IDT/BADIRECON_EXTRACT	BADI definition	BADI definition to add logic for 5 user defined fields
/IDT/EXTRACT_OUTPUT	Structure	Output file structure
/IDT/INT_RECON_EXTRACT	Interface	BADI Interface
/IDT/CL_RECON_EXTRACT	Class	Class implementing the BADI interface
/ IDT/EXTRACT_UDF	Structure	BADI return data structure
/ IDT/APP_SERVER_PATH	TVARV global variable	Variable in TVARV table to maintain Application server path
/IDT/LCLFILEPATH	Parameter ID	User parameter to maintain default Presentation server path

Enabling Custom Fields for ERP Reconciliation Extract Report

The file format of the Reconciliation Report allows for five (5) user defined fields (UDF) as pass through elements. Companies wishing to use these fields can use custom code to set a value from SAP and extract it into one of the UDF's. These values then will be imported into the Reporting reconciliation tables and show in the Reconciliation Report process.

A BADI /IDT/BADIRECON_EXTRACT has been provided as part of the SAP Reconciliation Report which can be implemented by the customers. The BADI method returns the 5 UDF's in the structure /IST/EXTRACT_UDF.

Reconciliation Extract Mapped from SAP for ERP Recon Extract New

This shows the mapping from SAP to ONESOURCE Indirect Tax Reconciliation Extract Output.

OUTPUT FIELD NAME	SAP FIELD NAME	TABLE FIELD NAME
External Company ID	SAP company code	BKPF-BUKRS
Host System	SAP system name (e.g. CO3)	SY-SYSID
Calling System Number	SAP client number	SY-MANDT
Merchant Role	Buyer (B) / Seller (S)	/IDT/D_TAX_DATA - COMPANY_ROLE
ERP Transaction ID	Reference key of SAP Accounting document	BKPF-AWKEY
Invoice Number	Accounting document number	FI: BKPF- BELNR LIV: RBKP- BELNR SD: VBRK-VBELN
Document Type	Accounting document type	BKPF-BLART
Document Description	Accounting document header text	BKPF-BKTXT
Document partner number	Customer number (AR transactions) / Vendor number (AP transactions)	BSEG-KUNNR / BSEG-LIFNR
Document Partner Name	Customer/Vendor Name	KNA1-NAME1 / LFA1- NAME1
ERP Period	Fiscal Period	BKPF-MONAT
Invoice Date	Document date in format: DD-MON- YYYY, e.g. 12-Aug-2012	BKPF- BLDAT
Fiscal Date	Posting Date Date Format: DD-MON-YYYY	BKPF-BUDAT
Gross Amount	Tax Base Amount in Local (Company code) Currency	/IDT/D_TAX_DATA- GROSS_AMOUNT

OUTPUT FIELD NAME	SAP FIELD NAME	TABLE FIELD NAME
Tax Amount	Tax Amount in Local (Company code) Currency	/IDT/D_TAX_DATA-TAX_AMOUNT
Document currency	Document currency key	/IDT/D_TAX_DATA - GROSS_AMT_CURR
Optional UDF 1 to 20	Custom field based on custom BAPI implementation	

Extract Selection Screen Field Definition for ERP Recon Extract New

The table below shows in detail each of the selection screens available at runtime of the Reconciliation Extract:

DESCRIPTION	FIELD NAME	TYPE	OPTIONAL/REQUIRED	DEFAULT NAME
Company Code	BKPF-BUKRS	Parameter	R	N/A
Fiscal Year	BKPF-GJAHR	Parameter	R	N/A
Posting Date	BKPF-BUDAT	Select options	O	N/A
Posting Period	BKPF-MONAT	Select options	O	N/A
Tax Code	BSEG-MWSKZ	Select options	O	N/A
Select Zero/Exempt tax records	N/A	Check Box	O	X
Download File	N/A	Check Box	O	X
Application Server	RLGRAP-FILENAME	Parameter	O	TVARV variable value
Presentation Server	RLGRAP-FILENAME	Parameter	O	User parameter value
Send Data to NGRA	N/A	Check Box	O	X
Company Code Prepend	N/A	Parameter	O	N/A
Package Size	N/A	Parameter	R	10000

