

ONESOURCE INDIRECT TAX INTEGRATION FOR SAP

INSTALL AND PROGRAMMERS GUIDE

PRODUCT VERSION 6.4.3.1

Document Version 1

© 2017 Thomson Reuters/ONESOURCE. All Rights Reserved. Proprietary and confidential information of Thomson Reuters. Disclosure, use, or reproduction without the written authorization of Thomson Reuters is prohibited. In compliance with the license agreements for the Open Source Libraries leveraged by Thomson Reuters, our customers can obtain copies of these libraries by contacting Technical Support at <https://tax.thomsonreuters.com/support/onesource/indirect-tax/>

The software documented within is Patent Pending in the United States.

DOCUMENT HISTORY

| Version Number | Version Date | Summary |
|----------------|--------------|--|
| v1 | Sept 1, 2017 | First version of this guide for ONESOURCE Indirect Tax Integration for SAP 6.4.3.1 |

Table of Contents

| | |
|--|-----------|
| Introduction | 1 |
| Welcome to ONESOURCE Indirect Tax Integration for SAP | 1 |
| Who Should Read This Guide? | 1 |
| Prerequisites | 2 |
| Resources | 3 |
| Support Protocol | 4 |
| Style Conventions | 4 |
| Integration Overview | 6 |
| What is Integration for SAP Version 6? | 6 |
| Quick View of New Features | 6 |
| Benefits of Using ONESOURCE Indirect Tax | 10 |
| Seamless Integration | 10 |
| Accurate Tax Calculation | 10 |
| Auditing/Reporting | 10 |
| User Processes That Trigger Determination Tax Calculations | 10 |
| Architecture and Design Overview | 11 |
| The ONESOURCE Indirect Tax Suite | 11 |
| Glossary | 14 |
| The Data Integration Model: The Train Station Analogy | 15 |
| Pre-Installation Steps | 17 |
| India GST SAP OSS Notes | 18 |
| Downloading the Software | 20 |
| Downloading the Documentation | 20 |
| System Preparations | 21 |
| Installing Integration | 22 |
| Special Instructions for 6.4.3.0 release | 22 |
| Importing Transport Objects | 22 |
| Adding User Menu to a Role | 23 |
| Adding Include Statements | 25 |
| Creating Condition Value Formulas | 26 |
| Creating Scale Based Formula | 28 |
| Condition Base Value | 29 |
| SD User Exit Code | 29 |
| Purchasing User Exit Code | 31 |
| Price Condition User Exit Code | 32 |
| Implicit Enhancements | 34 |
| Hana System Manual Adjustments Required Post Install | 35 |
| Brazil Implicit Enhancements | 36 |
| Reviewing BTE Events | 40 |
| Creating a Log Number Range | 41 |
| Connecting SAP and Determination | 42 |
| Hosted Customers Connectivity | 42 |

| | |
|---|------------|
| Accessing Determination UI | 42 |
| Setting Up the SOAP Interface Proxy | 43 |
| Creating an SAP Proxy | 43 |
| Setting up SOAMANAGER | 52 |
| WS Security Considerations on the Proxy for a Hosted Environment | 57 |
| Optional WS Security BAdl for the Proxy | 59 |
| Structure Table Modification After Proxy Set Up or Custom Journey Creation | 60 |
| Updates to Custom Tables | 61 |
| ConfigurationDetermine_Condition_Type_for_Taxes | 61 |
| SOAP Tax Calculation Test | 61 |
| SOAP Testing | 61 |
| Programmer Guide | 67 |
| Data Popup Tool | 67 |
| Developer's Log | 74 |
| Developer Log Configuration | 75 |
| Developer Log Viewer | 76 |
| Deletion of Developer Logs | 76 |
| Performance Improvement with Dynamically Generated Code | 76 |
| Clean Up Program for Dynamically Generated Field Mapping Code | 77 |
| Adding Custom Fields to /IDT/D_TAX_DATA Table | 79 |
| Simple Expressions | 81 |
| User-Exit in Field Mapper | 82 |
| Field Mapper Journey Drop Down List | 84 |
| Custom Address Source | 85 |
| Debugging | 89 |
| Optional BAdI or Functions that Can Be Instantiated | 91 |
| /IDT/BADI_ADJUST_TAX_SUMMATION | 91 |
| /IDT/BADI_ADJUST_PROXY | 92 |
| Adding IDT Address Field to Invoice Line Item Entry Screens | 92 |
| Adding the ZZIDTOVAD Field to the MIRO PO Reference Tab | 102 |
| Adding Address Field to Unplanned Delivery Function in MIRO Details Tab | 103 |
| Adding IDT Address Field to Service Entry Sheet Line Item | 108 |
| /IDT/BADI_SET_FREIGHT_LIV_UDC | 108 |
| Adding MIRO Multiple Account Assignment Address Overrides | 109 |
| Support of Multiple Account Assignment (MAA) in ML81N | 121 |
| Support of Override Address - Address Number at Service Level in PO and ML81N | 122 |
| Brazil: Tax Jurisdiction Code for Cost Center Created Through BAPI | 127 |
| India optional Hooks/Include Statements for Transaction J_1IG_INV | 128 |
| Appendix 1: References | 130 |
| List of Journeys | 130 |
| List of Routes | 134 |
| List of Bases | 136 |
| List of Delivered Tables | 140 |
| Configuration Tables | 140 |
| Transaction Tables | 142 |
| Reserved Attributes | 143 |
| List of Transaction Codes | 143 |
| Reconciliation Extract Program References | 146 |
| Reconciliation Extract Mapped From SAP | 146 |

| | |
|--|------------|
| Extract Selection Screen Field Definition..... | 147 |
| Onesource Indirect Tax Transport Objects | 148 |
| Enabling Custom Fields | 149 |
| Appendix 2: Logical Port | 150 |

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 audit database from which you can generate both standard and custom reports as well as returns.

ONESOURCE Indirect Tax Integration for SAP 6 is a totally new interface designed, built, and maintained by Thomson Reuters. It's a new global tax integration solution designed from the ground up with integration pointing into SAP ECC 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. The new Integration enables worldwide tax calculations, including VAT, US Sales and Use Tax, 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 new 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.

WHO SHOULD READ THIS GUIDE?

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 this sequence of documents:

1. User Guide
2. Installation and Programmers Guide
3. Configuration Guide

When working on Integration for SAP you must have a deep knowledge of the SAP tax features, covering all aspects of FI, MM, and SD and have spent significant 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 thoroughly 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 of 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 5. Please see tested platforms by Thomson Reuters in Platform Information section.
- 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

| Resource | Description |
|--|--|
| <u>Customer Support</u> | Look for answers in the Knowledge Base, or to open a support ticket. |
| <u>User Guide</u> | This is an overview of the architecture, basic business processes and touch points 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 setup the tax processes in SAP. |
| <u>Installation and Programmers Guide</u> | <p>This guide instructs on how to install the Integration for SAP. The target audience is the Basis person that will process the application of the transports to the SAP system and the ABAP programmers that 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.</p> <p>This guide describes the supported combinations of operating systems, databases, and application servers/web containers.</p> <p>This guide lists the end-of-life dates for ONESOURCE Indirect Tax Integrations for SAP.</p> <p>Consult this guide to see which combinations of software we test with Integrations.</p> |
| <u>Configuration Guide SAP Tables</u> | This guide instructs how to configure and setup SAP tables and processes to enable tax calculations to meet your unique requirements. |
| <u>Configuration Guide ONESOURCE tables</u> | This guide instructs how to configure and setup our ONESOURCE Indirect Tax tables and processes to enable tax calculations to meet your unique requirements. |
| <u>Configuration Guide for Special Functions</u> | This guide instructs how to configure and setup SAP and Integration tables and processes to enable tax calculations to meet your unique requirements for special functions within SAP such as Plants Abroad, Cash Discounts, Deferred Taxes, Service Entry Sheets, etc. |

SUPPORT PROTOCOL

The ONESOURCE Indirect Tax Integration for SAP is built, maintained, and owned by Thomson Reuters Tax & Accounting Indirect Tax. The business unit has a dedicated group of SAP Business Systems Analysts, ABAP Programmers, and Quality Assurance people who have built this product. 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 steps to reproduce the scenario leading to the issue.
3. Provide system environment information such as your SAP Version, EHP and SP level, as well as the Integration version.
4. Open a support ticket with Indirect Tax at
<https://tax.thomsonreuters.com/support/onesource/indirect-tax/>

STYLE CONVENTIONS

Style conventions are a guide to how to interpret information.

Bold text indicates most user interface elements, such as:

- Data you are expected to enter, such as in a text field
- Pages, buttons, tabs, and field names
- Dialog boxes, drop-down lists, selections within lists, and check box titles
- Windows
- Menu items

Italic text indicates the following:

- File and folder names
- Software programming terminology and executable files
- Document titles

CAPITALIZED text indicates keyboard commands, such as ENTER, or database components.

Courier text indicates command-line input/output.

<brackets> indicate user entry. For example, <host> indicates you should replace the text and angle brackets with your server name.

Book titles are shown in italics and sections within a book are in quotation marks, such as “Tips and Tricks” in the *ONESOURCE Indirect Tax User Guide*.



Indicates suggestions or additional, detailed information.



Indicates important text that should be carefully reviewed before proceeding.

INTEGRATION OVERVIEW

WHAT IS INTEGRATION FOR SAP VERSION 6?

Integration version 6 is a brand new interface designed to significantly improve the user experience and dynamically expand current and future system capabilities for indirect tax calculations, reporting, and data mapping. The new platform moves away from the use of the SAP Standard Tax Interface and SAP JCo adapter to new functionality based on ABAP coding directly within ONESOURCE Indirect Tax's SAP registered namespace. Users are no longer limited by jurisdiction code based calculations and design originally written for the US and Canada sales tax structure. This provides far greater flexibility to meet challenging and ever changing global tax requirements and simplifies US and Canadian tax.

The new interface is designed with the global customer in mind and closely to 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, the new design will likely avoid costly modifications to Integration programs.

Tax code usage within the new interface 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 new logic 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. Exciting improvements have also been made 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 new features and functions that are now available with the new 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 NEW FEATURES

| Function | Description |
|---|---|
| All configuration within SAP tables | All configuration is now located in a separate ONESOURCE Indirect Tax SAP partner namespace of /IDT/. |
| New ONESOURCE user menu within SAP | New user 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. |

| Function | Description |
|--|--|
| New flexible field and address mapping | New 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. This was also updated in the 6.4.1.0 release to include major performance enhancements with the addition of dynamically generated internal code whenever a mapping line is changed or added. Table user interface has also been updated. |
| Tax data storage for all business transactions | A new table has been created to store data from Determination in SAP for downstream processes for all business processes calculating tax. This limits and ultimately will avoid appends to SAP KONV table. Tax details are stored for SD, MM, and FI transactions. |
| New tax code and account assignment using Determination Tax Code Qualifier | Tax code and account key logic used by SAP to assign the G/L account number are now established via Determination's Tax Code Qualifier function and the use of specific condition logic to assign the correct codes and G/L account based on reporting needs. |
| New 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. Changes can be made on the fly without taking down Integration and stopping tax calculation. New flexible log viewer is provided for searching and displaying tax calculation logs to quickly troubleshoot taxing issues. |
| New log file delete/archive process | Menu option has been added to allow management of the log entries by deletion or download to zip file. |
| New Developer log report and configuration table | An additional log for developers has also been added as of release 6.4.1.0 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. |
| New Developer log delete function | Ability to delete old developer logs that were created for analysis and debug purposes. |
| New audit report | New 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 | SAP users on ECC 6.0 EHP5 or greater can now take advantage of SD, MM, and FI tax calculations. 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. |

| Function | Description |
|---|--|
| Removal of jurisdiction code method | Taxability now controlled by the address of the taxable entity and not by the assignment of a jurisdiction code. |
| Removal of limit on attribute mapping | Users are no longer limited by the number of fields they can map from SAP to the Determination request and the response from Determination back to SAP. Ability to map 40 attributes at both header and again at line item level. |
| Dynamic creation of tax lines removing limitation of 10 jurisdiction levels | External tax interface was limited to 10 jurisdiction levels for the summarization of tax authority information by level. This is now a 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. |
| 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. |
| Ability to use standard VAT reports | Removal of jurisdictional calculation method now provides the access and usability of many native SAP tax reports. |
| Standard SAP functionality now available with removal of jurisdictional method of calculation of tax. | Native SAP functions that are limited to non-jurisdictional calculations can now be made available for external tax calculations. |
| 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. |
| New SOAP interface replacing SAP's External Tax Interface | The JCo Server and External Tax Interface have been replaced with 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. |
| 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 |

| Function | Description |
|--|--|
| | 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 program with SAP standard configuration. |
| US Tax Report | US Tax Report which is clone of standard SAP US Report broken with Integration being non-jurisdictional is now available for use on US company codes within the Reports and Tools Menu. |
| New Brazil Enablement | New 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. |
| New India Enablement | New fields and functionality that allows the SAP India Country version tables and logic to work with our Determination Tax Engine in a non-jurisdictional environment. Special note that due to status of SAP release and India government final approval, there will possibly be another patch release to adjust for late changes after our release date. |
| HANA port over | As of Integration for SAP version 6.4.3.0, the system has been adjusted for needed table maintenance so that it can be easily installed on HANA version 1511. Higher versions of HANA release may require additional adjustments due to rapid SAP development path for HANA. |

New logic can be added to the Integration with greater ease and less ABAP programming due to the structures, methodologies, and dynamic table offerings. This will aid system users and ABAP programmers in meeting your specific requirements that are not currently part of this release, and provide a platform and process for incorporating these needs into standard product updates in the future.

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 is the premier solution for your Sales, Use Tax, and VAT needs. ONESOURCE Indirect Tax 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.

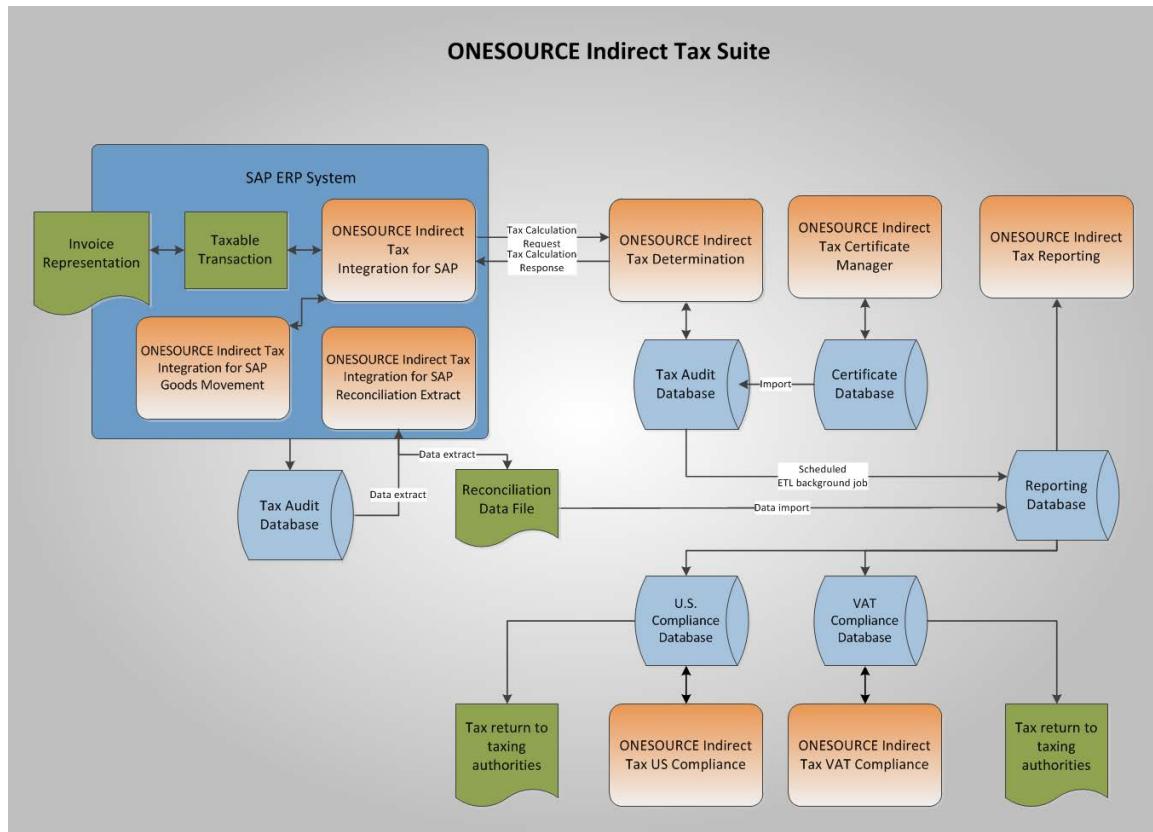
Once Determination is fully integrated with SAP, you can automatically generate correct tax results during the following actions:

- 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 shown in this screen shot.



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:

Newly updated 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 new version uses all of the table

tables and SOAP interface technology that was provided as part of Integration Version 6. This new menu driven version takes advantage of our entire new table mapping and menu features in order 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:

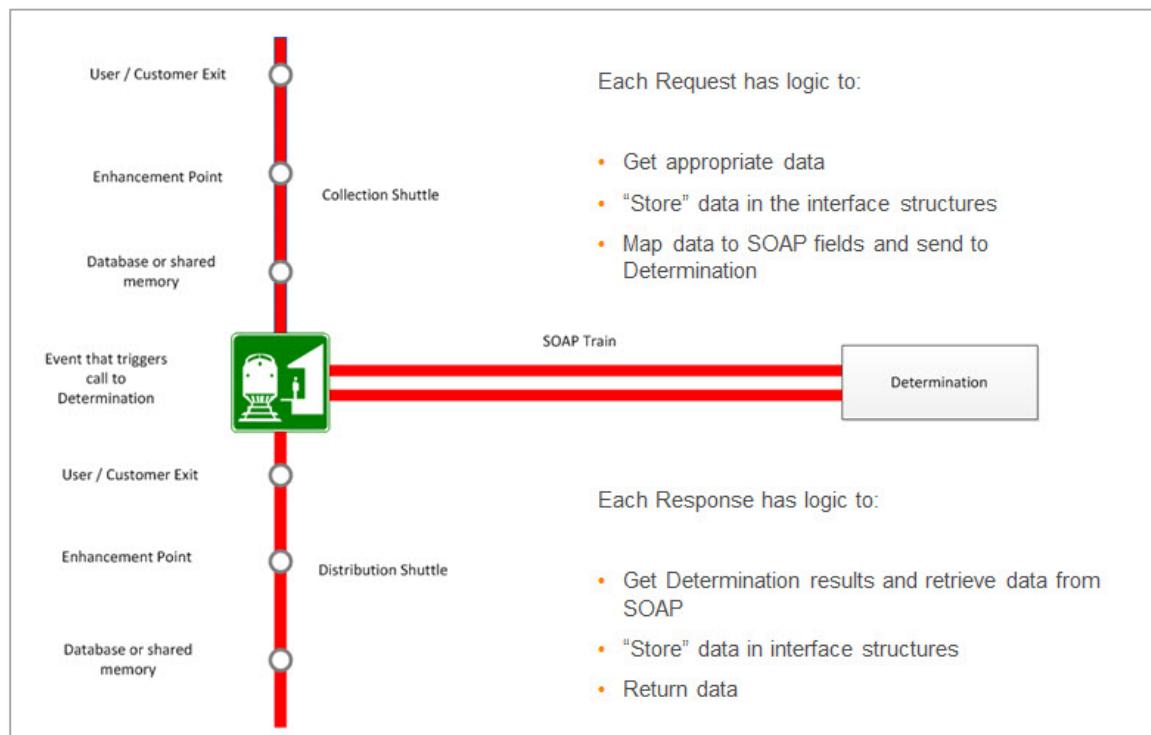
| Term | Meaning |
|--|---|
| SOAP (Simple Object Access Protocol) | SOAP is a way to build connections between software applications across networks including the internet. It works very much like a Yahoo search, where you pass in a request and get a response from a server. SOAP requests and responses 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. |
| 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

In order 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 new 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 in the following image 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 of 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 of the routes and their use see the *Installation Guide* section on additional information on routes.

PRE-INSTALLATION STEPS

Before you can start with the actual installation and configuring of Integration you will need to perform some preparations. This chapter outlines these steps.

For Non- India GST Update

If you wish to update your system with the technical and functional changes in this release that do not apply to the India GST Enablement then you can choose to skip this next section and go directly to [Downloading the Software](#).

For India GST Update

If you are installing this release for the purpose of taking advantage of the new functionality for India GST tax regime then there are many additional steps to perform. It is imperative you read and complete these instructions in this order prior to loading the Integration software transports.

1. Make sure you are on the correct support pack level for your system and update if necessary. See OSS note 1175384.
2. Install all GST draft OSS notes per list below.
3. Compare our list of OSS notes to a current SAP list in OSS note 2421394.
4. Apply notes added to OSS note 2421394 since our list was published.
5. Upgrade to Determination 5.9.0.0.
6. Install our Integration Transports per below.
7. India configuration and mappings per section of “India Enablement” in [Configuration Guide for Special Functions](#).



As a pre-requisite, SAP OSS note number 1175384 must be reviewed and your systems upgraded to the appropriate support pack level per this Note. You must do this prior to importing the other India GST OSS notes listed below AND the pre-requisite update to Determination 5.9.0.0.



As a pre-requisite, Determination 5.9.0.0 which supports India GST logic and has supported content is required to install prior to installing Integration for SAP 6.4.2.0.



After Installing you will have additional India configuration that must be done. Go to the [Configuration Guide for Special Functions](#) “India Enablement” section for configuration steps.

INDIA GST SAP OSS NOTES

At this time the India Enablement is based on a set of Draft SAP OSS notes that were published at the time of this release. Finalization of the India GST tax laws and SAP's solution for India GST is still in process. Because of this you will need to install this list of OSS notes manually as part of your installation and configuration tasks prior to development and processing of your India GST solution. Number of notes to install may vary depending on what EHP level you are on.



SAP OSS notes clearly specify that it is mandatory that users convert to the tax procedure TAXINN in order to take advantage of the India GST developments they are providing. All of our testing and processes rely on this assumption and usage of TAXINN tax procedure. In theory the older TAXINJ tax procedure may work but we do not test for this and caution users to convert to TAXINN per SAP requirements. If for some reason you cannot convert to TAXINN then we recommend that you talk to your Professional Services representative about alternatives. We do not support the older tax procedure.

As SAP may publish additional OSS notes prior to their final release you may need to address any additional notes by comparing their current list to the list that we have here. Additional notes will require your extra attention, testing and adaptation to this release. We anticipate a patch release that we will complete once the final SAP solution for India GST is final. It should address these additional notes and adjustments.