ONESOURCE Indirect Tax Transport Objects for ERP Recon Extract New

This section lists the technical objects delivered with the Reconciliation Extract Report transport provided:

OBJECT NAME	OBJECT TYPE	DESCRIPTION OB OBJECT
/IDT/	Name space	Container for all objects delivered by Thomson Reuters
/IDT/RECON_EXTRACT	Package	Package containing all report development objects
/IDT/RECON_EXT_CL	Transaction code	Transaction code to run the Reconciliation extract report
/IDT/RECON_EXTRACT_CL	Report program	Main program
/IDT/RECONEXT_SS	Include	Selection screen is defined under this include
/IDT/BADIRECON_IMP_EXTRACT	BADI definition	BADI definition to add logic for 2 0 user defined fields
/IDT/EXTRACT_OUTPUT	Structure	Output file structure
/IDT/INT_RECON_EXTRACT	Interface	BADI Interface
/IDT/CL_BADIRECON_EXTRACT	Class	Class implementing the BADI interface
/IDT/EXTRACT_UDF	Structure	BADI return data structure

Enabling Custom Fields for ERP Recon Extract New

The file format of the ERP Reconciliation New Report allows for 20 user defined fields (UDF) as pass through elements. Companies wishing to use these fields can use custom code to set a value from SAP and extract it into one of the UDF's. These values then will be passed into the NGRA Reporting reconciliation tables through ERP Recon Integration layer and show in the ERP Document Reconciliation Report process.

A BADI /IDT/BADIRECON_IMP_EXTRACT has been provided as part of the SAP ERP Reconciliation Report - New which can be implemented by the customers. The BADI method returns the 20 UDF's in the structure

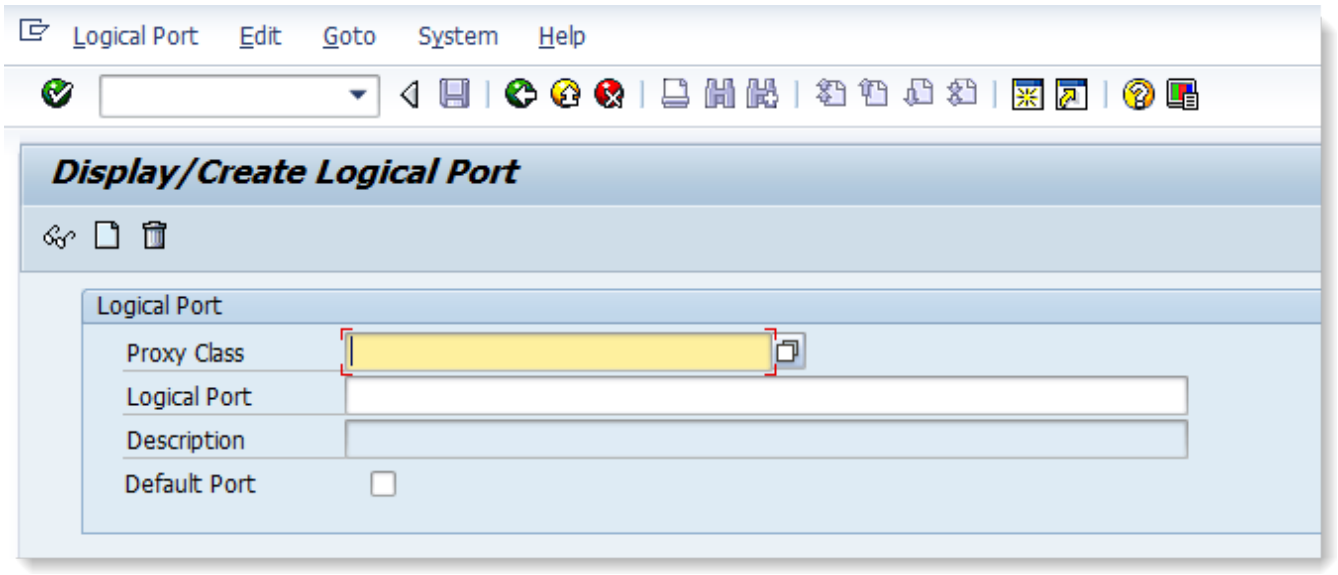
/IDT/EXTRACT_UDF.