Please make sure that the following list of OSS notes and their prerequisites have been added to your system prior to our 6.4.2.0 release installation transports. Also do comparisons of this list to OSS note 2421394 to make sure you pick up and install any additional notes that are not identified below:

| OSS Note | Description |
|----------|--|
| 2421394 | GST India Pre-requisites/FAQs /Solution (generic overview and list of other OSS notes. This note is continuously updated with new OSS draft notes for India GST. |
| 2194689 | IN_Goods and Service Tax |
| 2405502 | GST India - Changes to Master data- DDIC activities |
| 2385575 | GST India - Changes to Master data |
| 2415115 | GST India - Changes to Transaction data- Data Dictionary activities |
| 2407980 | GST India - Changes to tax procedure and pricing procedure |
| 2194689 | IN_Goods and Service Tax |
| 775919 | Missing tax number categories of business partner |

| | |
|------------|--|
| 2252781 | FAQ on Tax procedure migration from TAXINJ to TAXINN |
| 2417939 | GST Vendor classification issue in FV11 |
| 2410053 | Control Code and GST Vendor Classification fields not available in Field catalog |
| 2410917 | Activating Line item wise tax calculation for posting GST relevant billing document sales side - VF01 |
| 2410105 | Sales related changes for India GST for sales order creation and Invoice posting |
| 2417506 | GST India - Utilities |
| 2315692 | Information on GST |
| 2056080 | Notification No.21/2014-Central Excise (N.T.) dt. 11.07.2014. - Credit should not be availed after 6 months of date of invoice |
| 2073062 | Line item wise tax calculation not required for certain FI transactions like F-02, FB60, FB01, F-48 for Indian company codes |
| 2293375 ** | Line item wise tax calculation not happening for CRM billing document passed to ERP through BAPI_BAPI_ACC_DOCUMENT_POST |
| 2059696 | Notification No.21/2014-Central Excise (N.T.) dt. 11.07.2014- Manual changes as prerequisite for SAP Note 2056080 |
| 1568306 | Ed not calculated when BAPI is used for vendor invoice post |
| 1911521 | Wrong excise captured for depot purchase, inventory tax- J1IG |
| 1969329 | Unplanned delivery cost gets posted incorrectly in Vendor Invoice when posted using BAPI_INCOMINGINVOICE_CREATE |
| 2416018 | GST India: Stock Transfer |
| 2420891 | GST India: Error FF709 occurs while posting GST Invoice |
| 2419214 | GST IN: Changes to MIRO Invoice item level screen |
| 2419215 | GST IN: Changes for FI Invoice item level screen |
| 2378678 * | GST IN: BAdl implementation in Enjoy transactions of FI and MM |
| 2376723 | GST IN: BAdl definition for screen enhancement in Enjoy transactions of FI and MM |

| | |
|---------|--------------------------------|
| 2428495 | GST In: Corrections to 2416018 |
|---------|--------------------------------|

* Note; As of 3/8/2017 we have seen an issue in the OSS note code for 2378678 that is not correct and is not allowing the GST partner field to display on the screen. It is however in the database table. Customers may need to address this for display of the field on screen on FB60, FB65, and MIRO if SAP has not provided a fix.

** Note: OSS note 2293375 refers to CRM billing through BAPI_ACC_DOCUMENT_POST. This BAPI is not used by the Global Next Integration. A system user may not need to apply this OSS note unless they have customizations applied.

There may be a couple of prerequisite OSS notes that are required in addition to those called out by this list. SAP may also address other issues with additional Draft view OSS notes after this list was created.

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.



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, Determination has to be installed as well. Make sure you have Determination installed and configured.

Consult the [Platform Support Guide](#) to installation to make sure you have the proper version prerequisites met.

INSTALLING INTEGRATION

SPECIAL INSTRUCTIONS FOR 6.4.3.X RELEASES

The major feature of this release involves the testing and adaptations that were required for install and implementation within a Hana environment. Testing was done for this release within a HANA 1511 version as well as ECC 6.0. Code was adapted so that there is only one code version for both environments. Users wishing to use the Integration for SAP with higher or more recent Hana release versions may encounter newer issues that must be addressed due to the rapidly evolving platform changes for Hana. Users should always fully test their needed function and configuration requirements keeping this in mind.

This release is different from our normal release process in that it also includes the new India GST tax regime changes that are targeted to be active by the India Government as of September, 2017. Although our prior 6.4.2.0 release addressed most of these changes, this release also included a new OSS note for an Intra-Company STO transaction for India that is new. The new regime change to GST is in process and is not fully approved and updated by both the India Government and by SAP development. SAP has released a set of Draft OSS notes in order to give users an advanced look into their future GST release. When final they will be changed from a Draft status and incorporated into their new release.

Until then we are applying our changes based on draft tax law and draft OSS Notes from SAP. Because of this you will need to load all of the draft OSS notes to your system prior to moving our list of transports into your system. Please go to the Special Functions Guide that is attached to this release and follow the instructions in the India Enablement section regarding the load and comparison of these notes to the system. You will need to add all listed notes to your system and address any new notes as well that SAP has added since the 6.4.2.0 release list was published. We anticipate an additional patch release after this one to catch any new issues that were identified after our release.

IMPORTING 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 of our objects are in the Thomson Reuters Indirect Tax name space /IDT/. We don't change any customer owned objects.

All Thomson Reuters delivered code and objects are in the /IDT/ name space. None of our transports will directly update SAP owned objects, user-exits, or similar. Once the transports have been successfully imported into your SAP system you will need to perform some manual steps to add coding blocks into SAP objects to call the Thomson Reuters delivered tax interface code.



This install document assumes a fresh install of Integration. If you are a customer who received a pre General Release version please contact Thomson Reuters Customer Support. If you are installing this new version into an environment with our prior version 5.x Integration already installed please contact Thomson Reuters Professional Services to assist you in the upgrade.

If you are upgrading your system from a prior 6.x Integration version, there is an additional upgrade transport that you must do to remove objects no longer used in the new release. This Upgrade transport must be transported and completed first in order to avoid any errors in the Code transport. Import the provided transports in the following order:

| Transport | Content |
|------------|---|
| QO4K900313 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Upgrade |
| QO4K900314 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Code |
| QO4K900315 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Config |
| DS4K900350 | Optional Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 India Code. Only install this one if you are using India GST. Do not install if you have not already applied all of the required OSS notes for India GST as this code is dependent on the notes supplied by SAP |
| QO4K900317 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - BTE |
| QO4K900316 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Role |



Generation of some of the ABAP objects imported might take some time. Please make sure to check successful import of each transport before proceeding to the next one in the list.

If you are doing a brand new install to your system, import the provided transports in the following order. You do not need to use the upgrade transport that is shown above for upgrade customers.

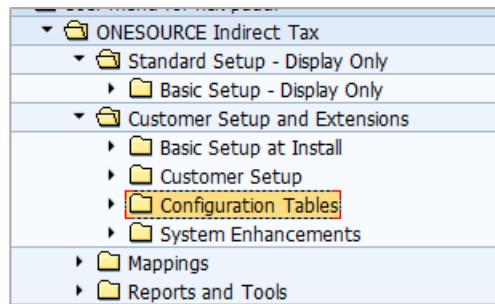
| Transport | Content |
|------------|---|
| QO4K900314 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Code |
| QO4K900315 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Config |
| DS4K900350 | Optional Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 India Code. Only install this one if you are using India GST. Do not install if you have not already applied all of the required OSS notes for India GST as this code is dependent on the notes supplied by SAP |
| QO4K900317 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - BTE |
| QO4K900316 | Thomson Reuters ONESOURCE IDT Integration 6.4.3.1 - Role |

ADDING USER MENU TO A ROLE

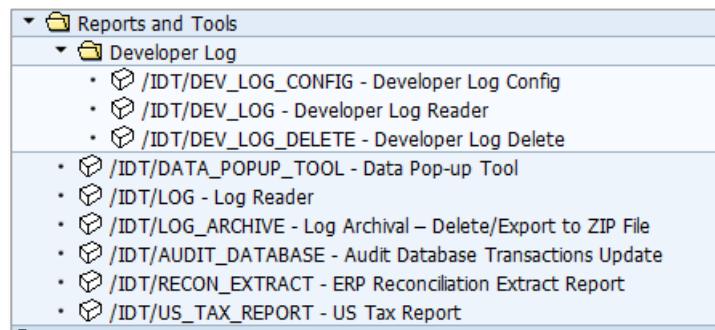
For users to be able to access the new ONESOURCE Indirect Tax for SAP setup, configurations, and reports we delivered an SAP User Menu. For users to be able to access the User Menu they would need to be given access to the /IDT/GENERAL or /IDT/USER role.

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

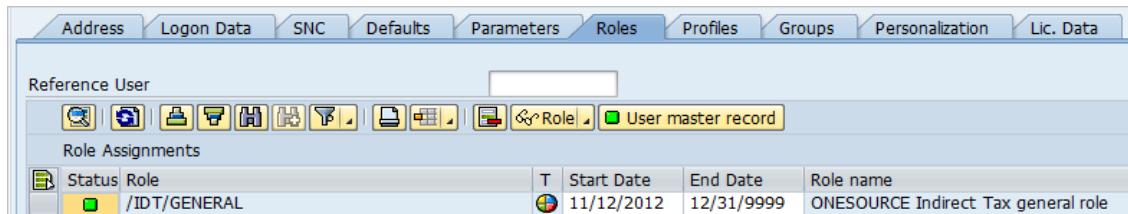
Adding User Menu to a Role



/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.



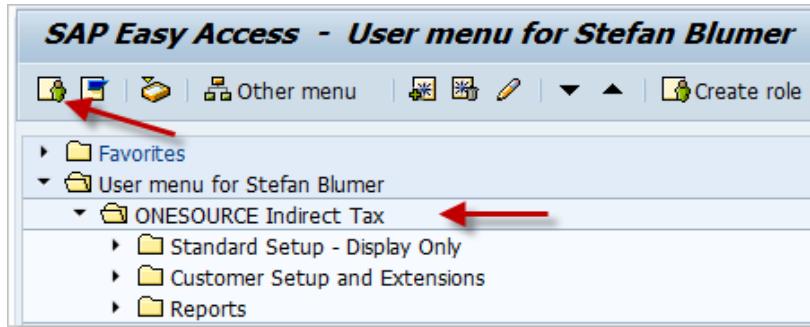
To assign the role to a user go to Transaction Code **SU01**. On the **Roles Tab**, insert the new role name, beginning and ending dates as shown below.



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 has to 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.



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:

| Maintain: Formulas Condition value | | | | |
|---|----------------------|-------------------------------------|-------------|--|
|     | | | | |
| Maintain: Formulas Condition value | | | | |
| Routine number | Description | Active | Application | |
| 990 | IDT Collect and Call | <input checked="" type="checkbox"/> | | |
| 991 | IDT Return Tax (n-g) | <input checked="" type="checkbox"/> | | |
| 992 | IDT Collect Tax Data | <input checked="" type="checkbox"/> | | |
| 993 | IDT Call Determin. | <input checked="" type="checkbox"/> | | |
| 994 | IDT Return Tax (grp) | <input checked="" type="checkbox"/> | | |

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

```
INCLUDE /IDT/COND_FORMULA_CALL_DET_NG.
```

| | | |
|---------|--|--------|
| Include | RV64A990 | Active |
| 1 | FORM frm_kondi_wert_990. | |
| 2 | | |
| 3 | INCLUDE /idt/cond_formula_call_det_ng. | |
| 4 | | |
| 5 | ENDFORM. | |

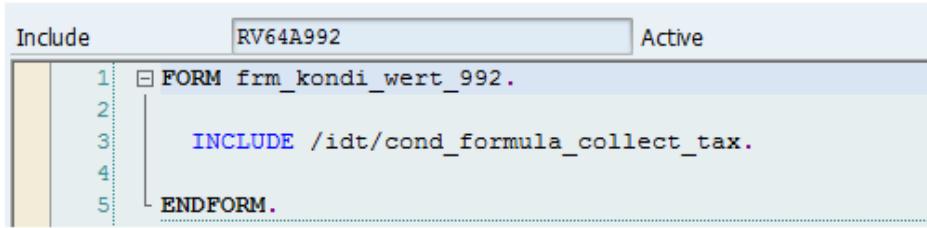
2. Update tax condition with tax data (non-group process) [RV64A991]

```
INCLUDE /IDT/COND_FORMULA_UPD_TAX_NG.
```

| | | |
|---------|---------------------------------------|--------|
| Include | RV64A991 | Active |
| 1 | FORM frm_kondi_wert_991. | |
| 2 | | |
| 3 | INCLUDE /idt/cond_formula_upd_tax_ng. | |
| 4 | | |
| 5 | ENDFORM. | |

3. Collecting line tax data (group process) [RV64A992]

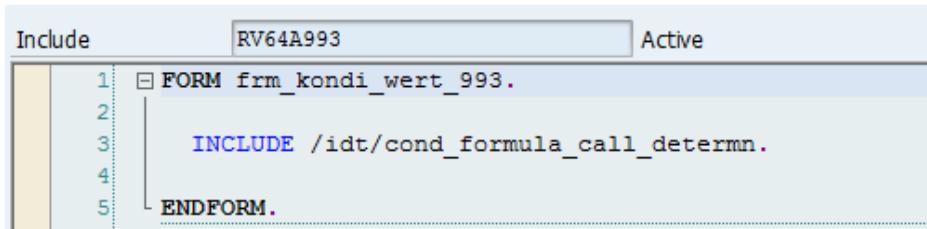
```
INCLUDE /IDT/COND_FORMULA_COLLECT_TAX.
```



```
Include RV64A992 Active
1  □ FORM frm_kondi_wert_992.
2
3  INCLUDE /idt/cond_formula_collect_tax.
4
5  ENDFORM.
```

4. Calling Determination (group process) [RV64A993]

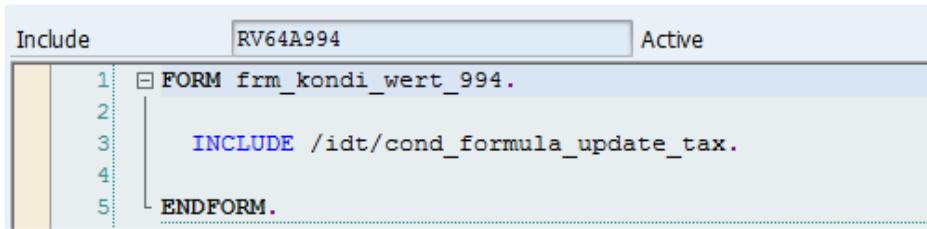
```
INCLUDE /IDT/COND_FORMULA_CALL_DETERMN.
```



```
Include RV64A993 Active
1  □ FORM frm_kondi_wert_993.
2
3  INCLUDE /idt/cond_formula_call_determin.
4
5  ENDFORM.
```

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

```
INCLUDE /IDT/COND_FORMULA_UPDATE_TAX.
```



```
Include RV64A994 Active
1  □ FORM frm_kondi_wert_994.
2
3  INCLUDE /idt/cond_formula_update_tax.
4
5  ENDFORM.
```

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: Routine number 992, Description IDT Tax (grp)

| Maintain: Formulas Scale base value | | | | |
|---|--------------------|-------------------------------------|-------------|--|
|     | | | | |
| Maintain: Formulas Scale base value | | | | |
| Routine number | Description | Active | Application | |
| 992 | IDT Init Tax (grp) | <input checked="" type="checkbox"/> | | |

1. Triggering the start of pricing (group process) [RV62A992]

`INCLUDE /IDT/SCALE_BASE_FORM_START_TAX.`

| Include | RV62A992 | Active |
|---------|---|--------|
| | <pre> 1 FORM frm_staffelbas_992. 2 3 INCLUDE /idt/scale_base_form_start_tax. 4 5 ENDFORM. </pre> | |

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: Routine number 994, Description IDT Return Tax (n-g)

| Maintain: Formulas Condition base value | | | |
|---|----------------------|-------------------------------------|-------------|
| Routine number | Description | Active | Application |
| 994 | IDT Return Tax (n-g) | <input checked="" type="checkbox"/> | |
| | | <input type="checkbox"/> | |
| | | <input type="checkbox"/> | |
| | | <input type="checkbox"/> | |

1. Updating the tax base value (non-group process)

INCLUDE /IDT/COND_FORMULA_UPD_BASE_NG.

| | | |
|---------|--|--------|
| Include | RV63A994 | Active |
| 1 | FORM FRM_KOND_BASIS_994. | |
| 2 | *{ INSERT D03K901995 | |
| 3 | INCLUDE /idt/cond_formula_upd_base_ng. | |
| 4 | *} INSERT | |
| 5 | ENDFORM. | |
| 6 | | |

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

`INCLUDE /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.`



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



Customers on an earlier version of 6.x will need to remove the include in user-exit EXIT_SAPLV60B_008.

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_SAPMLSP_030 / EXIT_SAPMLSP_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 new project needs to be created and activated in CMOD and it should include the two enhancements SRVESLL and SRVESSR.

1. EXIT_SAPMLSP_030

```
INCLUDE /IDT/SERVICE_ENTRY_SHEET_ITEM.
```

2. EXIT_SAPMLSP_010

```
INCLUDE /IDT/SERVICE_ENTRY_SHEET_HDR.
```

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.
```

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. Actually, there are two S4 Hana versions out there. One is pure S4 simple finance and another is S4 Enterprise management that 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 new structure PRCS_ELEMENTS_DATA which is included in database tables PRCD_ELEMENTS and PRCD_ELEM_DRAFT and structure PRCS_ELEMENTS."

In 6.4.3.x release we have a new standalone structure /IDT/PRCS_ELEMENTS that needs to be included in a new 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 the image on the following page.

```
/idt/auta,  
/idt/etc,  
/idt/uuid,  
/idt/not,  
/idt/item,  
/idt/hdct,  
/idt/dot,  
/idt/ttyp,
```

Additional note per SAP documentation:

Portion from *Cookbook_Pricing_ConditionTechnique_20160725.pdf*

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.

BRAZIL IMPLICIT ENHANCEMENTS

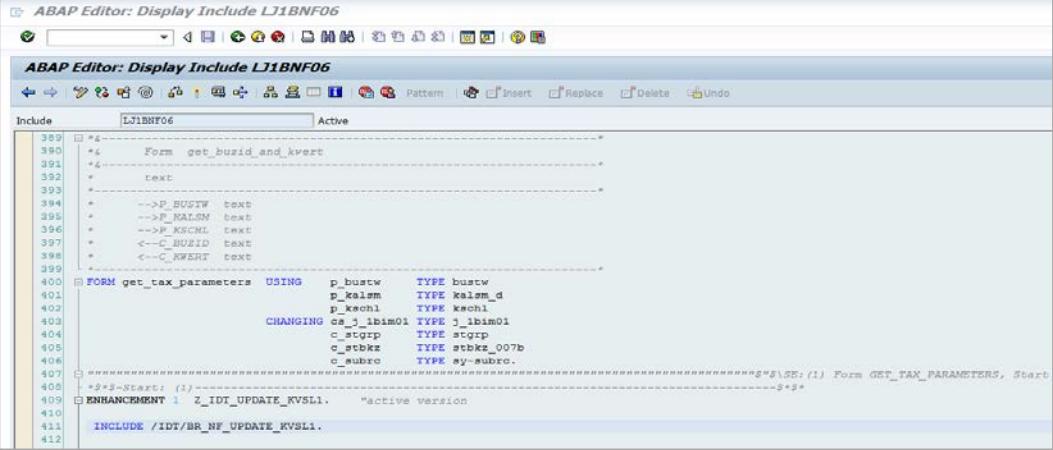


Important: The following Brazil includes and hooks should only be added to the system if you are planning to use the Integration 6.4 Brazil enablement logic. If you have elected to leave Brazil on the Integration 5.x old interface then by adding these hooks the program will create short dumps in your system. You can either elect to not add these includes into the programs or comment them out so as 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.



```

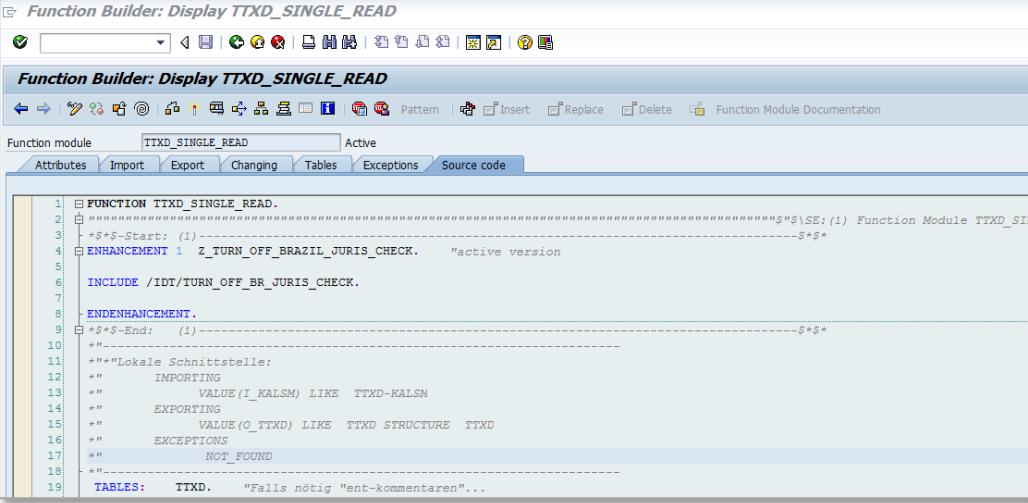
ABAP Editor: Display Include LJ1BNF06
ABAP Editor: Display Include LJ1BNF06
Include LJ1BNF06 Active
389 34 Form get_buzeug_and_kwert
390 34
391 34
392 34
393 34
394 34
395 34
396 34
397 34
398 34
399 34
400 34 FORM get_tax_parameters USING p_bustw TYPE bustw
401 34 p_kalsm TYPE kalsm_d
402 34 p_kalid TYPE kalid
403 34
404 34 CHANGING cs_j_lbimo1 TYPE j_lbimo1
405 34 c_stgrp TYPE stgrp
406 34 c_stbkz TYPE stbkz_007b
407 34 c_subrc TYPE ay_subrc.
408 34
409 34 ENHANCEMENT 1 Z_IDT_UPDATE_KVSL1. "active version
410 34
411 34 INCLUDE /IDT/BR_NF_UPDATE_KVSL1.
412 34

```

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.
Shown in the following image.