LOGICAL PORT

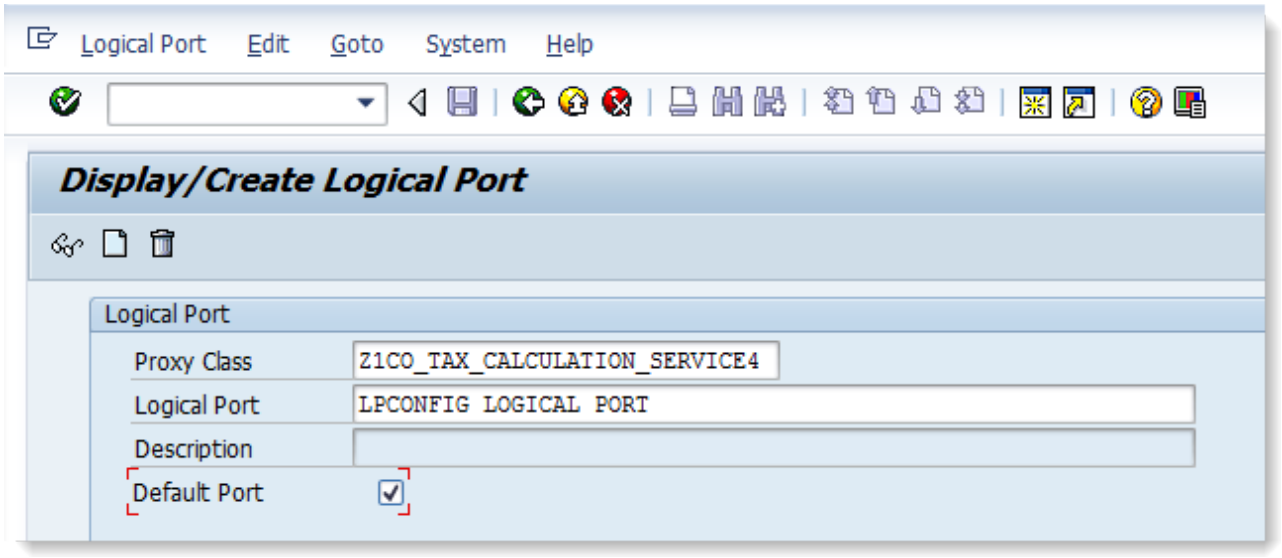
An alternate way of configuring the communication between SAP and Determination is using the logical port configuration. This option might be used if there is no J2EE layer enabled in the SAP environment Integration is deployed in.

Note: Please note, this method has not been extensively tested. The below setup screen is a courtesy in case you would like to use this alternate way of setup. We don't recommend using it as a production setup.