```

Function Builder: Display TTXD_SINGLE_READ
Function Builder: Display TTXD_SINGLE_READ
Function module TTXD_SINGLE_READ Active
Attributes Import Export Changing Tables Exceptions Source code
1 FUNCTION TTXD_SINGLE_READ.
2
3 *$*-$-Start: (1)-----$*-$
4 ENHANCEMENT 1 Z_TURN_OFF_BR_JURIS_CHECK. "active version
5
6 INCLUDE /IDT/TURN_OFF_BR_JURIS_CHECK.
7
8 ENDEHNANCEMENT.
9 *$*-$-End: (1)-----$*-$
10
11 *"" Lokale Schnittstelle:
12 *"" IMPORTING
13 *"" VALUE(I_KALSM) LIKE TTXD-KALSM
14 *"" EXPORTING
15 *"" VALUE(O_TTXD) LIKE TTXD STRUCTURE TTXD
16 *"" EXCEPTIONS
17 *"" NOT_FOUND
18 *"
19 TABLES: TTXD. "Falls nötig "ent-kommentieren"...
```

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.

Function Builder: Display ADDR_CHECK

Function module ADDR_CHECK Active

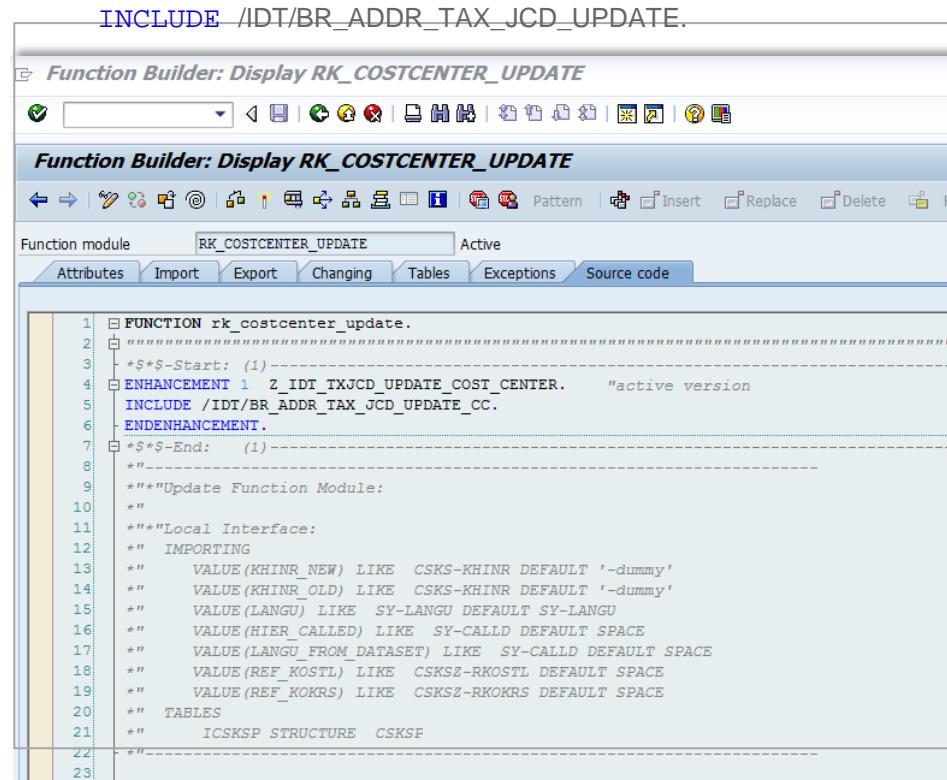
Attributes Import Export Changing Tables Exceptions Source code

```
1 FUNCTION addr_check.
2
3   *$*$-Start: (1) ----- $*$\SE
4   ENSHANCEMENT 1 Z_IDT_TXJCD_UPDATE.      "active version
5   INCLUDE /IDT/BR_ADDR_TAX_JCD_UPDATE.
6   ENDENHANCEMENT.
7
8   *$*$-End: (1) ----- $*$\SE
9
10  *"**"Local Interface:
11  *"  IMPORTING
12  *"    VALUE(ADDRESS_OBJECT_TYPE)  LIKE  SZAD_FIELD-ADDR_TYPE
13  *"    VALUE(ADDRESS_HANDLE)      LIKE  SZAD_FIELD-HANDLE OPTIONAL
14  *"    VALUE(ADDRESS_NUMBER)      LIKE  ADRC-ADDRNUMBER OPTIONAL
15  *"    VALUE(NATION)            LIKE  ADRC-NATION OPTIONAL
16  *"  EXPORTING
17  *"    VALUE(RETURNCODE)        LIKE  SZAD_FIELD-RC_ERRORS
18  *"  TABLES
19  *"    ERROR_TABLE STRUCTURE  ADDR_ERROR OPTIONAL
20  *"  CHANGING
21  *"    VALUE(ADDRESS_DATA_1)    LIKE  ADDR1_DATA STRUCTURE  ADDR1_DATA
22  *"      OPTIONAL
23  *"    VALUE(ADDRESS_DATA_2)    LIKE  ADDR2_DATA STRUCTURE  ADDR2_DATA
24  *"      OPTIONAL
25  *"    VALUE(ADDRESS_DATA_3)    LIKE  ADDR3_DATA STRUCTURE  ADDR3_DATA
26  *"      OPTIONAL
27  *"  EXCEPTIONS
28  *"    PARAMETER_ERROR
29
30  *"-
```

Adding Include Statements

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

INCLUDE /IDT/BR_ADDR_TAX_JCD_UPDATE.



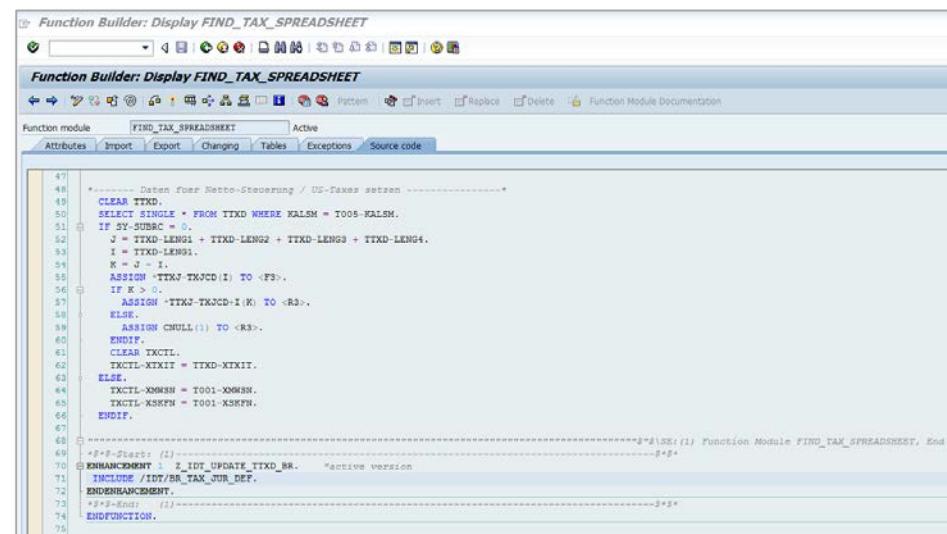
```

1 FUNCTION rk_costcenter_update.
2
3 *$*$-Start: (1)-----
4 ENHANCEMENT 1 Z_IDT_TXJCD_UPDATE_COST_CENTER. "active version
5 INCLUDE /IDT/BR_ADDR_TAX_JCD_UPDATE_CC.
6 ENDENHANCEMENT.
7
8 *$*$-End: (1)-----
9
10 *"**"Update Function Module:
11
12 *" IMPORTING
13   VALUE(KHINR_NEW)  LIKE CSKS-KHINR DEFAULT '-dummy'
14   VALUE(KHINR_OLD)  LIKE CSKS-KHINR DEFAULT '-dummy'
15   VALUE(LANGU)      LIKE SY-LANGU DEFAULT SY-LANGU
16   VALUE(HIER_CALLED) LIKE SY-CALLD DEFAULT SPACE
17   VALUE(LANGU_FROM_DATASET) LIKE SY-CALLD DEFAULT SPACE
18   VALUE(REF_KOSTL)  LIKE CSKSZ-RKOSTL DEFAULT SPACE
19   VALUE(REF_KOKRS)  LIKE CSKSZ-RKOKRS DEFAULT SPACE
20
21 *" TABLES
22   ICSKSP STRUCTURE CSKSP
23

```

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 (Eg: Service Entry Sheet, STO etc.).

INCLUDE /IDT/BR_TAX_JUR_DEF.



```

47
48 *----- Daten fuer Netto-Steuerung / US-Taxes setzen -----
49   CLEAR TTXD.
50   SELECT SINGLE * FROM TTXD WHERE KALSM = T005-KALSM.
51   IF SY-SUBRC = 0.
52     J = TTXD-LENG4 + TTXD-LENG2 + TTXD-LENG3 + TTXD-LENG4.
53     I = TTXD-LEN01.
54     K = J - I.
55     ASSIGN TTXJ-TXJCD(I) TO <F3>.
56     IF K > 0.
57       ASSIGN -TTXJ-TXJCD-I(K) TO <R3>.
58     ELSE.
59       ASSIGN CHULL(1) TO <R3>.
60     ENDIF.
61     CLEAR TXCTL.
62     TXCTL-XIXIT = TTXD-XIXIT.
63   ELSE.
64     TXCTL-XMMBN = T001-XMMBN.
65     TXCTL-XSKPN = T001-XSKPN.
66   ENDIF.
67
68
69 *----- End of Enhancement 1 -----
70 ENHANCEMENT 1 Z_IDT_UPDATE_TTXD_BR. "active version
71 INCLUDE /IDT/BR_TAX_JUR_DEF.
72 ENDENHANCEMENT.
73
74 *$*$-End: (1)-----
75 ENDFUNCTION.

```

New Table for Brazil

New 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 screen-shot below. This table can be maintained with the transaction code /IDT/D_TXJCD_DEF.

| MANDT | LAND1 | LENG1 | LENG2 | LENG3 | LENG4 | TEXT1 |
|-------|-------|-------|-------|-------|-------|---------------------------------|
| 400 | BR | 3 | 7 | | | Tax Jurd. Definition for Brazil |

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. 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**.

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. 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.

| Change View "Publish&Subscribe BTE: Customer Enhanc | | | | | | | |
|---|----------|---------|----------|-----|-----|-------|-------------------------|
| | Event | Partner | Product | N.. | Ctr | Appl. | Function Module |
| | 00001030 | /IDT/ | /IDT/TAX | | | | /IDT/BTE_EVENT_00001030 |
| | 00001050 | /IDT/ | /IDT/TAX | | | | /IDT/BTE_EVENT_00001050 |

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** 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.

| Maintain Intervals: Log counter | | | | | |
|---------------------------------|-----|----------|-----------|-----------|--------------------------|
| | N.. | From No. | To Number | NR Status | Ext |
| | 01 | 00000001 | 99999999 | | <input type="checkbox"/> |

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. This new transmission process will no longer utilize the RFC based SAP Standard External Tax Interface.

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.



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.

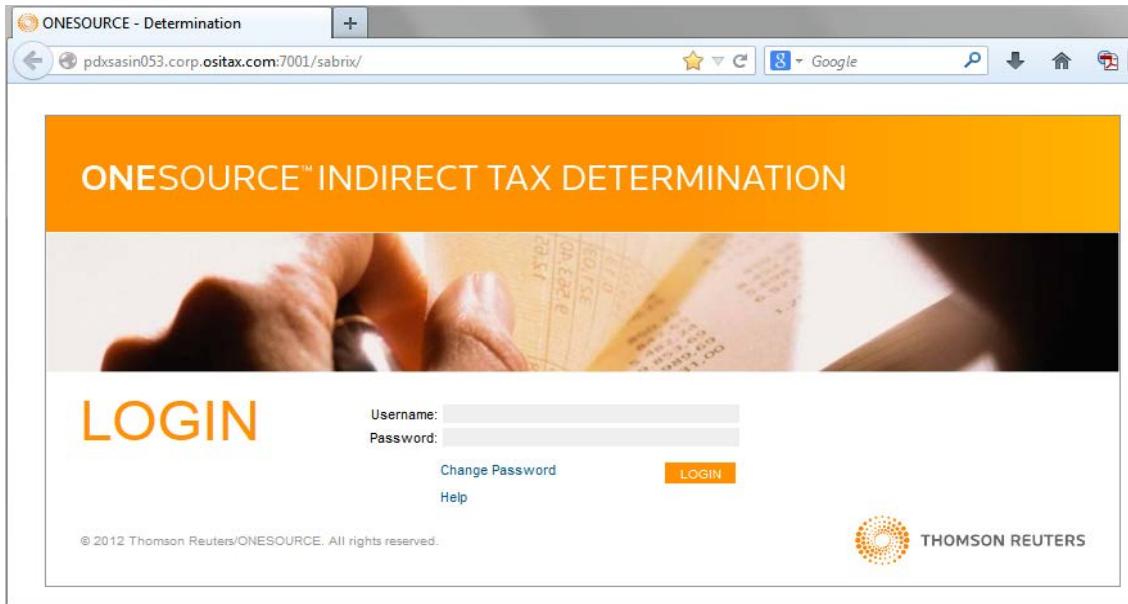
This KB article includes information on:

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.

ACCESSING 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.

Image of the Login screen.



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).



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

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



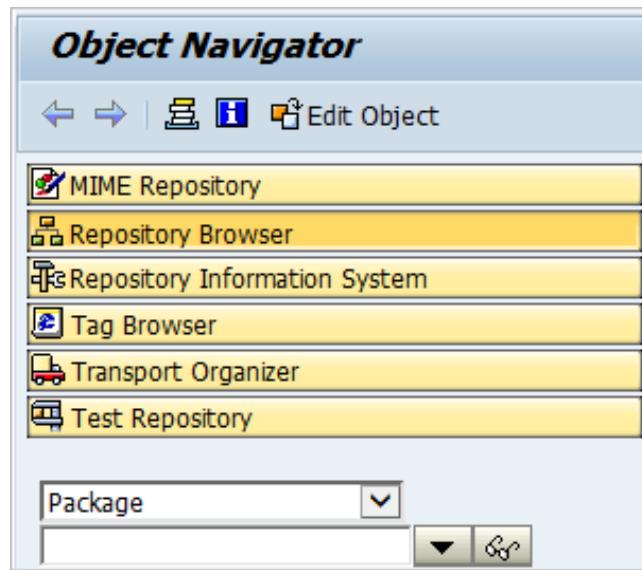
Screens and steps shown in below illustrations are mainly from an SAP ECC 6.0 EHP6. Due to SAP's frequent changes in this area of the system, your specific screens might look slightly different.

CREATING AN SAP PROXY

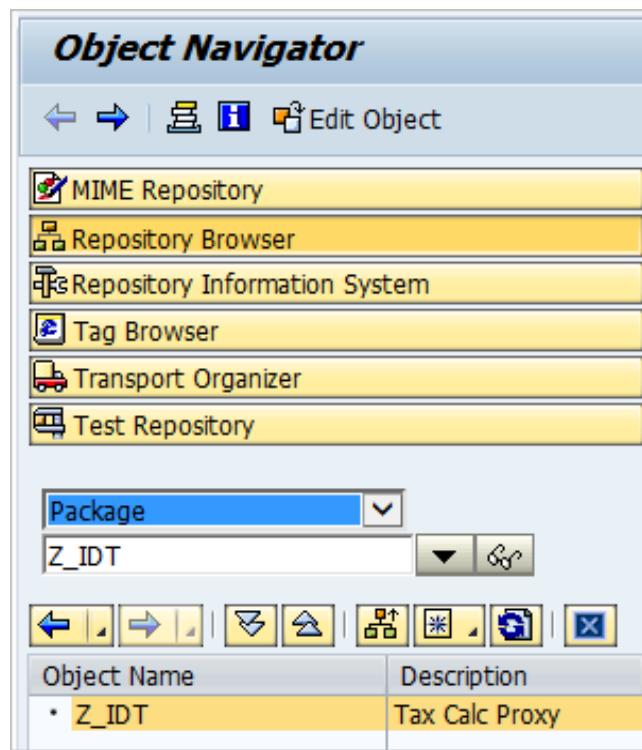
The steps below require a developer's key and access to the correction and transport system.

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

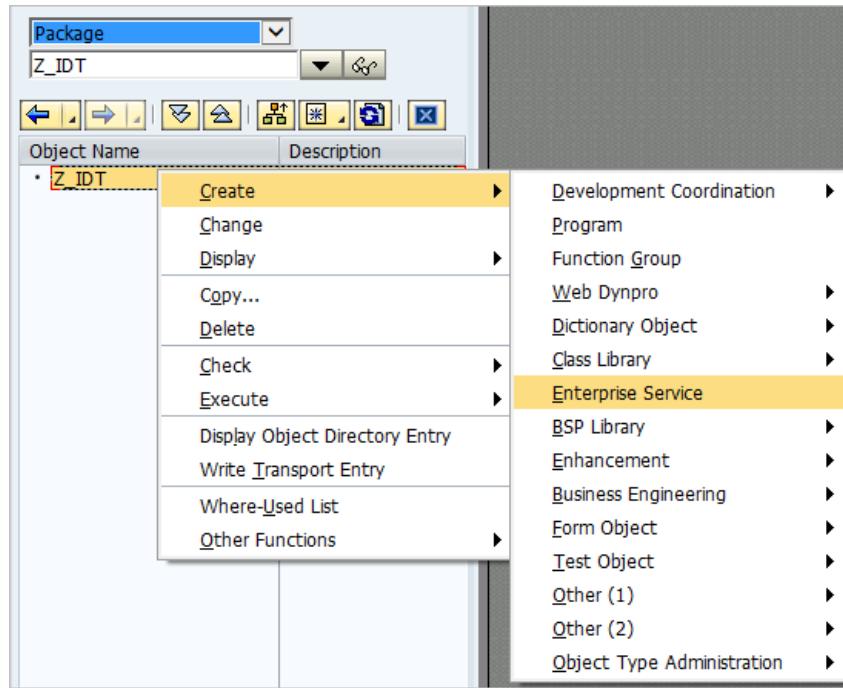
Image of the Package setting.



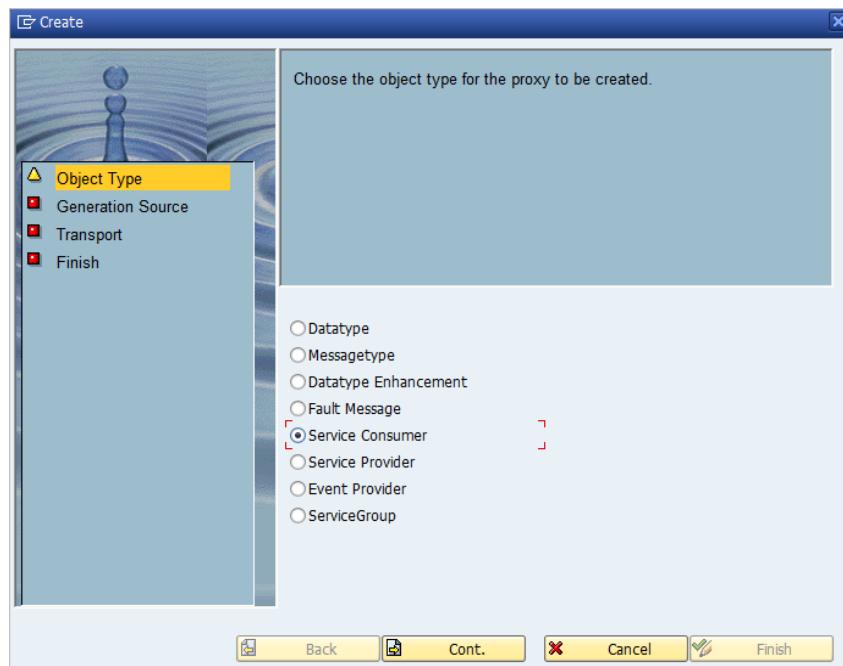
2. Enter the package you want the proxy to be in and press enter. (You may need to create the package at this point and add it to a transport.)



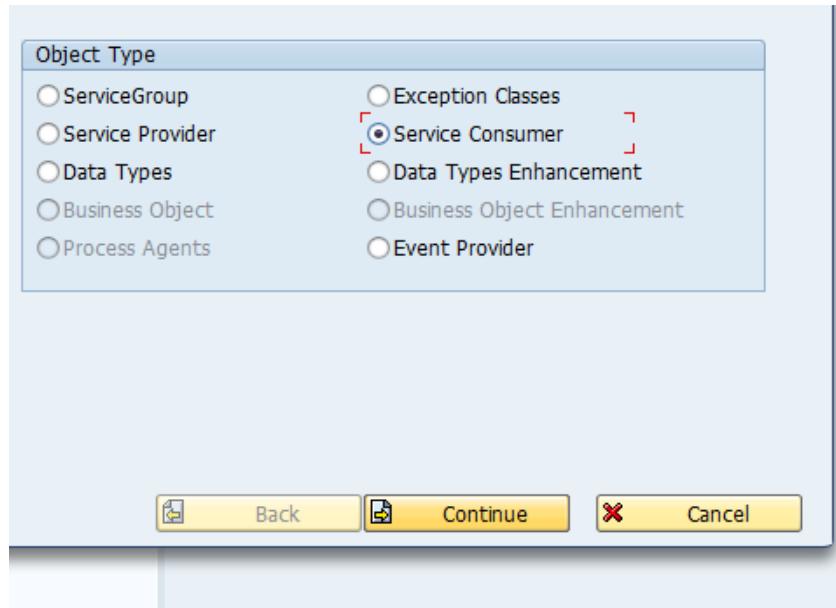
3. Select the package root and right-click. Select Create->Enterprise Service.



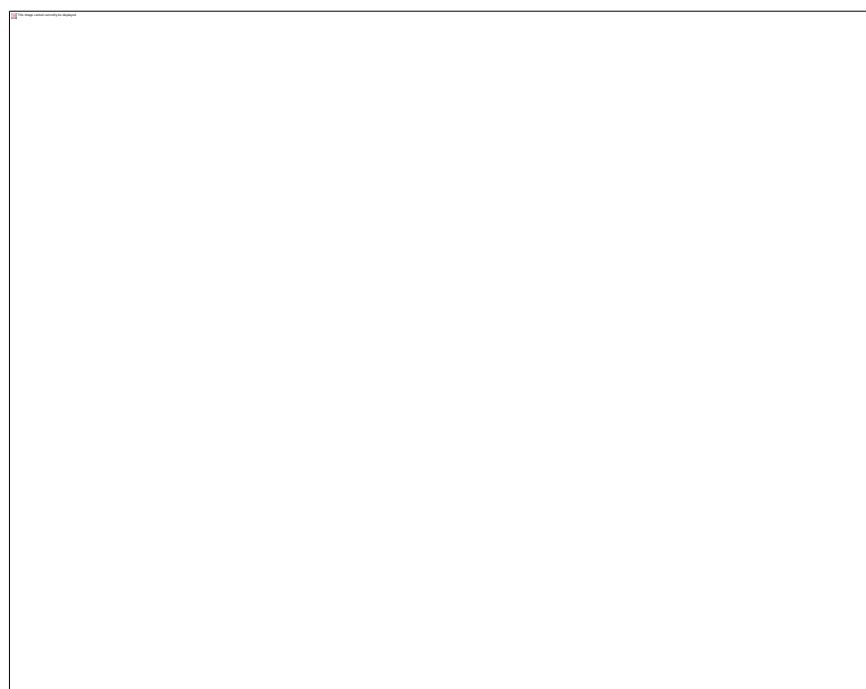
This should take you to something like below picture (this might be slightly different depending on the NetWeaver version used).

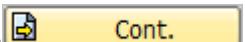


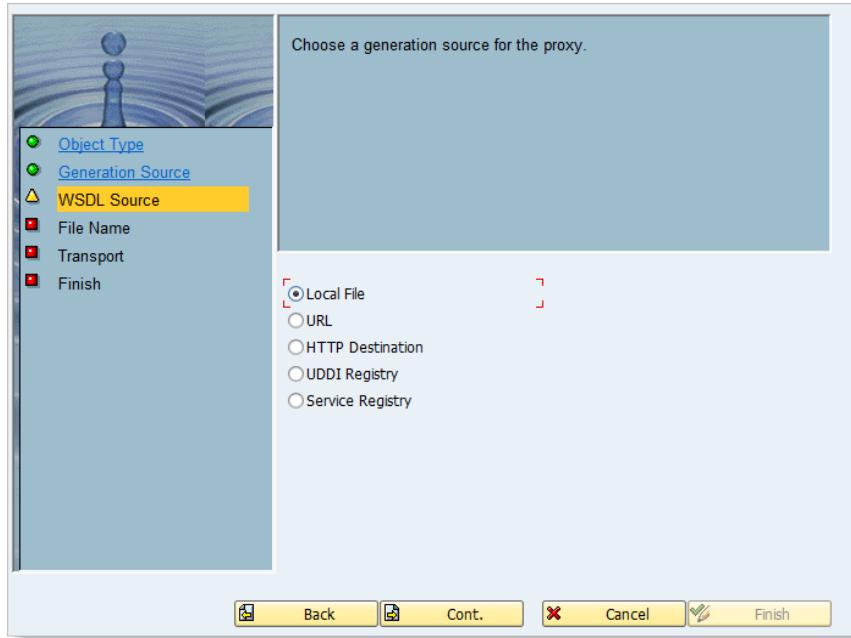
4. Select “Service Consumer”.

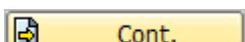


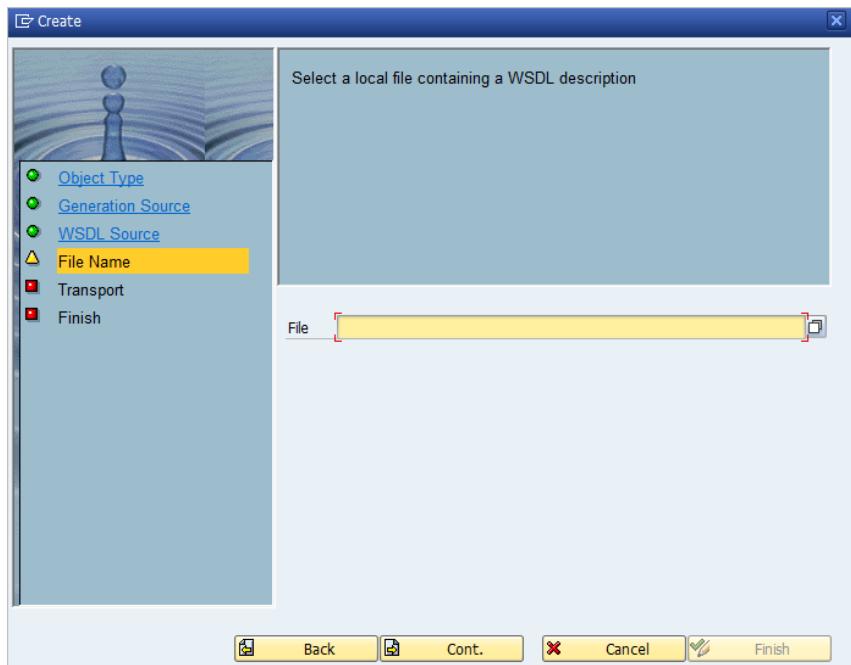
5. Then press  which should take you to something like the screen below.



6. Select **external WSDL** and press  **Cont.**, then select “Local File”.

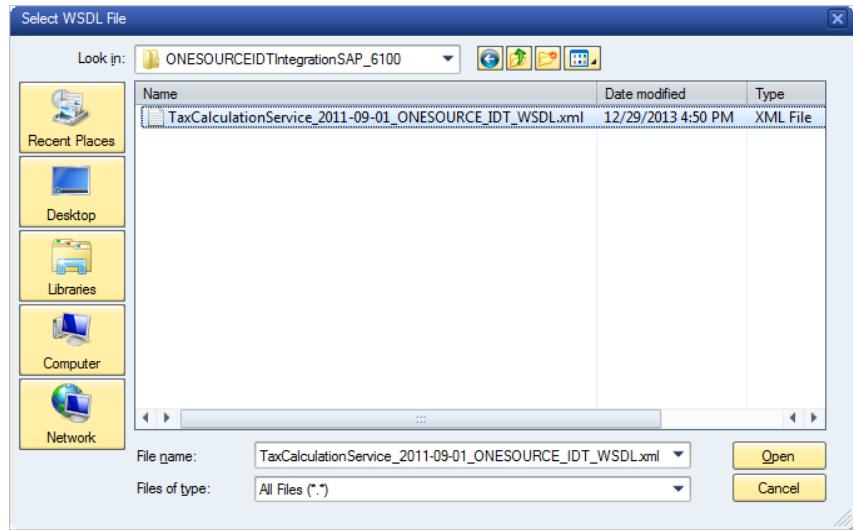


7. Then press  **Cont.** which displays a file selection option.

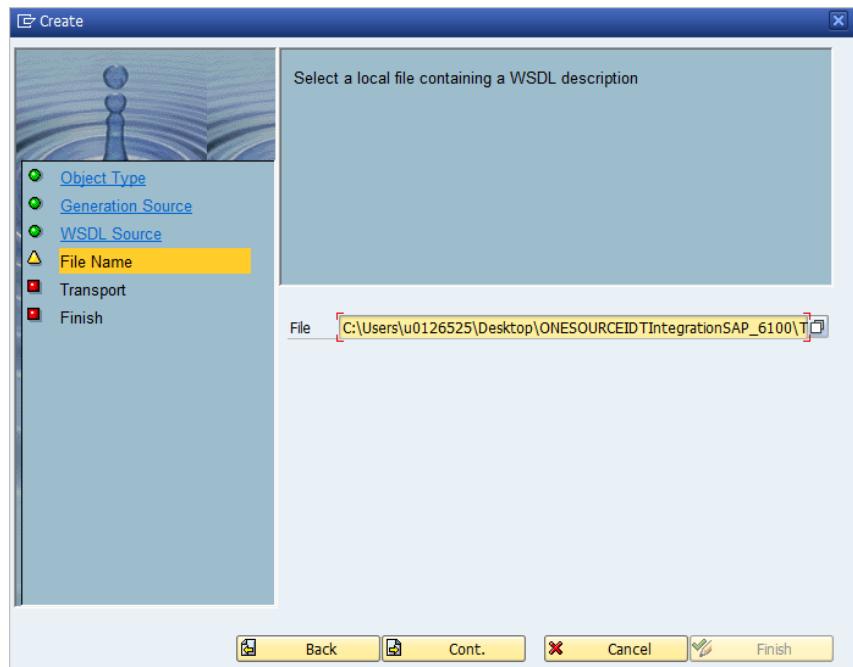


Setting Up the SOAP Interface Proxy

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.



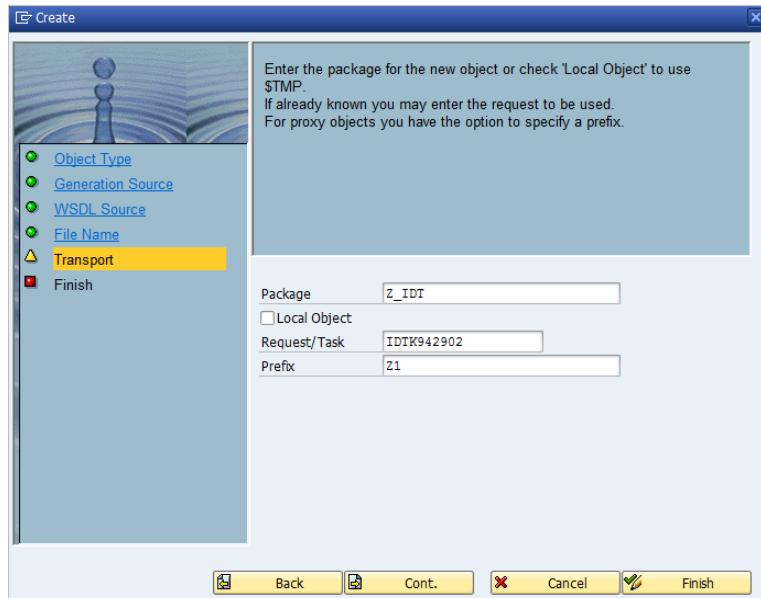
9. Press "Open" to get the file selected.



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

Setting Up the SOAP Interface Proxy

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.).



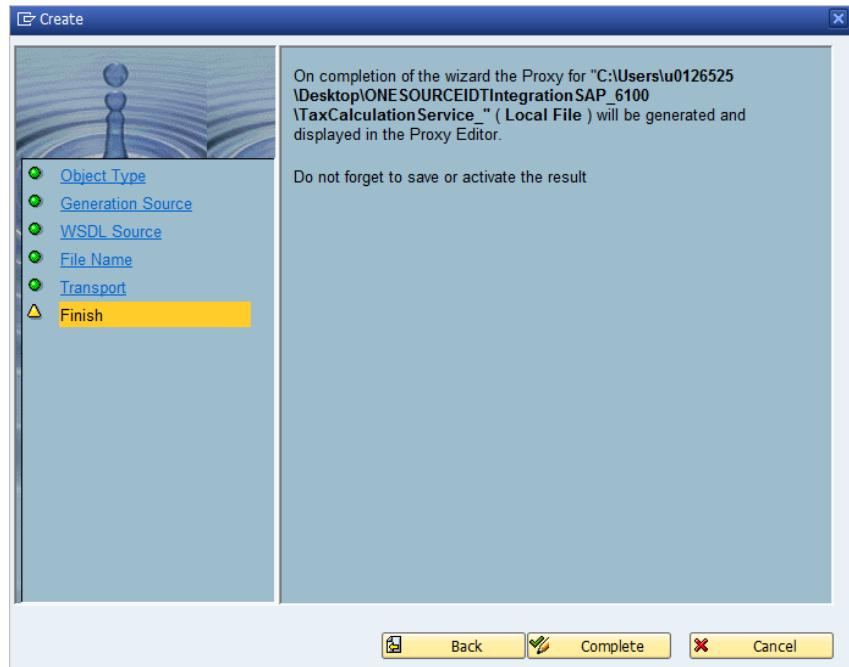
Important: Do not use more than 4 characters in the prefix! What we have found is that using more than four characters may screw up the proxy and you start getting strange and incorrect XML log line identifiers within the XML logs. Errors may occur if you use more characters.

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



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. This table is documented below as needing special adjustment if your prefix differs from our example. Click on link below for structure table modification required. [Structure Table Modify](#)

Setting Up the SOAP Interface Proxy



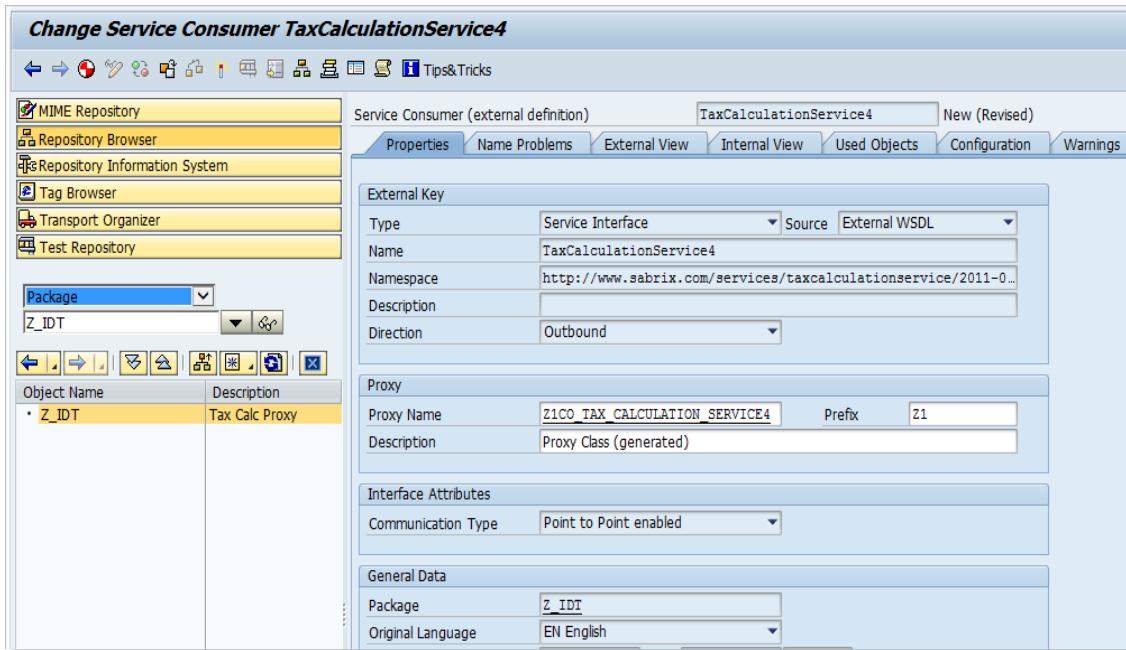
13. Select  to finish setting up the proxy.



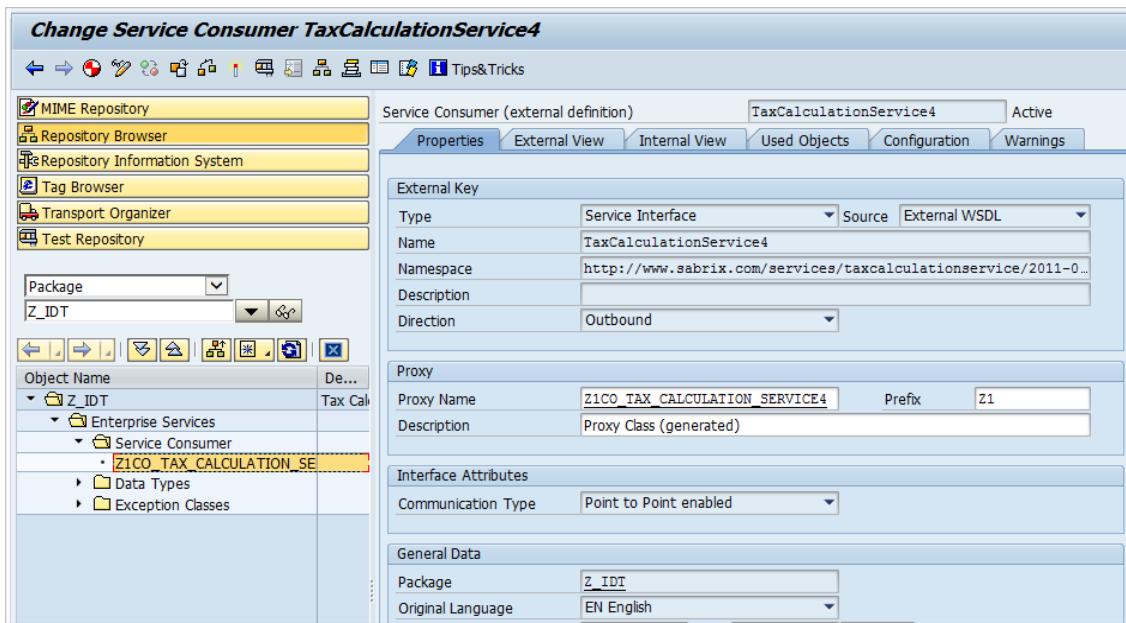
You might receive a warning message by SAP that the system is trying to access a local file, confirm and proceed.

Setting Up the SOAP Interface Proxy

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*).

SETTING 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**

1. Logon to the web portal.

SOA Management (IDT;400)

Service Administration Technical Administration Logs and Traces Management

[Web Service Configuration](#)
Configure Service Definitions, Consumer Proxies and Service Groups

[Simplified Web Service Configuration](#)
Configure Service Definitions for Web Service Consumers with limited capabilities

[Business Scenario Configuration](#)
Configure multiple Service Definitions and Service Groups supporting Change Management

[Logon Data Management](#)
Maintain logon data used by Business Scenario Configuration

[Pending Tasks](#)
Process Pending Tasks generated by Business Scenario Configuration

[Logical Receiver Determination](#)
Maintain rules for finding Business Applications at runtime based on the business context

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

Web Service Configuration (IDT;400)

Search Design Time object for Web Service Configuration

Search Browse

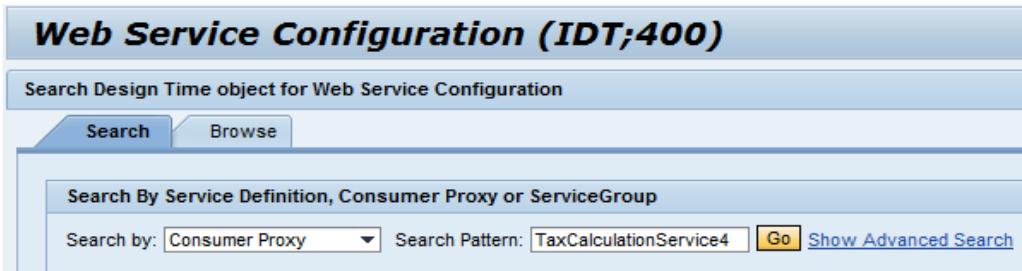
Search By Service Definition, Consumer Proxy or ServiceGroup

Search by: Service Definition Search Pattern: [Show Advanced Search](#)

Search Results

| Internal Name | External Namespace |
|---------------|--------------------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

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**.



Web Service Configuration (IDT;400)

Search Design Time object for Web Service Configuration

Search Browse

Search By Service Definition, Consumer Proxy or ServiceGroup

Search by: Consumer Proxy Search Pattern: TaxCalculationService4 Go Show Advanced Search

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



| Search Results | | | | |
|-------------------------------|---|------------------------|----------------|-------------|
| Internal Name | External Namespace | External Name | Type | Description |
| Z1CO_TAX_CALCULATION_SERVICE4 | http://www.sabrix.com/services/taxcalculationservice/2011-09-01 | TaxCalculationService4 | Consumer Proxy | |

Setting Up the SOAP Interface Proxy

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

Web Service Configuration (IDT;400)

Search Design Time object for Web Service Configuration

Search Browse

Search By Service Definition, Consumer Proxy or ServiceGroup

Search by: Consumer Proxy Search Pattern: TaxCalculationService4 Go Show Advanced Search

Search Results

| Internal Name | External Namespace |
|---------------------------------|---|
| • Z1CO_TAX_CALCULATION_SERVICE4 | http://www.sabrix.com/services/taxcalculationservice/2011-09-01 |
| | |
| | |
| | |
| | |

Apply Selection

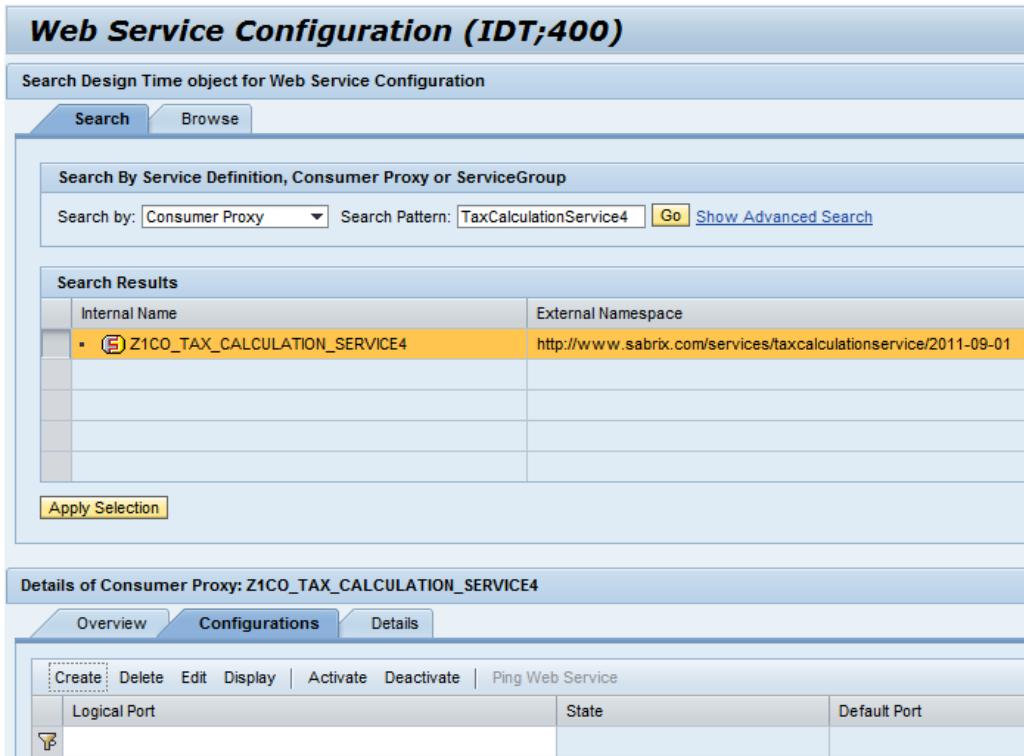
Details of Consumer Proxy: Z1CO_TAX_CALCULATION_SERVICE4

Overview Configurations Details

General Attributes

| | |
|------------------------|---|
| Configuration Status: | Logical Ports: 0 |
| External Namespace: | http://www.sabrix.com/services/taxcalculationservice/2011-09-01 |
| External Name: | TaxCalculationService4 |
| Internal Name: | Z1CO_TAX_CALCULATION_SERVICE4 |
| SOAP Application: | urn:sap-com:soap:application:esr:client:710 |
| Package Name: | Z_IDT |
| Software Component: | HOME |
| Application Component: | |

5. Move to the **Configurations** tab.



The screenshot shows the 'Web Service Configuration (IDT;400)' interface. In the search bar, 'Consumer Proxy' is selected and 'TaxCalculationService4' is entered in the search pattern. The search results table shows one entry: 'Z1CO_TAX_CALCULATION_SERVICE4' with an external namespace of 'http://www.sabrix.com/services/taxcalculationservice/2011-09-01'. An 'Apply Selection' button is at the bottom of the table.

Details of Consumer Proxy: Z1CO_TAX_CALCULATION_SERVICE4

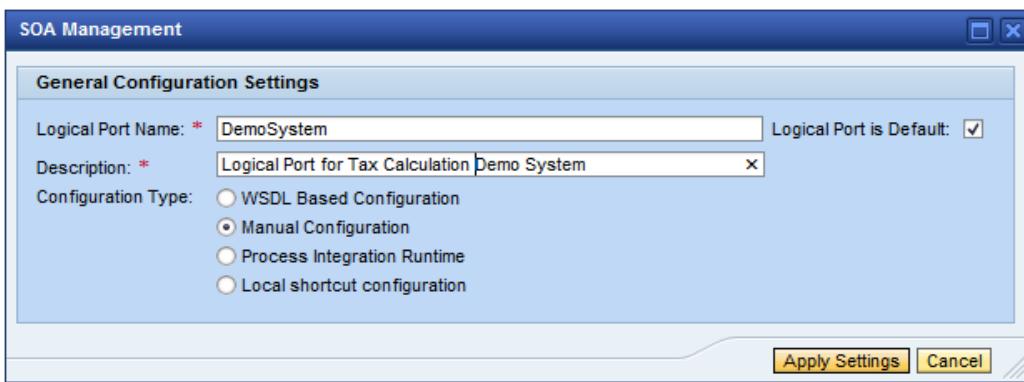
Overview Configurations Details

Create Delete Edit Display Activate Deactivate Ping Web Service

| Logical Port | State | Default Port |
|--------------|-------|--------------|
| | | |

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**



SOA Management

General Configuration Settings

Logical Port Name: * Logical Port is Default:

Description: *

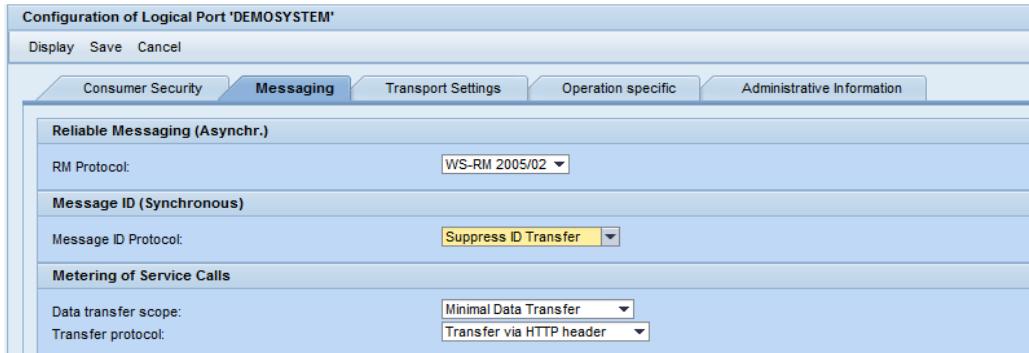
Configuration Type:

- WSDL Based Configuration
- Manual Configuration
- Process Integration Runtime
- Local shortcut configuration

Apply Settings Cancel

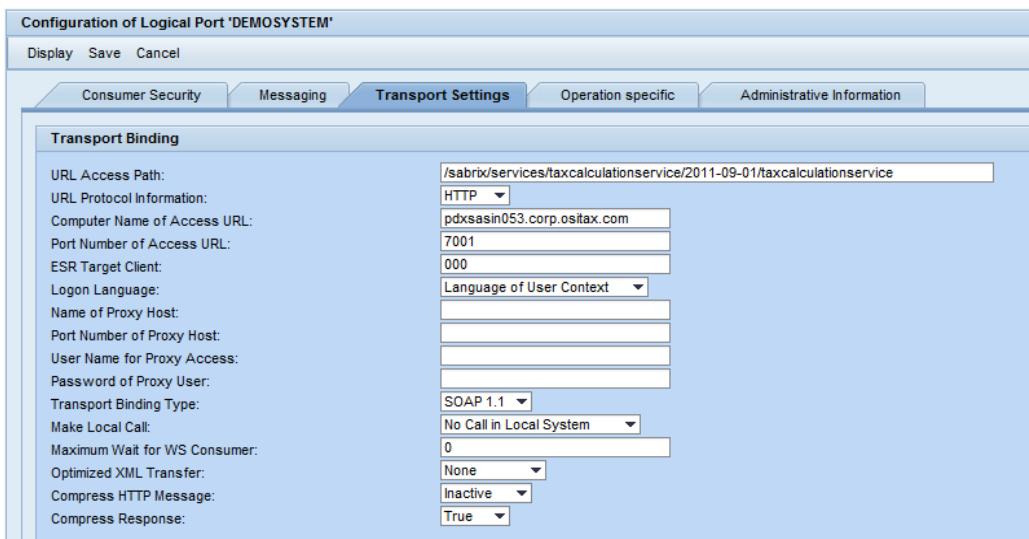
Setting Up the SOAP Interface Proxy

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>



WS Security Considerations on the Proxy for a Hosted Environment

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

The screenshot shows the SAP Web Service Configuration interface. At the top, a search bar is used to find a consumer proxy named 'TaxCalculationService4'. The search results table shows one entry: 'ZICO_TAX_CALCULATION_SERVICE4' with an external namespace 'http://www.sabrix.com/services/taxcalculationservice2011-09-01', an external name 'TaxCalculationService4', and a type 'Consumer Proxy'. Below this, the 'Details of Consumer Proxy: ZICO_TAX_CALCULATION_SERVICE4' screen is displayed, showing a logical port named 'DEMOSYSTEM' in an active state. The 'Transport Settings' tab is selected, showing the transport binding configuration. The URL access path is set to '/sabrix/services/taxcalculationservice/2011-09-01/taxcalculationservice', the protocol is 'HTTP', the computer name is 'pdxxas053.corp.ostax.com', and the port number is '7001'.

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 BAdl 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 new BAdl for your specific security needs. If you wish to utilize GZIP for the compression of the HTTP message 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:

Configuration of Logical Port 'USBTAXENGINE'

Display Save Cancel

Consumer Security Messaging Transport Settings Operation specific Administrative Information

Transport Binding

URL Access Path: /sabrix/services/taxcalculationservice/2011-09-01/taxcalculationservice

URL Protocol Information: HTTP

Computer Name of Access URL: 10.212.144.118

Port Number of Access URL: 6500

Logon Language: Language of User Context

Name of Proxy Host:

Port Number of Proxy Host:

User Name for Proxy Access:

Password of Proxy User:

Transport Binding Type: SOAP 1.1

Make Local Call: No Call in Local System

Maximum Wait for WS Consumer: 0

Optimized XML Transfer: None

Compress HTTP Message: Active

Compress Response: True

Option 2: Using the compression only on the message

Example of screen shot is shown below:

Configuration of Logical Port 'USBTAXENGINE'

Display Save Cancel

Consumer Security Messaging Transport Settings Operation specific Administrative Information

Transport Binding

URL Access Path: /sabrix/services/taxcalculationservice/2011-09-01/taxcalculationservice

URL Protocol Information: HTTP

Computer Name of Access URL: 10.212.144.118

Port Number of Access URL: 6500

Logon Language: Language of User Context

Name of Proxy Host:

Port Number of Proxy Host:

User Name for Proxy Access:

Password of Proxy User:

Transport Binding Type: SOAP 1.1

Make Local Call: No Call in Local System

Maximum Wait for WS Consumer: 0

Optimized XML Transfer: None

Compress HTTP Message: Active

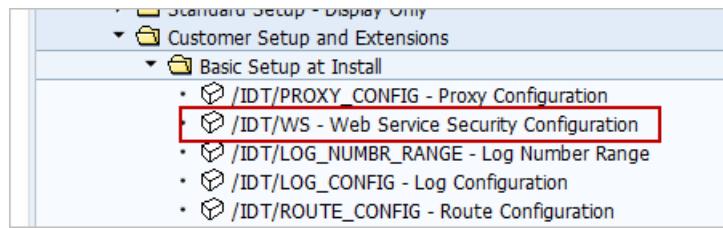
Compress Response: False

A new transaction code: **/N/IDT/WS** has also been created in order for the user to establish the special user name 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 new Security BAdl on the proxy. If the table is not populated then that BAdl 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.

| Display View "Configuration for Proxy Call": Overview | | | | | | | | |
|---|-------------------------------------|-----------|------|------|-------------------|-------------------------------|---------------|----------------------------------|
| Sort ... | ... | User Name | C... | CoCd | Application Se... | Proxy Class | Proxy Method | Logical Port |
| 100010 | <input checked="" type="checkbox"/> | * | * | * | * | Z6CO_TAX_CALCULATION_SERVICE2 | CALCULATE_TAX | DEFAULT |
| 100050 | <input type="checkbox"/> | * | * | * | * | Z6CO_TAX_CALCULATION_SERVICE2 | CALCULATE_TAX | JAKEDESKTOP |
| 100060 | <input type="checkbox"/> | SAPADMIN | * | * | * | Z6CO_TAX_CALCULATION_SERVICE2 | CALCULATE_TAX | PDXSASIN081.CORP.OSITAX.COM:7003 |
| 100070 | <input type="checkbox"/> | * | * | * | * | Z6CO_TAX_CALCULATION_SERVICE2 | CALCULATE_TAX | PDXSASIN082.CORP.OSITAX.COM:7003 |
| 100080 | <input checked="" type="checkbox"/> | JAKET | * | * | * | Z6CO_TAX_CALCULATION_SERVICE2 | CALCULATE_TAX | PDXSASIN082.CORP.OSITAX.COM:7003 |

Note the sort order field in the table as you will need to use the new transaction code to insert a new user name 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 USER NAME and PASSWORD that the system will use to verify the connection. **EXECUTE** to save the new setup to table /IDT/D_WS. Since the USER NAME 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 SET UP OR CUSTOM JOURNEY CREATION

The table /IDT/D_STRUCTS is supplied with standard configuration as part of the configuration transport to install or upgrade Integration. Within it there are line items that reference the proxy and they must be adjusted to the correct prefix in the absolute type name based on what you used for the prefix of your proxy. In our proxy set up instructions we recommend using Z1 as the prefix for the proxy name. 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 example below) and update with your proxy 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.

| Display View "Data Dictionary objects for Journey source and target structure" | | | |
|--|-----------|------------------------|----------|
| Journey Name | Structure | Absolute type name | Src/Trgt |
| /IDT/JOURNEY_AUDIT_UPD_DB_BILL DET_INVOICE | | Z1INDATA_INVOICE_TYPE1 | Target |
| /IDT/JOURNEY_AUDIT_UPD_DB_BILL DET_LINE | | Z1INDATA_LINE_TYPE1 | 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_TYPE1 | Target |
| /IDT/JOURNEY_AUDIT_UPD_DB_GL DET_LINE | | Z1INDATA_LINE_TYPE1 | 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_TYPE1 | Target |
| /IDT/JOURNEY_FB05_COMPANY_ROLE DET_INVOICE | | Z1INDATA_INVOICE_TYPE1 | Target |
| /IDT/JOURNEY_FB05_COMPANY_ROLE SAP_HEADER | | /IDT/SAP_HEADER | Source |

UPDATES TO CUSTOM TABLES

CONFIGURATION DETERMINE_CONDITION_TYPE_FOR_TAXES



Once you populate this table with a new installation you will have to check it with every new release update to the product that you apply to make sure that any new routes get added if new routes are established within the release being applied.

Example: in 6.4.1.0 release two new routes were added:

```
/IDT/ROUTE_NON_GROUP_DOC_A_GL  
/IDT/ROUTE_UPDATE_AUDIT_DB
```

Three lines for each new route would have to be added for this change in 6.4.1.0: one for fee, percentage, and exempt statuses for Nature of Tax along with the correct condition type.

SOAP TAX CALCULATION TEST

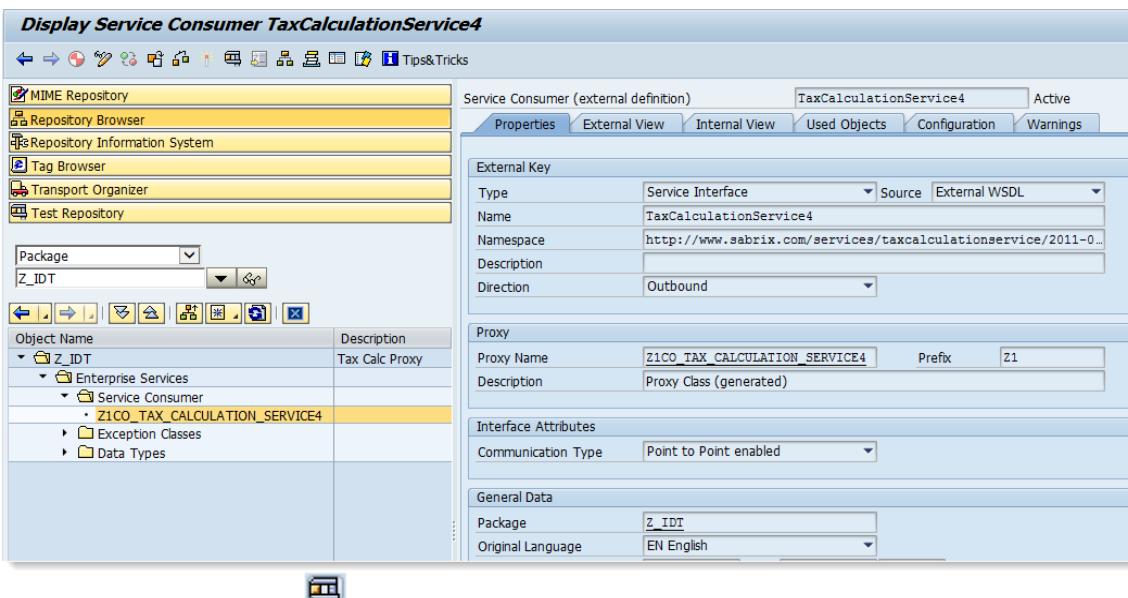
In this section we will make a SOAP test call to Determination using the Proxy we setup above. This will confirm communication between SAP and Determination. It is a prerequisite for a later successful tax calculation from SAP business transactions that the communication layer between the two systems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

SOAP TESTING

Transaction Code **SE80**

SOAP Tax Calculation Test

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.

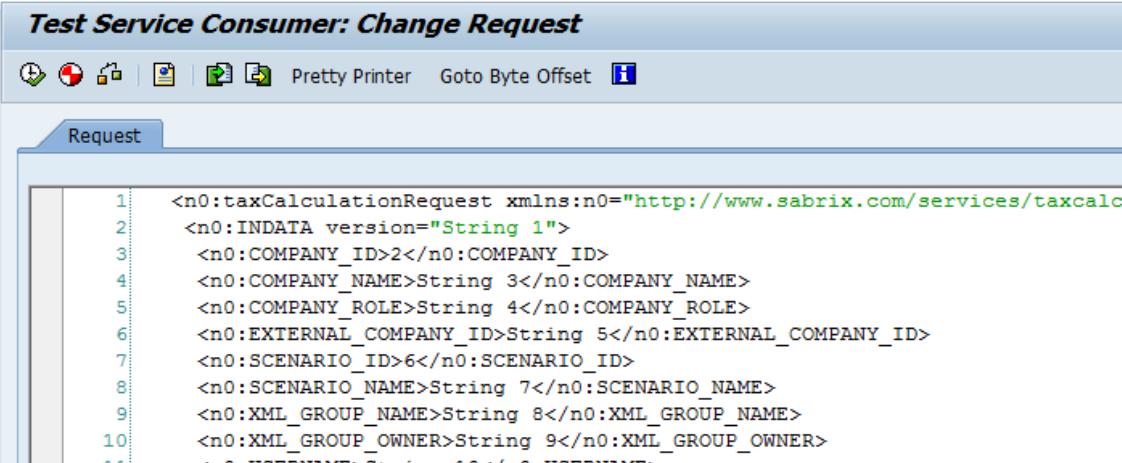


2. Press the **Test** icon . On the pop-up select the **Logical Port** you created in prior steps.
3. Select the **Execute** icon which will take you to the Test Service Consumer screen.



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

Test Service Consumer: Change Request



The screenshot shows the SAP Test Service Consumer interface. The title bar reads "Test Service Consumer: Change Request". The toolbar includes icons for copy, paste, find, print, pretty printer, goto byte offset, and help. The main area is titled "Request" and contains an XML code editor. The XML code is as follows:

```
1 <n0:taxCalculationRequest xmlns:n0="http://www.sabrix.com/services/taxcalc
2 <n0:INDATA version="String 1">
3 <n0:COMPANY_ID>2</n0:COMPANY_ID>
4 <n0:COMPANY_NAME>String 3</n0:COMPANY_NAME>
5 <n0:COMPANY_ROLE>String 4</n0:COMPANY_ROLE>
6 <n0:EXTERNAL_COMPANY_ID>String 5</n0:EXTERNAL_COMPANY_ID>
7 <n0:SCENARIO_ID>6</n0:SCENARIO_ID>
8 <n0:SCENARIO_NAME>String 7</n0:SCENARIO_NAME>
9 <n0:XML_GROUP_NAME>String 8</n0:XML_GROUP_NAME>
10 <n0:XML_GROUP_OWNER>String 9</n0:XML_GROUP_OWNER>
```

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:INDATA version="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:LINE_ID="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).

Test Service Consumer: Change Request

Pretty Printer Goto Byte Offset [Help](#)

Request

```
1> <taxCalculationRequest xmlns="http://www.sabrix.com/services/taxcalculation
2>   <INDATA version="G">
3>     <INVOICE>
4>       <CALCULATION_DIRECTION>F</CALCULATION_DIRECTION>
5>       <EXTERNAL_COMPANY_ID>3000</EXTERNAL_COMPANY_ID>
6>       <COMPANY_NAME>Products</COMPANY_NAME>
7>       <COMPANY_ROLE>S</COMPANY_ROLE>
8>       <CURRENCY_CODE>USD</CURRENCY_CODE>
9>       <INVOICE_DATE>2011-11-09</INVOICE_DATE>
10>      <IS_AUDITED>false</IS_AUDITED>
11>      <IS_REPORTED>false</IS_REPORTED>
12>      <IS_REVERSED>false</IS_REVERSED>
13>      <POINT_OF_TITLE_TRANSFER>I</POINT_OF_TITLE_TRANSFER>
14>      <SHIP_TO>
15>        <COUNTRY>US</COUNTRY>
16>        <STATE>OR</STATE>
17>        <CITY>PORTLAND</CITY>
18>        <POSTCODE>97214</POSTCODE>
19>      </SHIP_TO>
20>      <TRANSACTION_TYPE>GS</TRANSACTION_TYPE>
21>      <LINE ID="1">
22>        <GROSS_AMOUNT>1111</GROSS_AMOUNT>
23>        <LINE_NUMBER>1</LINE_NUMBER>
24>        <QUANTITIES>
25>          <QUANTITY>
26>            <AMOUNT>1</AMOUNT>
27>            <UOM>each</UOM>
28>          </QUANTITY>
29>        </QUANTITIES>
30>      </LINE>
31>    </INVOICE>
32>  </INDATA>
33> </taxCalculationRequest>
```



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 .

SOAP Tax Calculation Test

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