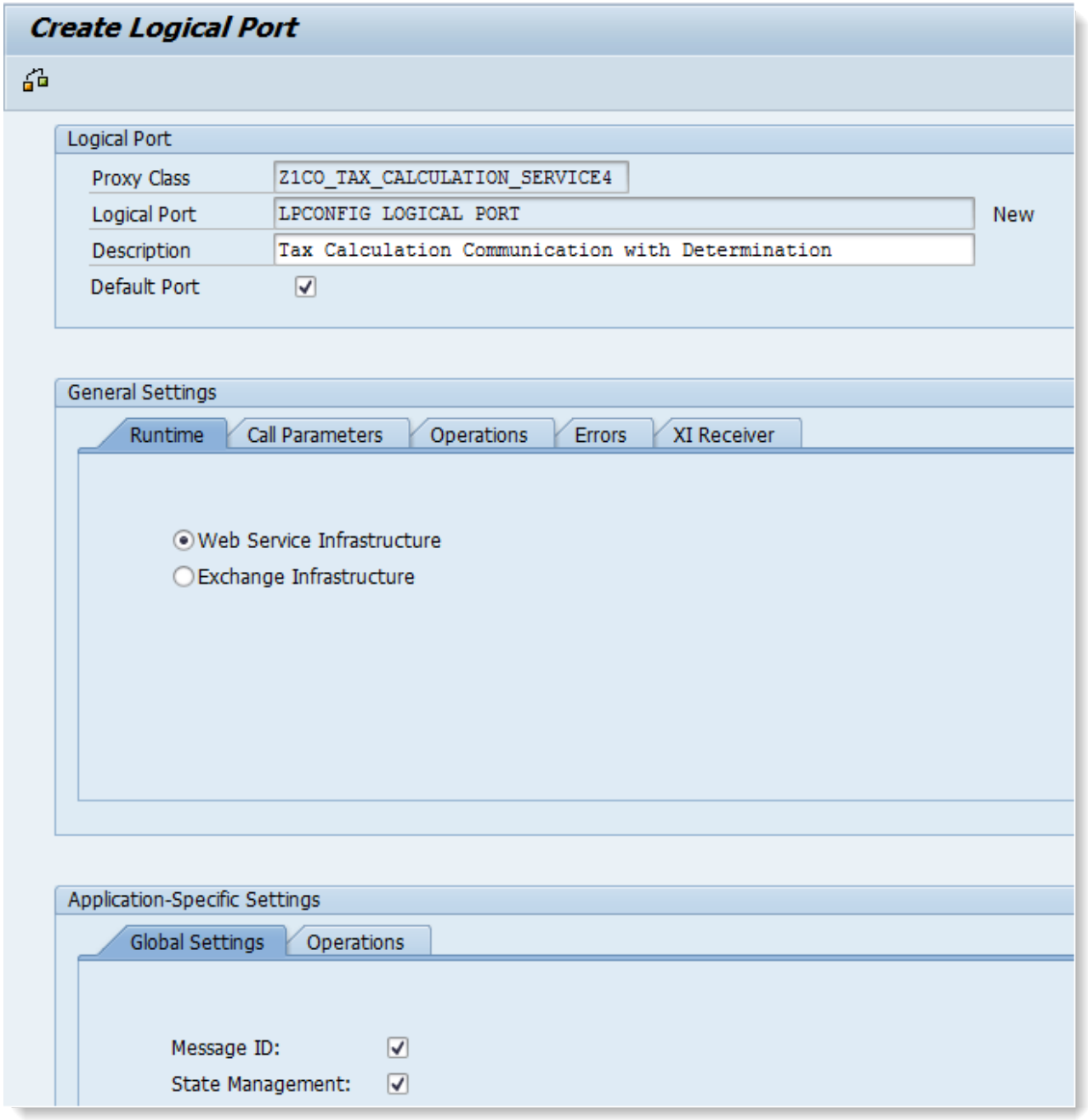
1. Transaction Code: LPCONFIG



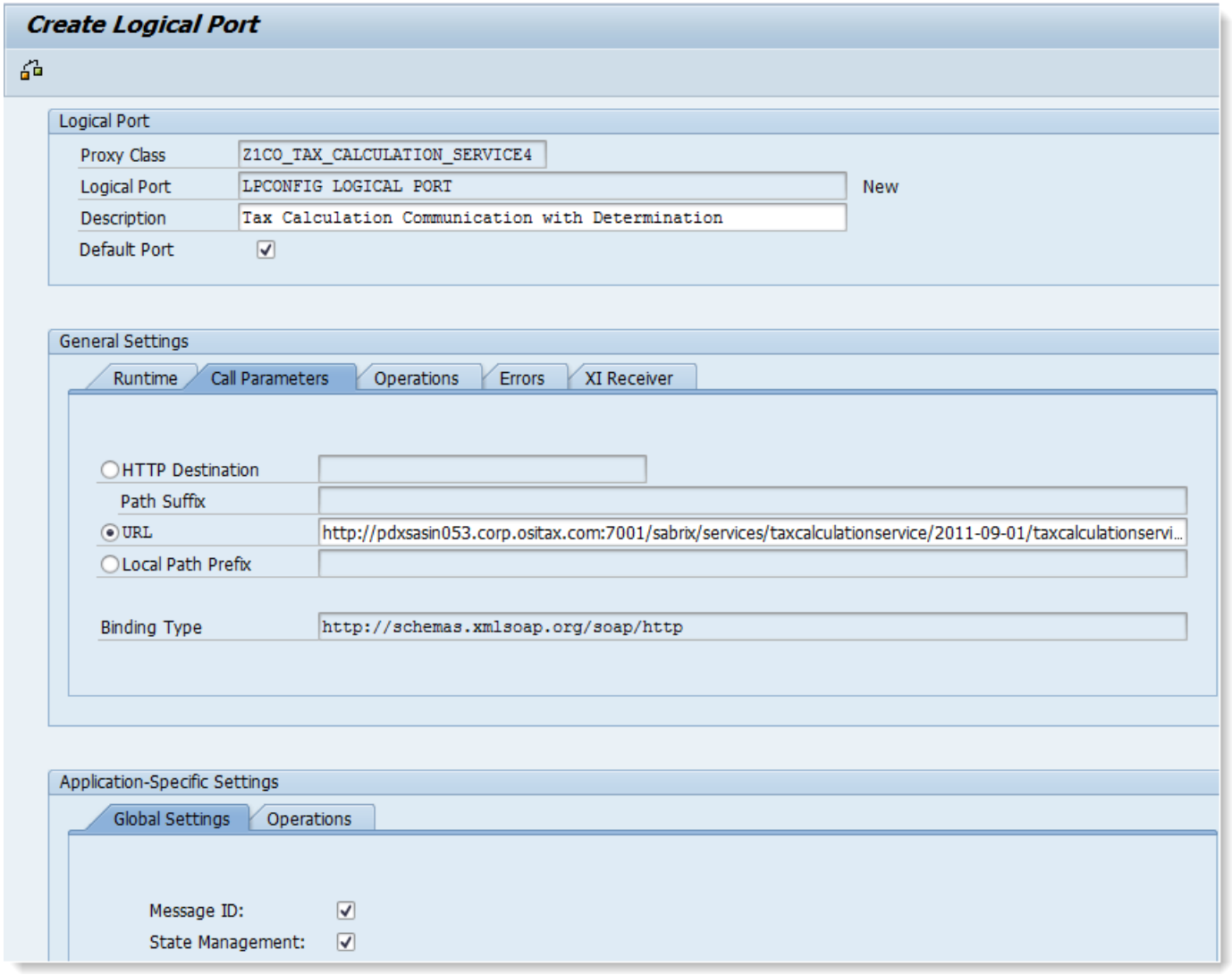
2. Use the Proxy Name from Creating an SAP Proxy section and create a Logical Port Name. Then Check “Default Port”



3. Press to Create the logical port. Fill in the following values:
   Logical Port Name: Name of the port you want to create
   Description: Text explaining to use of the port
   Default Port: Needs to be checked
   On the Runtime tab select Web Service Infrastructure and in the Global Settings tab make sure both Message ID and State Management are checked



4. Click on the Call Parameters tab, check the URL and then enter the calculation URL as follows: http://<HOSTNAME>:<PORT>/sabrix/services/taxcalculationservice/2011-09-01/taxcalculationservice



Note: Replace <host> with the name of the computer hosting the application server and replace <port> with the port number. http://<HOSTNAME>:<PORT>/sabrix/ should take you to the Determination logon screen.

5. Save your logical port and then Activate it.
6. You can follow steps outlined in the SOAP Tax Calculation Test section to test your logical port.