```

<n0:taxCalculationResponse xmlns:n0="http://www.sabrix.com/services/taxcalculationservi
- <n0:OUTDATA version="G">
- <n0:REQUEST_STATUS>
  <n0:IS_SUCCESS>true</n0:IS_SUCCESS>
  <n0:IS_PARTIAL_SUCCESS>true</n0:IS_PARTIAL_SUCCESS>
</n0:REQUEST_STATUS>
- <n0:INVOICE>
- <n0:REQUEST_STATUS>
  <n0:IS_SUCCESS>true</n0:IS_SUCCESS>
  <n0:IS_PARTIAL_SUCCESS>false</n0:IS_PARTIAL_SUCCESS>
</n0:REQUEST_STATUS>
<n0:CALCULATION_DIRECTION>F</n0:CALCULATION_DIRECTION>
<n0:COMPANY_ID>203</n0:COMPANY_ID>
<n0:COMPANY_NAME>SAP US INC 3000</n0:COMPANY_NAME>
<n0:COMPANY_ROLE>C</n0:COMPANY_ROLE>

```

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.

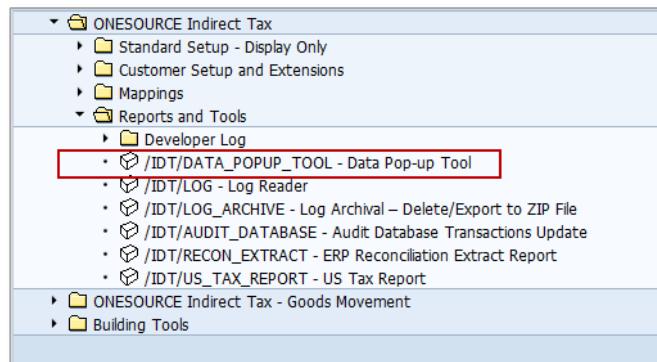
DATA POPUP TOOL

The Data Popup Tool is a new 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 journey 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**



Data Popup Tool

Actual Data Pop-up Control Configuration table.

| Actual Data Pop-Up Control Configuration Table | | | |
|--|--------------------------------|-----------|-------------------------------------|
| SAP ... | Journey Name | User Name | Active |
| D03 | /IDT/JOURNEY_HEADER_REQUEST | JAKET | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_HEADER_REQUEST | JENNIFERH | <input checked="" type="checkbox"/> |
| D03 | /IDT/JOURNEY_HEADER_REQUEST | NEELIMAK | <input checked="" type="checkbox"/> |
| D03 | /IDT/JOURNEY_HEADER_REQUEST | RICKP | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_ITEM_REQUEST | JAKET | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_ITEM_REQUEST | NEELIMAK | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_ITEM_REQUEST | RICKP | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_NG_ITEM_REQUEST | JAKET | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_NG_ITEM_REQUEST | NEELIMAK | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_NG_ITEM_REQUEST | RICKP | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_STANDARD_RESPONSE | JAKET | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_STANDARD_RESPONSE | RICKP | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_TAX_DATA_RESPONSE | JAKET | <input type="checkbox"/> |
| D03 | /IDT/JOURNEY_TAX_DATA_RESPONSE | RICKP | <input type="checkbox"/> |

Once in the screen you will be able to change it to change mode and add lines to the table for your user name 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**

Data Popup Tool

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.

| Component | Value | Values Present | Short Description |
|-------------|-------|----------------|---|
| KONK | X | | Communication Header for Pricing |
| BKPF | X | | Accounting Document Header |
| VBAK | | | Sales Document: Header Data |
| VBRK | | | Billing Document: Header Data |
| EKKO | | | Purchasing Document Header |
| RBKP | | | Document Header: Invoice Receipt |
| T001 | | | Company Codes |
| T005 | | | Countries |
| T180 | | | Screen Sequence Control: Transaction Default Values |
| VBDK | | | Sales Document: Business Data |
| TVFK | | | Billing: Document Types |
| TVTA | | | Organizational Unit: Sales Area(s) |
| TVAK | | | Sales Document: Types |
| TVKO | | | Organizational Unit: Sales Organizations |
| VBUK | | | Sales Document: Header Status and Administrative Data |
| KNA1 | | | General Data in Customer Master |
| LFA1 | | | Vendor Master (General Section) |
| LFM1 | | | Vendor master record purchasing organization data |
| T001W | | | Plants/Branches |
| WEDATEN | | | Data for Price Determination at Time of Goods Receipt |
| CALC_HDR | X | | Calculated header |
| PARTNER_TAB | | | |
| ITEMS | X | | |
| J_1BBRANCH | | | Business Place |
| J_1IMOVEND | | | Vendor Master Excise Additional Data |
| J_1IMOCUST | | | Customer Master Excise Additional Data |
| J_1IMOCOMP | | | Company and Plant - Excise Additional Data |
| T001Z_TAB | X | | |
| ESSR | | | Service Entry Sheet Header Data |
| EKPO | | | Purchasing Document Item |
| LKP | | | SD Document: Delivery Header Data |
| MKPF | | | Header: Material Document |

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.

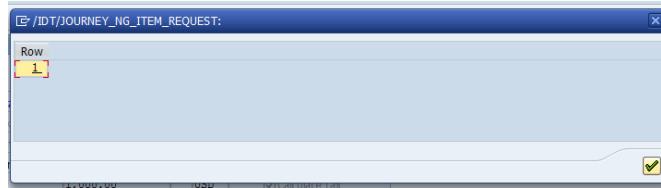
Data Popup Tool

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.

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.

| /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 |
| VKGRO | | | 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) |
| WKREG | | | Region in which plant is located |
| WKCOU | | | County in which plant is located |
| WKCTY | | | City in which plant is located |
| BUKRS | 3000 | X | Company Code |
| GSBER | | | Business Area |
| WERKV | | | Resale plant |
| HWAER | | | Local Currency |
| WAERK | USD | X | SD Document Currency |
| KURST | | | Exchange Rate Type |
| KURRF | 0.00000 | | Exchange rate for FI postings |
| KURRF_DAT | 0000000 | | Translation Date |
| KUNNR | | | Sold-to party |
| KNRZE | | | Payer |
| KUNRE | | | Bill-to party |
| KUNWE | | | Ship-to party |
| SPDNR | | | Forwarding agent |
| LIFNR | 0000003000 | X | Vendor Account Number |
| VRTNR | 00000000 | | Sales employee |
| ZTERM | | | Terms of Payment Key |
| ZBD1P | 0.000 | | Cash Discount Percentage 1 |
| KAPPL | TX | X | Application |
| KALSM | ZTPUS | X | Procedure (Pricing, Output Control, Acct. Det., Costing,...) |
| PRSDT | 20160729 | X | Date for pricing and exchange rate |
| FBUDA | 00000000 | | Date on which services rendered |
| FKDAT | 00000000 | | Billing date for billing index and printout |
| ERDAT | 00000000 | | Date on Which Record Was Created |
| AUDAT | 00000000 | | Document Date (Date Received/Sent) |
| INCO1 | | | Incoterms (Part 1) |
| INCO2 | | | Incoterms (Part 2) |
| LAND1 | | | Country of Destination |
| REGIO | | | Region (State, Province, County) |
| COUNC | | | County Code |
| CITYC | | | City Code |
| TXJCD | | | Tax Jurisdiction |

Example of an item request dialogue.



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.

| Component | Value | Values Present | Short Description |
|-------------|-------|----------------|---|
| HDR | X | | Header for field mapping data |
| KOMK | X | | Communication Header for Pricing |
| BKPF | X | | Accounting Document Header |
| VBAK | | | Sales Document: Header Data |
| VBRK | | | Reference Structure for XVBRAK/YVBRP |
| EKKO | | | Purchasing Document Header |
| RRBKPV | | | MRM_RRBKPV |
| T001 | | | Company Codes |
| T180 | | | Screen Sequence Control: Transaction Default Values |
| TVFK | | | Billing: Document Types |
| TVTA | | | Organizational Unit: Sales Area(s) |
| TVAK | | | Sales Document: Types |
| TVKO | | | Organizational Unit: Sales Organizations |
| VBUK | | | Reference structure for XVBUK/YVBUK |
| LFA1 | | | Vendor Master (General Section) |
| KNA1 | | | General Data in Customer Master |
| HDR_VBKD | | | Reference structure for XVBKD/YVBKD |
| BSEG | X | | Accounting Document Segment |
| CALC_ITEM | X | | IDT Item Calculated Fields |
| KOMP | X | | Communication Item for Pricing |
| EKPO | | | Purchasing Document Item |
| PRICE_COND | | | KOMV structure plus index |
| VBAP | | | Sales Document: Item Data |
| VBRP | | | Billing Document: Item Data |
| TVAP | | | Sales Document: Item Categories |
| T001W | | | Plants/Branches |
| ITM_VBKD | | | Reference structure for XVBKD/YVBKD |
| VBUP | | | Reference Structure for XVBUP/YVBUP |
| VBAPF | | | Order Item Flow: Dynamic Division |
| MT06E | | | Material Master Fields: Purchasing |
| EINA | | | Purchasing Info Record: General Data |
| EINE | | | Purchasing Info Record: Purchasing Organization Data |
| EKKNU | | | Work Structure for Account Assignments in Purchasing Doc. |
| MARA | | | General Material Data |
| MARC | | | Plant Data for Material |
| MARD | | | Storage Location Data for Material |
| MBRW | | | Material Valuation |
| MAKT | | | Material Descriptions |
| MVKE | | | Sales Data for Material |
| T163 | | | Item Categories in Purchasing Document |
| DRSEG | | | MMCR_DRSEG structure |
| PARTNER_TAB | | | |
| J_1BBRANCH | | | Business Place |
| J_1IMOCOMP | | | Company and Plant - Excise Additional Data |
| J_1BSDICA | | | Sales Document: Item Category table |
| ESLL | | | Lines of Service Package |
| LIPS | | | SD document: Delivery: Item data |
| CSKS | X | | Cost Center Master Data |
| ESKN | | | Structure for Change Document: Generated by RSSCD000 |
| ESKL | | | Change Document Structure: Generated by RSSCD000 |

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

| /IDT/JOURNEY_STANDARD_RESPONSE: | | | |
|---------------------------------|-------|----------------|--|
| Component | Value | Values Present | Short Description |
| CONTROLLER | | | |
| PARAMETERS | X | | The top level response to a request for tax calculation. |

Data Popup Tool

Parameters screen.

| Component | Value | Values Present | Short Description |
|------------|---------|----------------|-----------------------------|
| CONTROLLER | OUTDATA | X | Proxy Structure (generated) |

Then to the outdata...

| Component name | Value | Values Present | Short Description |
|---------------------|-----------|----------------|--|
| CONTROLLER | | | |
| REQUEST_STATUS | X | | A structure which contains a summary of the status of the e |
| COMPANY_ID | 0 | | The internal Determination ID of the merchant. |
| COMPANY_NAME | ONESOURCE | | The name of the company running the transaction. |
| COMPANY_ROLE | B | X | The role the company plays in a given transaction: Buyer (B) |
| EXTERNAL_COMPANY_ID | 3000 | X | The unique identifier used by your business application to |
| SCENARIO_NAME | | | This is the name of the Determination Workbench scenario. |
| INVOICE | X | | |
| VERSION | G | X | The version of the determination object model schema. |

... and invoice level data...

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

Developer's Log

... and finally back to the transaction..

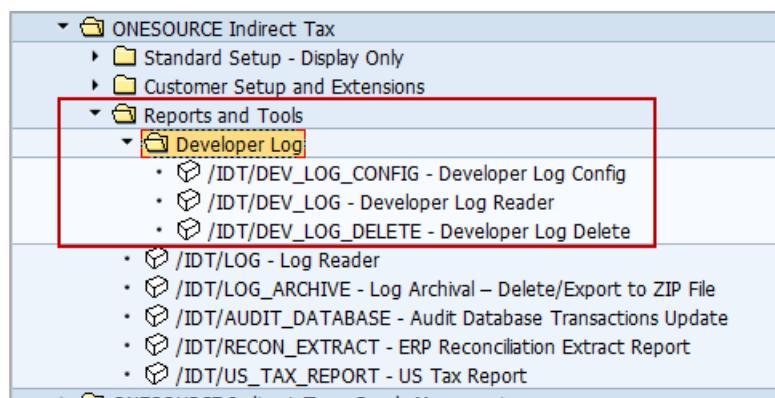
| S... | G/L Acct | Short Text | D/C | Amount in do... | Tax co... | W/o ... | Text | L... | Co... | Cost center | Tradu... |
|--------|------------|------------|----------|-----------------|-----------|---------|------|------|-------|-------------|----------|
| 475000 | Company ca | Debit | 1,000.00 | 11 | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |
| | | Debit | | | | | | 3000 | 3000 | 3000 | |



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 new 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 located 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 also 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**

| Change View "Developer Log Configuration: Custom": Overview | | | | | | | | | | | | |
|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|
| Developer Log Configuration: Custom | | | | | | | | | | | | |
| User Name | Request | Response | Log Error | Route | FI Process | Journey | Hooks | Show Resp | Show Req | Summing | US Logic | |
| CHAITANYAS | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| DEEPUV | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| JAKET | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| JENNIFERH | <input type="checkbox"/> | |
| NEELIMAK | <input checked="" type="checkbox"/> | |
| PRAFULLT | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| RICKP | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

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

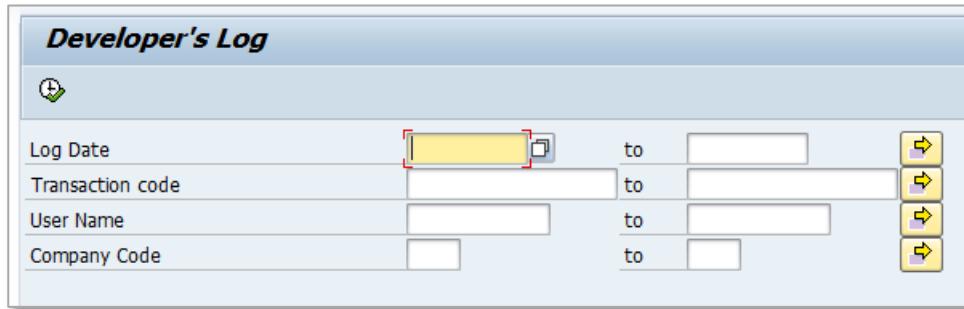
| | |
|---------------|--|
| 1. Request | The request data only will display as a separate log file |
| 2. Response | The response data only will display as a separate log file |
| 3. Log Error | Include error messages |
| 4. Route | Display the routes that were used on the call |
| 5. FI Process | Display 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 |



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

| Developer's Log Reader | | | | | | |
|------------------------|-----------------------|------------------------|-------|------------|----------------------------|-------|
| Client | UUID | Message Type | User | Date | Time Stamp | CoCd |
| | | | | | | TCode |
| 400 | 051MYW5b7k07r03A2h60 | taxCalculationResponse | JAKET | 05/20/2016 | 20:160_520_183_117_6260000 | 3000 |
| 400 | 051MYW5b7k07r03uQYH0 | taxCalculationResponse | JAKET | 05/20/2016 | 20:160_520_183_148_6250000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/20/2016 | 20:160_520_183_208_9070000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/20/2016 | 20:160_520_183_209_0630000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/20/2016 | 20:160_520_193_520_0970000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/20/2016 | 20:160_520_193_520_2530000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/20/2016 | 20:160_520_195_604_1620000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/20/2016 | 20:160_520_195_604_3690000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/18/2016 | 20:160_519_003_928_5530000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/18/2016 | 20:160_519_004_010_9120000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/18/2016 | 20:160_519_003_452_2460000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/18/2016 | 20:160_519_003_455_6210000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/18/2016 | 20:160_519_003_856_6000000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationResponse | JAKET | 05/18/2016 | 20:160_519_003_856_2200000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/18/2016 | 20:160_519_004_112_5050000 | 3000 |
| 400 | 051MYW5b7k07r07kGsd60 | taxCalculationRequest | JAKET | 05/20/2016 | 20:160_520_183_236_6680000 | 3000 |

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

New improvements within the new field mapping feature have been added as of release 6.4.1.0 with the addition of dynamically generated code. Prior to this release the field mapping table logic relied on a different set of code to convert the table mapping entries. This process is a first generation development for the new field mapper logic however, was not efficient for large scale invoices. What was needed was a way to generate different code that was much faster and specifically tailored to the exact mapping line

of the field mapper. In order to do this a new program was written. We are trying to duplicate the performance that can be achieved by coding unique field mapping programs for each customer, but in an automatic process. This new process makes the field mapper extremely efficient for the processing of large scale invoices that contain more than 1,000 lines. Users will now see a speed improvement from 10x to 100x within the field mapping function for large invoices being processed to and from Determination. The new dynamic code function greatly reduces processing time but does not change the user interface or function of the field mapper itself.

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 new 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 in order to keep the system clean. There is a new program that you will need to run in background on a regular basis so as 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.

Performance Improvement with Dynamically Generated Code

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 | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_AUDIT_UPD_DB_GL | 20,160,808,142,631.7790000 | /IDT/GN_051MG3JM7JQNHECNWDN0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_CHECK_AUDIT_MESS | 20,160,808,142,632.4030000 | /IDT/GN_051MG3JM7JQNHECNWDOWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_FB05_COMPANY_ROLE | 20,160,815,141,954.5520000 | /IDT/GN_051MG3JM7JQOTCGQ85PWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_FB05_RESPONSE | 20,160,815,141,955.2860000 | /IDT/GN_051MG3JM7JQOTCGQ85U0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_HEADER_REQUEST | 20,160,808,142,524.1350000 | /IDT/GN_051MG3JM7JQNHDUDJTT0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_ITEM_REQUEST | 20,160,808,142,524.7430000 | /IDT/GN_051MG3JM7JQNHDUDJTV0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_ITEM_REQUEST | 20,160,808,210,601,3340000 | /IDT/GN_051MG3JM7JQNJR0RY0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_ITEM_REQUEST | 20,160,815,180,820.1650000 | /IDT/GN_051MG3JM7JQOUCTN44MWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_FB05 | 20,160,815,141,953.1170000 | /IDT/GN_051MG3JM7JQOTCGQ85N0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_REQUEST | 20,160,808,142,555.3520000 | /IDT/GN_051MG3JM7JQNHE1FALQWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_SERV_ENTR | 20,160,808,153,749.5800000 | /IDT/GN_051MG3JM7JQNHY7GWW00JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_STANDARD_RESPONSE | 20,160,808,142,525.5540000 | /IDT/GN_051MG3JM7JQNHDUDJTZWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_TAX_TAB_RESPONSE | 20,160,808,142,526.1320000 | /IDT/GN_051MG3JM7JQNHDUDJTJWJW | FF0602010102800034313033000000000030E000... |

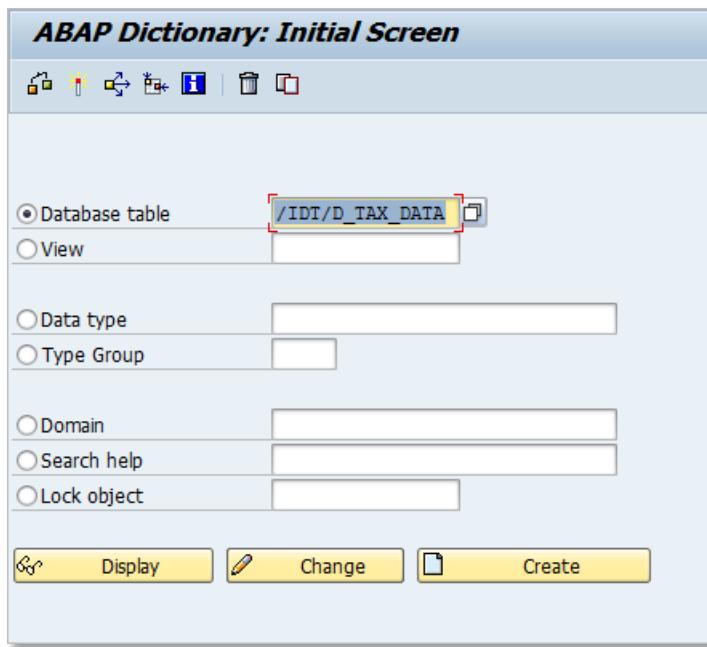
After running the program you can see it is deleting the correct entries and that the most current line for Journey_Item_Request is now the only one on the list:

| 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 | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_AUDIT_UPD_DB_GL | 20,160,808,142,631.7790000 | /IDT/GN_051MG3JM7JQNHECNWDN0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_CHECK_AUDIT_MESS | 20,160,808,142,632.4030000 | /IDT/GN_051MG3JM7JQNHECNWDOWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_FB05_COMPANY_ROLE | 20,160,815,141,954.5520000 | /IDT/GN_051MG3JM7JQOTCGQ85PWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_FB05_RESPONSE | 20,160,815,141,955.2860000 | /IDT/GN_051MG3JM7JQOTCGQ85U0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_HEADER_REQUEST | 20,160,808,142,524.1350000 | /IDT/GN_051MG3JM7JQNHDUDJTT0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_ITEM_REQUEST | 20,160,815,180,820.1650000 | /IDT/GN_051MG3JM7JQOUCTN44MWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_FB05 | 20,160,815,141,953.1170000 | /IDT/GN_051MG3JM7JQOTCGQ85N0JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_REQUEST | 20,160,808,142,555.3520000 | /IDT/GN_051MG3JM7JQNHE1FALQWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_NG_ITEM_SERV_ENTR | 20,160,808,153,749.5800000 | /IDT/GN_051MG3JM7JQNHY7GWW00JW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_STANDARD_RESPONSE | 20,160,808,142,525.5540000 | /IDT/GN_051MG3JM7JQNHDUDJTZWJW | FF0602010102800034313033000000000030E000... |
| 400 | /IDT/JOURNEY_TAX_TAB_RESPONSE | 20,160,808,142,526.1320000 | /IDT/GN_051MG3JM7JQNHDUDJTJWJW | FF0602010102800034313033000000000030E000... |

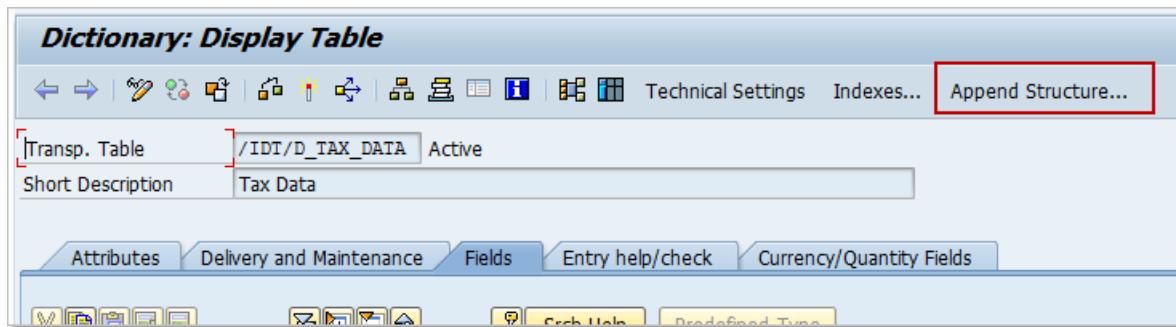
ADDING CUSTOM FIELDS TO /IDT/D_TAX_DATA TABLE

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**

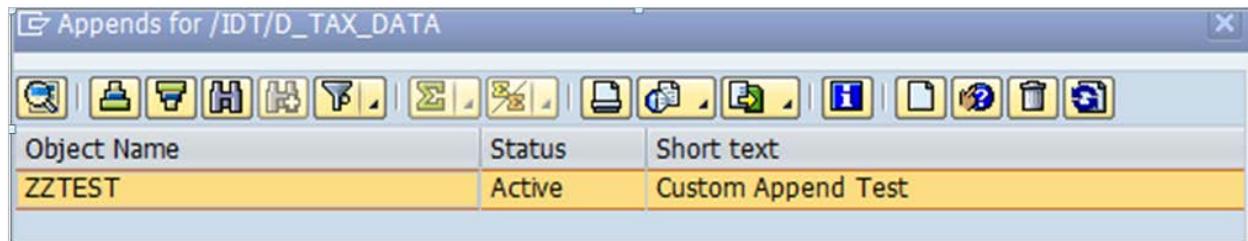


Select "Display". Once in the table, use the "Append Structure (F5)" menu option.

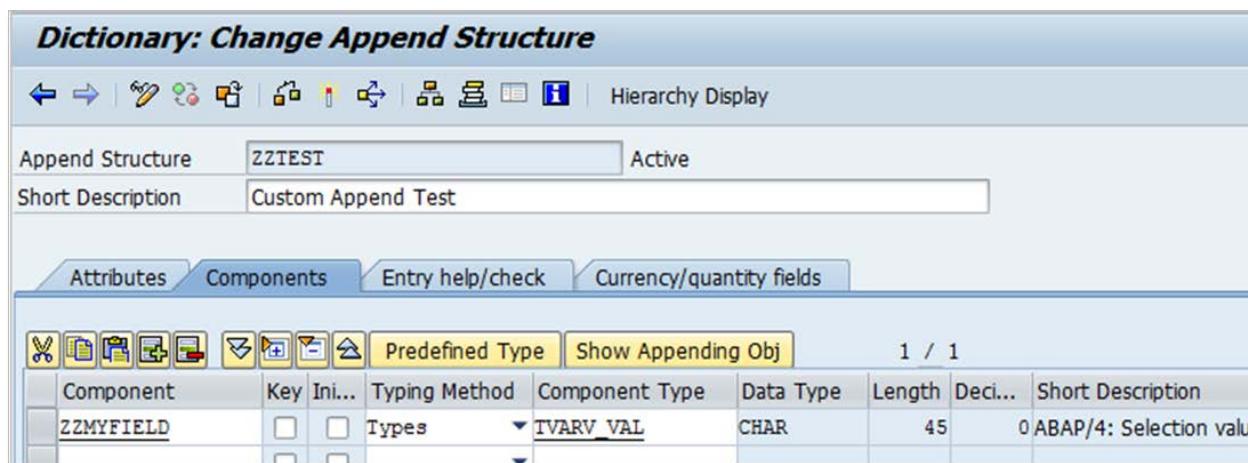


Adding Custom Fields to /IDT/D_TAX_DATA Table

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 now can be used in the Global Next Flexible Field Mapper to map data to.

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 taken into account.

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:

| | |
|--------------|--------------------------|
| o EQ, = | Equal To |
| o NE, <>, >< | Not Equal To |
| o LT, < | Less Than |
| o LE, <= | Less Than or Equal To |
| o GT, > | Greater Than |
| o GE, >= | Greater than or Equal To |
| o CO | Contains Only |
| o CN | Contains Not only |
| o CA | Contains Any |
| o NA | Contains Not Any |
| o CS | Contains String |
| o NS | Contains No String |
| o CP | Matches Pattern |
| o NP | Does Not Match Pattern |
| o IS_INITIAL | Field is initial value |

IS_INITIAL is a special command that can be used with EQ or NE in order 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 Order | Ac. | Source Base | Source Field | Target Base | Target Field | O.. | Description | Simple Expression |
|-----------------------------|------------|-------------------------------------|-------------|--------------|-------------|--------------|--------------------------|-----------------|----------------------------|
| /IDT/JOURNEY_HEADER_REQUEST | 100011 | <input checked="" type="checkbox"/> | SAP_HEAD... | KOMK-PRSDT | DET_INVOICE | INVOICE_DATE | <input type="checkbox"/> | MM INVOICE_DATE | |
| /IDT/JOURNEY_HEADER_REQUEST | 100012 | <input checked="" type="checkbox"/> | SAP_HEAD... | KOMK-AUDAT | DET_INVOICE | INVOICE_DATE | <input type="checkbox"/> | SD INVOICE_DATE | &KOMK-AUDAT& NE IS_INITIAL |
| /IDT/JOURNEY_HEADER_REQUEST | 100013 | <input checked="" type="checkbox"/> | SAP_HEAD... | BKPF-BLDAT | DET_INVOICE | INVOICE_DATE | <input type="checkbox"/> | 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.

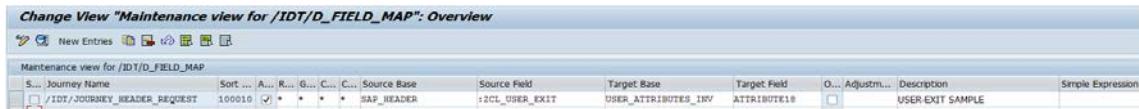
Other example for operands listed above:

| 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 end result would look something like this sample:



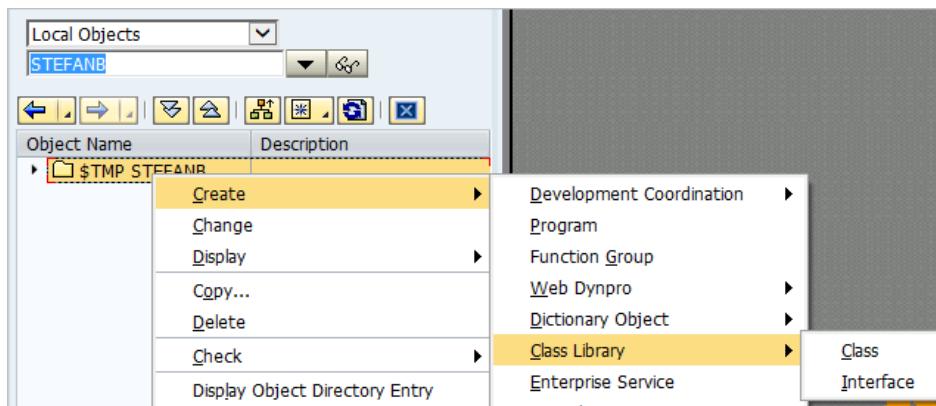
As you can see in above sample the Source Field has been replaced with an ABAP Class. This class has to be created first, and then assigned in the field mapping to the appropriate line. The class name has to be pre-pended with a **colon** (:) .

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.

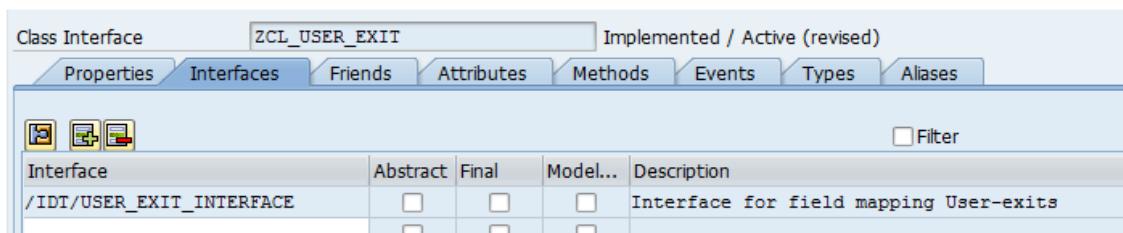
Creating a user-exit based mapping requires a few simple steps of coding. The actual code logic within the user-exit might be complex however depending on the use case.

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

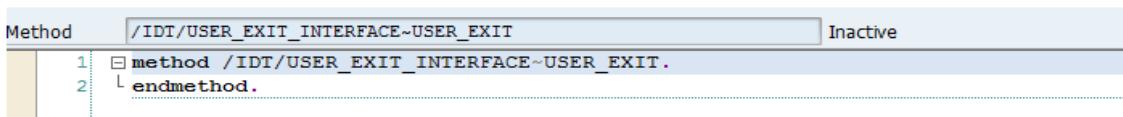
1. Create your class.



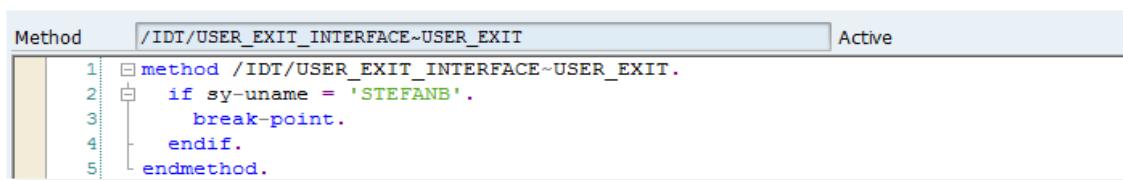
2. **Save** your class. You will be taken to the new class. Navigate to the **Interfaces** tab. Here you will need to include the predefined interface **/IDT/USER_EXIT_INTERFACE** for the Field Mapping which inherits all the necessary data for the mapper.



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).



Field Mapper Journey Drop Down List

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.

| Change View "Maintenance view for /IDT/D_FIELD_MAP": Overview | | | | | | | | | | | | | | | |
|---|-----------------------------|----------|------|------|------|------|------|-------------|----------------|---------------------|--------------|------|------------|------------------|-------------------|
| Maintenance view for /IDT/D_FIELD_MAP | | | | | | | | | | | | | | | |
| S... | Journey Name | Sort ... | A... | R... | G... | C... | C... | Source Base | Source Field | Target Base | Target Field | O... | Adjustm... | Description | Simple Expression |
| | /IDT/JOURNEY_HEADER_REQUEST | 100010 | ✓ | * | * | * | * | SAP_HEADER | :2CL_USER_EXIT | USER_ATTRIBUTES_INV | ATTRIBUTE18 | | | USER-EXIT SAMPLE | |

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 in order 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.

In order to accomplish either of these scenarios via ABAP programming, you can use this new 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
  MV_VARIABLE = M_REF_UTIL_ITEM->PATH( 'EKPO-?????' )-
  >GET_VALUE_AS_STRING( ).

ENDDO.

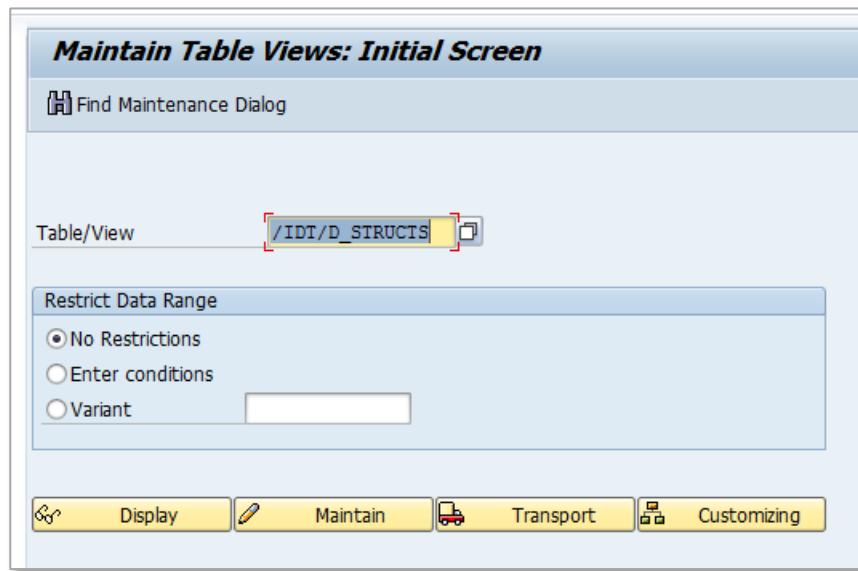
```

FIELD MAPPER JOURNEY DROP DOWN LIST

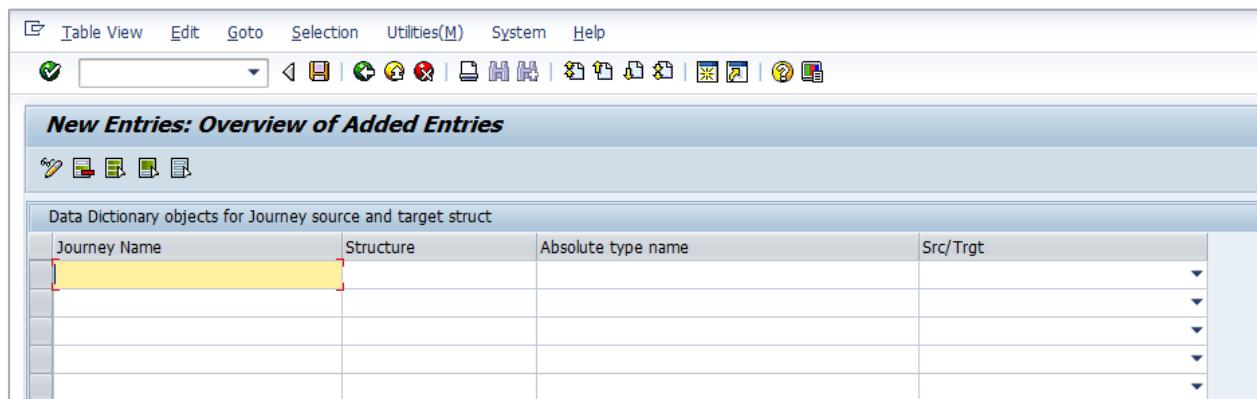
The drop down list of available journeys is a feature of the Field Mapping Table. This drop down list comes from an internal table within the Integration called /IDT/D_STRUCTS. If a user creates a new custom journey that will be used for field mapping purposes, the user will want to make sure that the 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

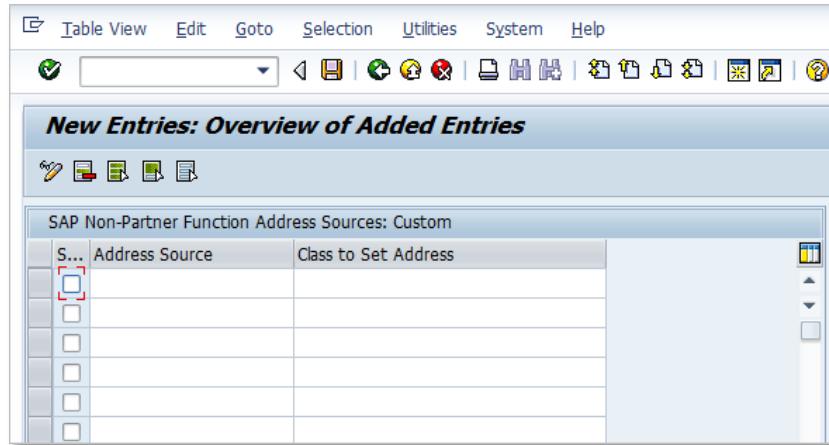
Thomson Reuters provides an extensive list of address sources which can be used in the Address Mapping for tax calculations. In some cases the provided sources might not meet the unique business requirements you might have. You will be able to create your own address source, add code to a user-exit to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



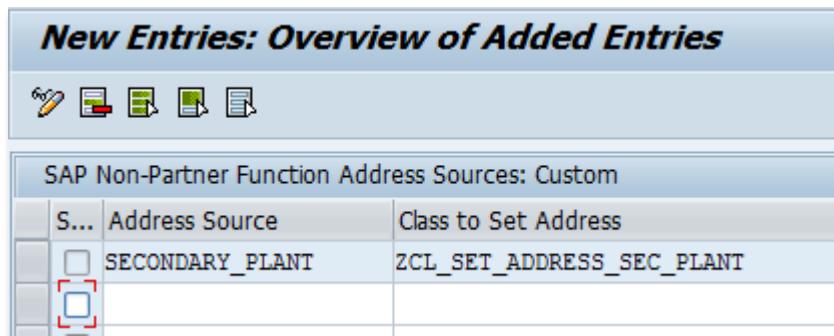
All code samples provided below are for reference purposes only to give an experienced ABAP developer some directions on how to code your own solution. They are not intended to work when copied as is.

Custom Address Source

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



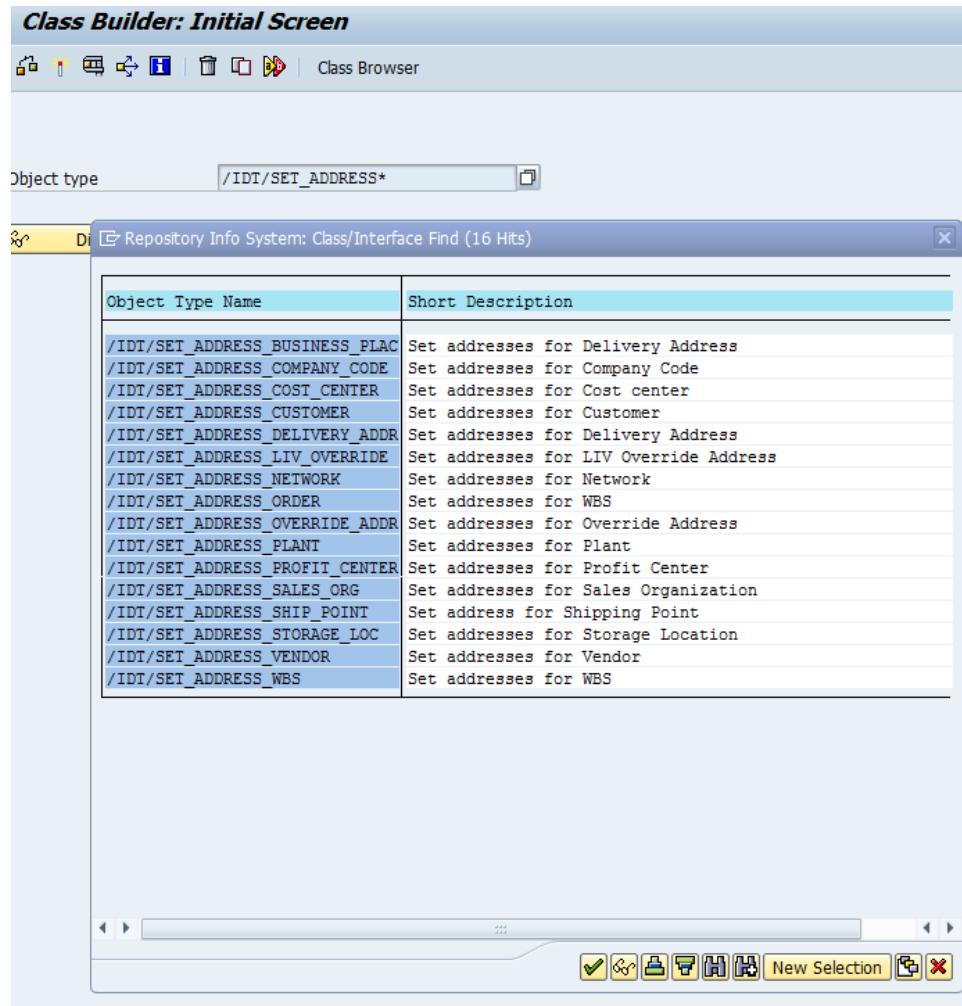
Create the class which will have methods to have logic to populate secondary plant address. Step 3 explains more about class creation procedure and example.



2. Configure transaction **/N/IDT/ADDRESS_MAPPING** to map the new address source to an address type.

| 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 | <input type="checkbox"/> | * | * | WBS ELEMENT | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ShipTo-WBS Element |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 100006 | <input type="checkbox"/> | * | * | NETWORK ELEMENT | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ShipTo-Network Elem |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 100007 | <input type="checkbox"/> | * | * | INTERNAL ORDER | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ShipTo-InternalOrder |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 100011 | <input checked="" type="checkbox"/> | * | * | LIV OVERRIDE ADDRESS | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Override address-Shi |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 100031 | <input type="checkbox"/> | * | * | | IP | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 111111 | <input checked="" type="checkbox"/> | * | * | SECONDARY_PLANT | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Test for INTSAPECC-1062 |
| | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 112111 | <input type="checkbox"/> | * | * | STORAGE LOCATION | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

3. Class **ZCL_SET_ADDRESS_SEC_PLANT** can be created by copying any existing **/IDT/SET_ADDRESS*** class.



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. Now code your logic to populate address in respective method. For example if you need address to be populated at the line item level of MIRO/LIV, then code the logic in the 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**.

Custom Address Source

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.

Class Builder: Class ZCL_SET_ADDRESS_SEC_PLANT Change

Method /IDT/IF_BUILD_ADDRESS-SET_ADDRESS_ITM_LIV Active

```

1  METHOD /IDT/IF_BUILD_ADDRESS-SET_ADDRESS_ITM_LIV.
2  * Copyright 2015 Thomson Reuters/ONESOURCE. All Rights Reserved.
3
4  DATA: ms_sap_address  TYPE /idt/journey_addresses->ty_sap_address.
5
6  ms_sap_address-address          = /idt/address->build_address_from_key('0000000112').
7  ms_sap_address-source_key-address_source = 'SECONDARY_PLANT'.
8  rs_sap_address = ms_sap_address.
9
10 ENDMETHOD.

```

5. Activate the code and you are done.

ADRC: Display of Entries Found

Table to be searched ADRC Addresses (Business Address Services)

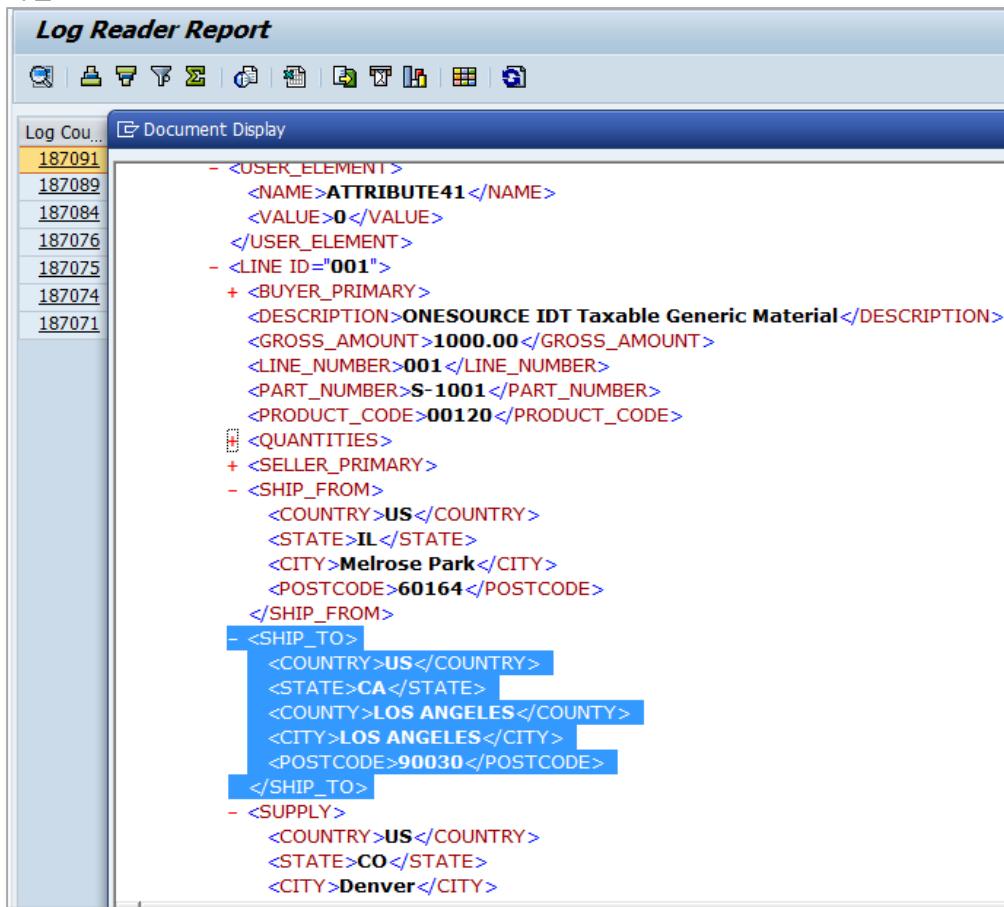
Number of hits 1

Runtime 0 Maximum no. of hits 500

Details

| Addr. n... | From | To | Name | Name 2 | City | District | Postl Co... | Street | House No. | Ctr | Language | Rg | Key | Flag |
|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|----------|-----------|-----|----------|----|------|------|
| 112 | 01/01/0001 | 12/31/9999 | Los Angeles | Los Angeles | LOS ANGELES | LOS ANGELES | 90030 | Broadway | 5950 | US | EN | CA | CA01 | |

Create a MIRO document and check logs to confirm if it is populating the correct address for "ship_to" at line item level.



```

<USER_ELEMENT>
  <NAME>ATTRIBUTE41</NAME>
  <VALUE>0</VALUE>
</USER_ELEMENT>
- <LINE ID="001">
  + <BUYER_PRIMARY>
    <DESCRIPTION>ONESOURCE IDT Taxable Generic Material</DESCRIPTION>
    <GROSS_AMOUNT>1000.00</GROSS_AMOUNT>
    <LINE_NUMBER>001</LINE_NUMBER>
    <PART_NUMBER>S-1001</PART_NUMBER>
    <PRODUCT_CODE>00120</PRODUCT_CODE>
    <QUANTITIES>
    + <SELLER_PRIMARY>
    - <SHIP_FROM>
      <COUNTRY>US</COUNTRY>
      <STATE>IL</STATE>
      <CITY>Melrose Park</CITY>
      <POSTCODE>60164</POSTCODE>
    </SHIP_FROM>
    - <SHIP_TO>
      <COUNTRY>US</COUNTRY>
      <STATE>CA</STATE>
      <COUNTY>LOS ANGELES</COUNTY>
      <CITY>LOS ANGELES</CITY>
      <POSTCODE>90030</POSTCODE>
    </SHIP_TO>
    - <SUPPLY>
      <COUNTRY>US</COUNTRY>
      <STATE>CO</STATE>
      <CITY>Denver</CITY>
    </SUPPLY>
  </SELLER_PRIMARY>
</QUANTITIES>
</BUYER_PRIMARY>
</LINE>

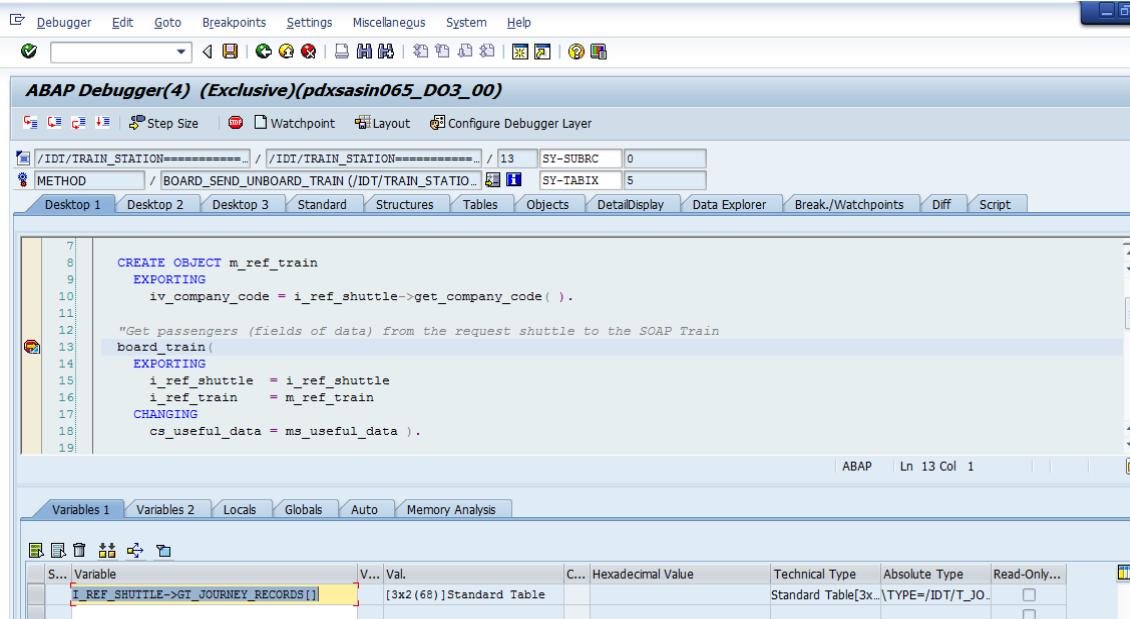
```

DEBUGGING

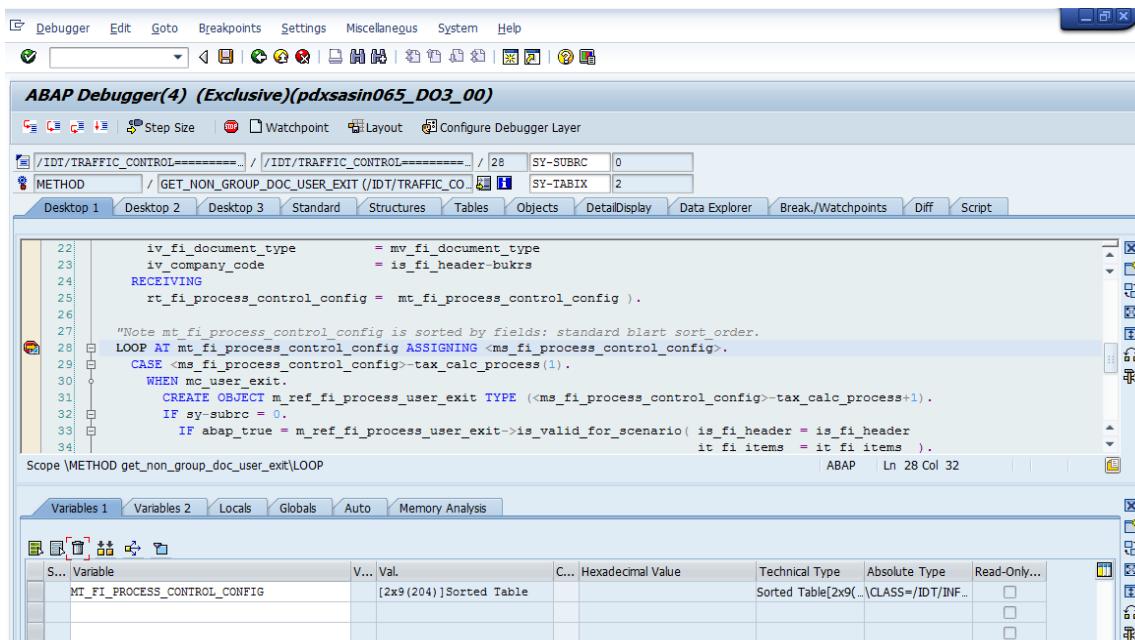
To know what Route/Journeys have been used, put a break-point at /IDT/TRAIN_STATION=>BOARD_SEND_UNBOARD_TRAIN(). From there you can see the actual list of Journeys for boarding and/or unboarding in methods BOARD_TRAIN() and UNBOARD_TRAIN().

Debugging

The following image is an example of the break point.



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 BAdl is used to add security to the proxy for HTTPS communication with Determination. The proxy will require a special user name 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 BAdl 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 BAdl to address these needs.

ADDING IDT ADDRESS FIELD TO INVOICE LINE ITEM ENTRY SCREENS

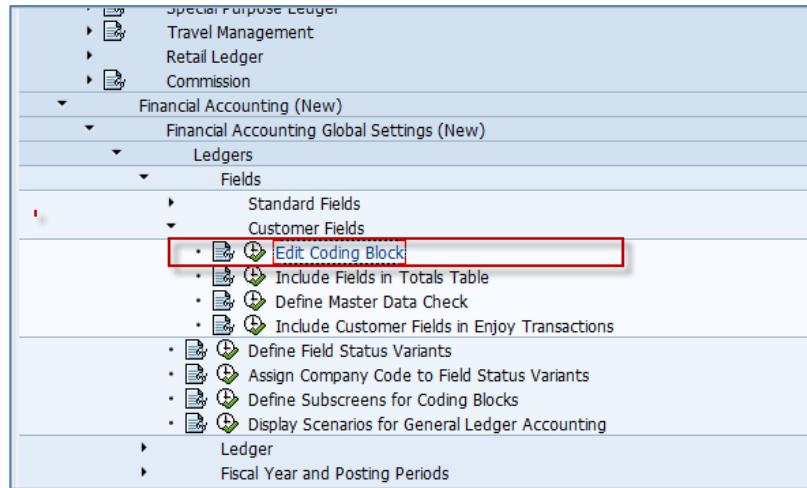
Within the User Guide we discussed a new feature “Changing Ship-To Address at Invoice” where a user could change the default ship to address at time of invoice on transactions FB60, FB65, FB70, FB75, FB01, and MIRO General Ledger tab. The following instructions show you how you can add the new **IDT 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 new 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 new 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. Section below talks about this process and OSS note to use to add a new tab to the line item entry section of the MIRO transaction.

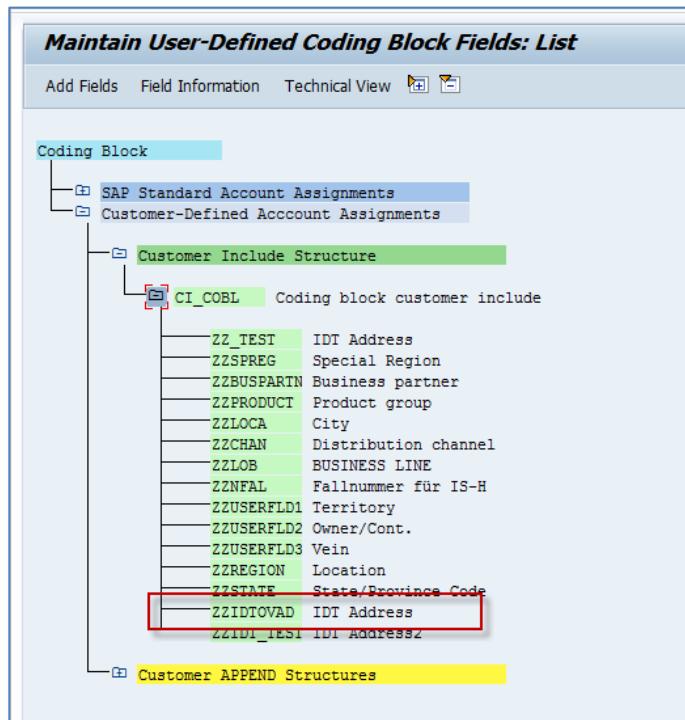
Note that this is not a simple process and will need to be done by an experience ABAP programmer that is familiar with adding custom fields to the code block and screen variants. There are a total of 5 customer fields that are provided by SAP within the coding block. If your company has already used 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**

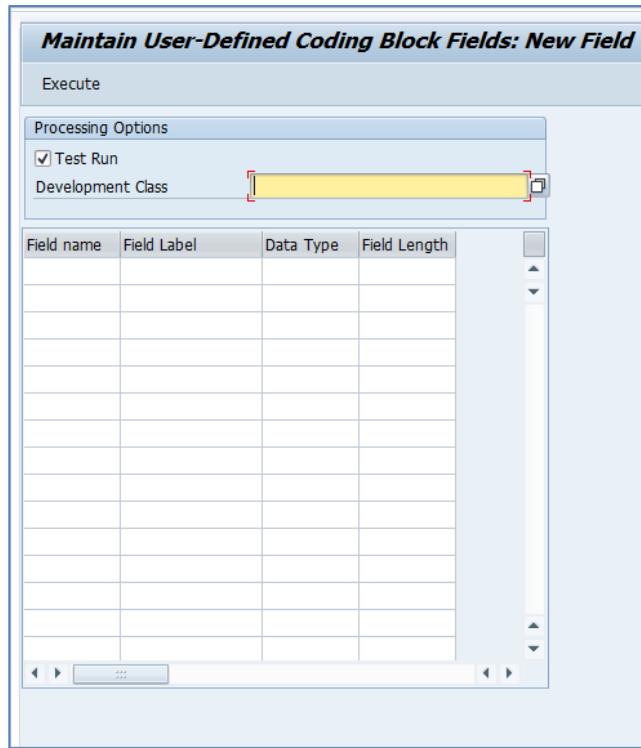


Expand menu to get to Customer Include Structure and then follow the expert mode instructions that are provided for this step. Step 2 and 3 of these instructions are already provided by our system transport and can be skipped.

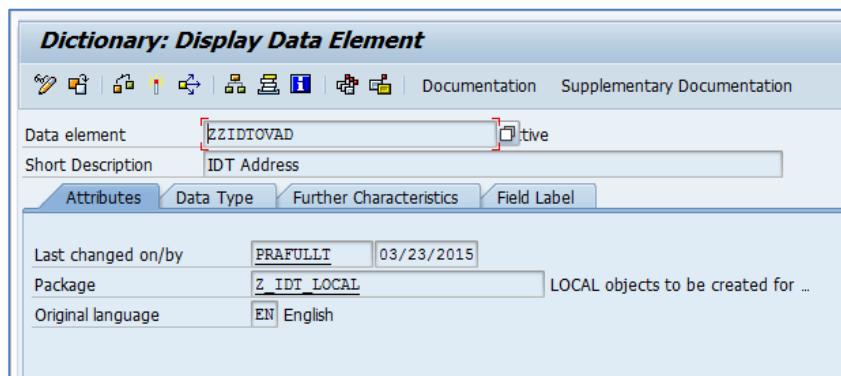


Adding IDT Address Field to Invoice Line Item Entry Screens

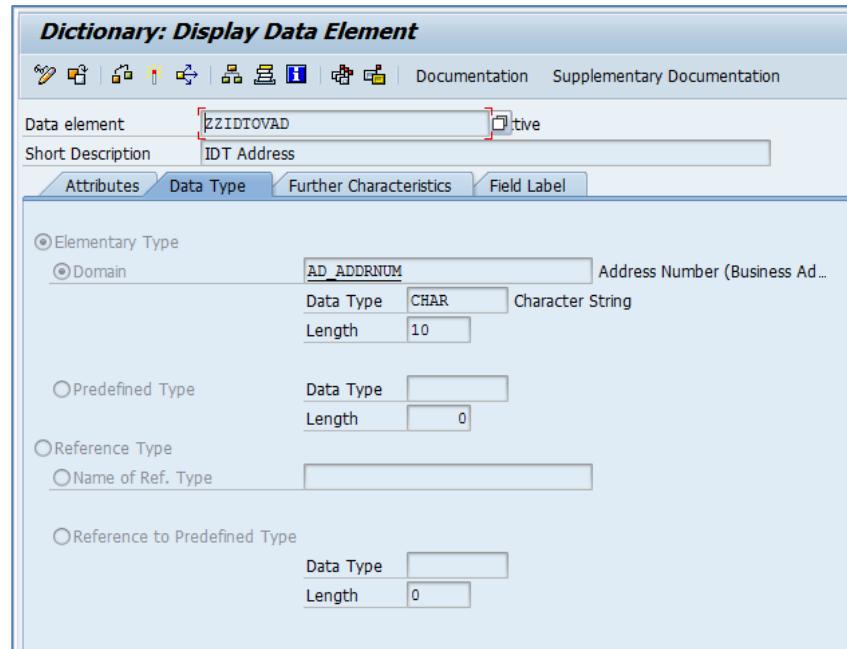
This image shows the User-Defined Coding Block Fields: New Field dialogue.



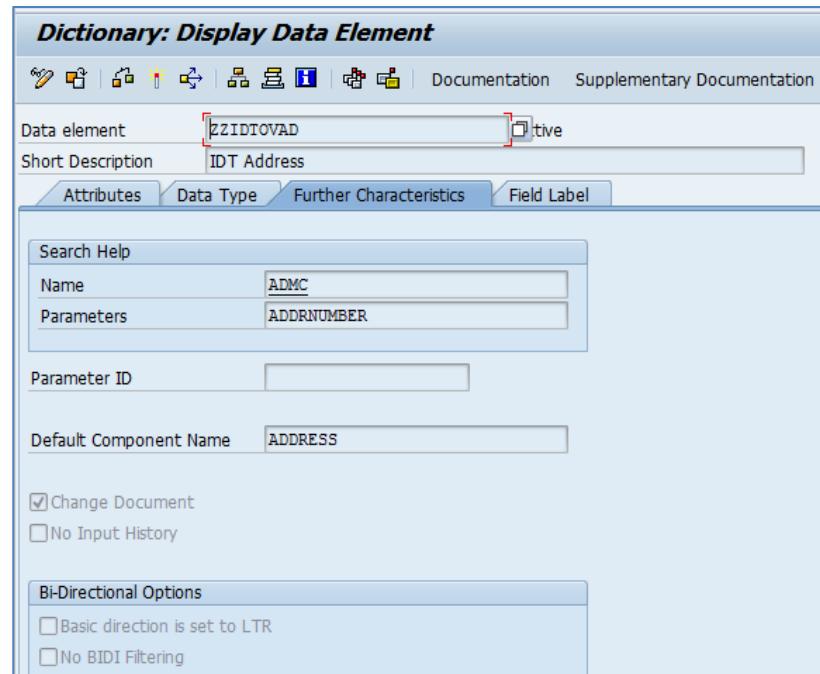
- Development class: **(per your system)**
- Field Name: **ZZIDTOVAD**
- Field Label: **IDT Address**
- Data Type: **CHAR**
- Field Length: **10**

Dictionary: Display Data Element dialogue - Attributes tab

Dictionary: Display Data Element dialogue - Data Type tab

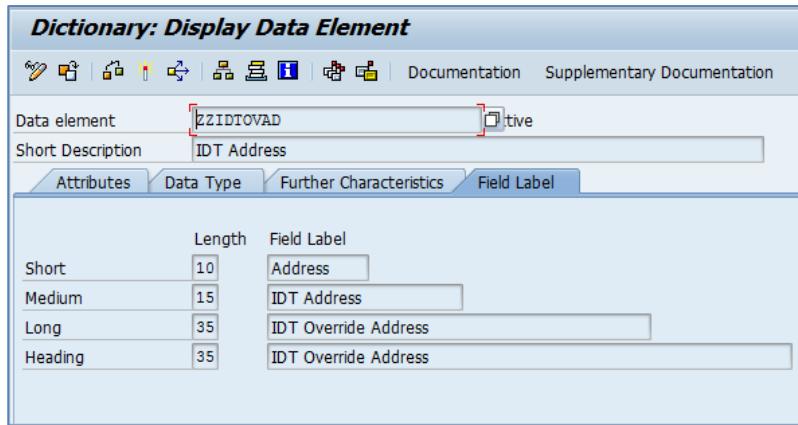


Dictionary: Display Data Element dialogue - Further Characteristics tab



Adding IDT Address Field to Invoice Line Item Entry Screens

Dictionary: Display Data Element dialogue - Field Label tab



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.

Transaction: **SPRO** navigate to > **FINANCIAL ACCOUNTING (NEW) > LEDGERS > FIELDS > CUSTOMER FIELDS >INCLUDE CUSTOMER FIELDS IN ENJOY TRANSACTIONS**

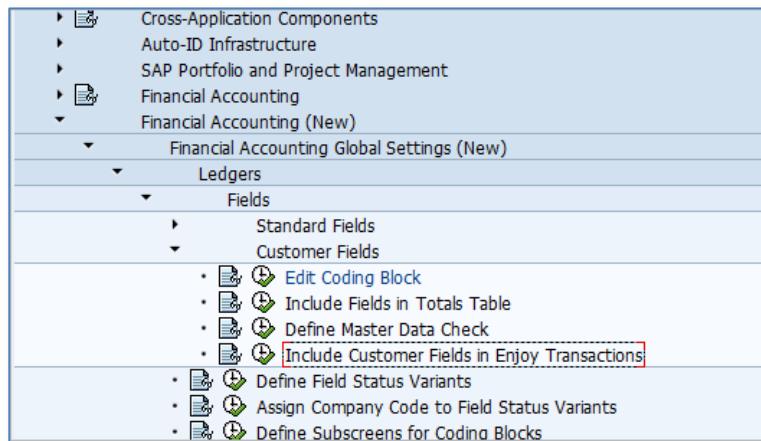
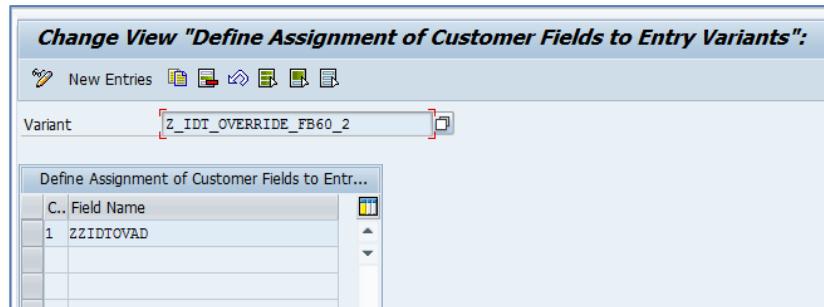
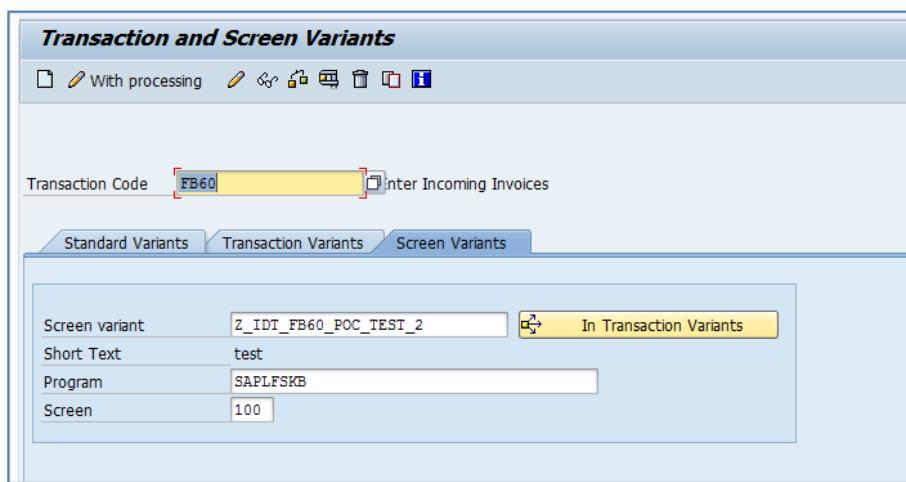


Image of the Change View "Define Assignment of Customer Fields to Entry Variants":



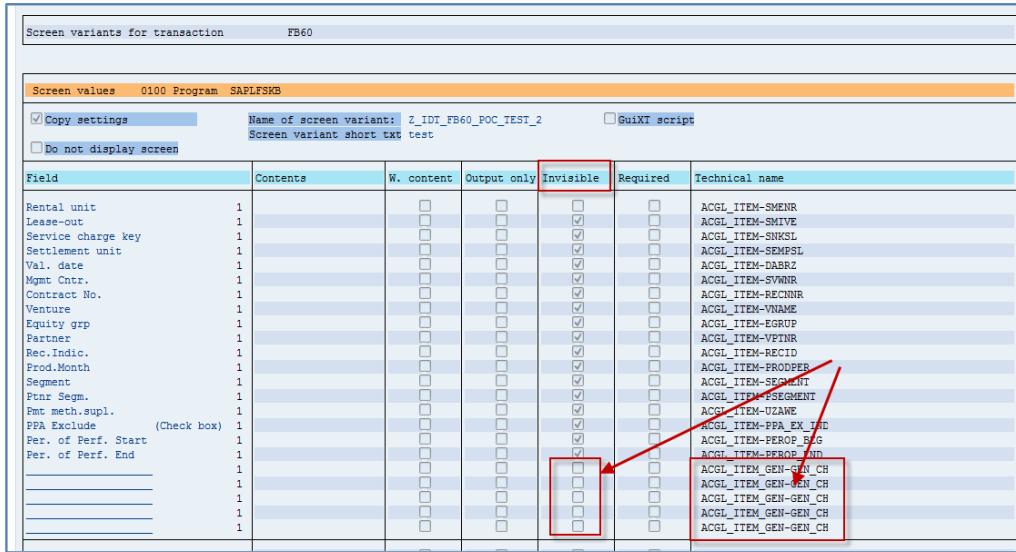
Your variant name will be different.

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



Adding IDT Address Field to Invoice Line Item Entry Screens

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:



The screenshot shows the 'Screen variants for transaction FB60' table. The 'Invisible' column is highlighted with a red border. Two rows at the bottom, corresponding to the field **ACGL_ITEM_GEN_GEN_CH**, have their 'Invisible' checkboxes checked. Red arrows point from the text above to these two rows.

| Field | Contents | W. content | Output only | Invisible | Required | Technical name |
|---------------------|-------------|--------------------------|--------------------------|-------------------------------------|--------------------------|----------------------|
| Rental unit | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SMENR |
| Lease-out | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SMIVE |
| Service charge key | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SNKS1 |
| Settlement unit | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SEMPSL |
| Val. date | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-DABRZ |
| Mgmt. Cntr. | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SVNR |
| Contract No. | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-RECNR |
| Venture | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-UNAME |
| Equity grp | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-EGRUP |
| Partner | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-VPINR |
| Rec.Indic. | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-RECID |
| Prod. Month | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-PRODPER |
| Segment | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-SEGMENT |
| Ptrn Segm. | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-FSEGMENT |
| Pmt meth.supl. | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-UZAWN |
| PFA Exclude | (Check box) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-PFA_EX_IND |
| Per. of Perf. Start | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-FEROP_IND |
| Per. of Perf. End | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM-FERPRO_IND |
| | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM_GEN_GEN_CH |
| | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM_GEN_GEN_CH |
| | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM_GEN_GEN_CH |
| | 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | ACGL_ITEM_GEN_GEN_CH |

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

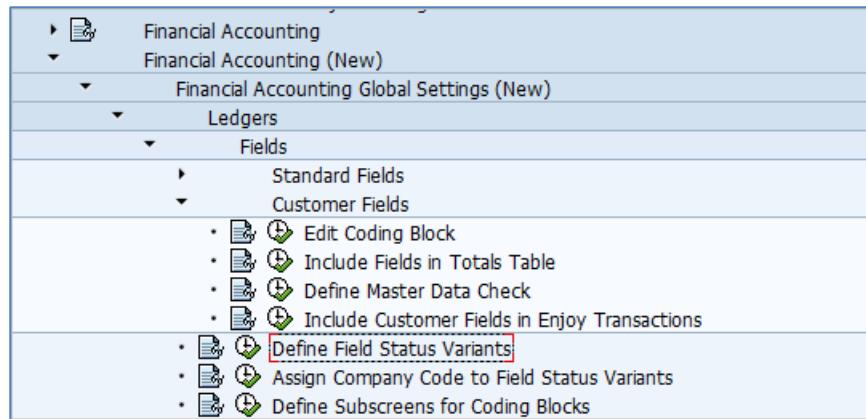
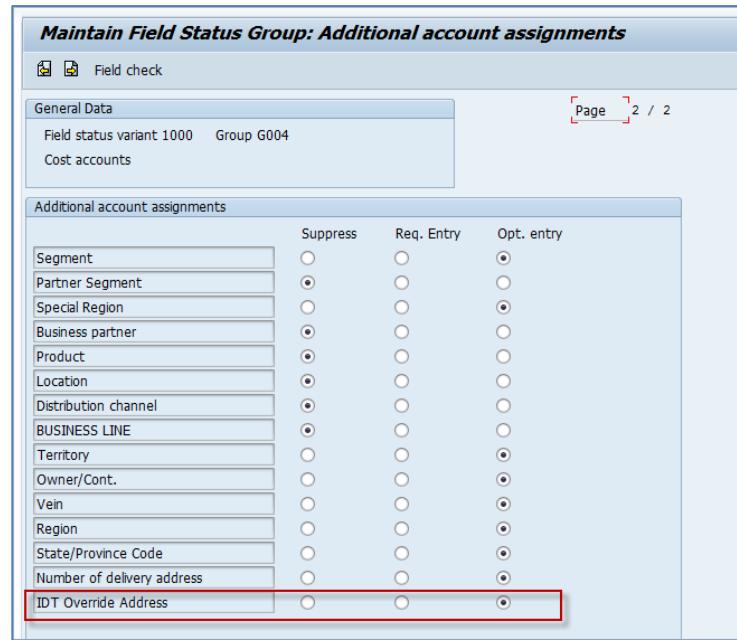


Image of Maintain Field Status Group: Additional account assignments dialogue.



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



Adding IDT Address Field to Invoice Line Item Entry Screens

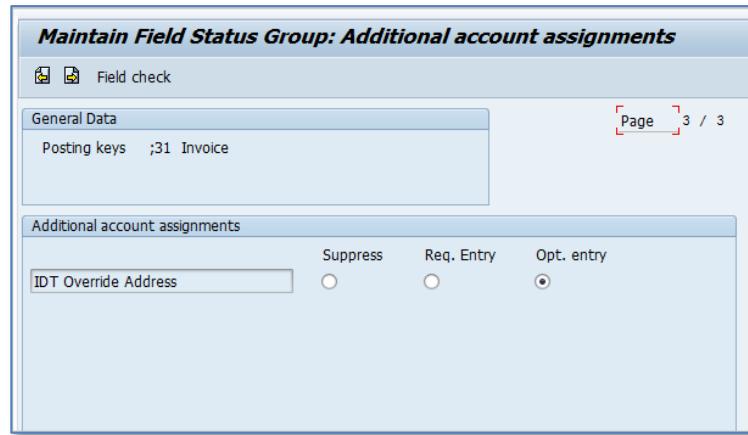
Maintain Accounting Configuration: Posting Keys - List screen shot

| Maintain Accounting Configuration : Posting Keys - List | | | | |
|---|---------------------|--------------|--------------|--|
| 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 | |

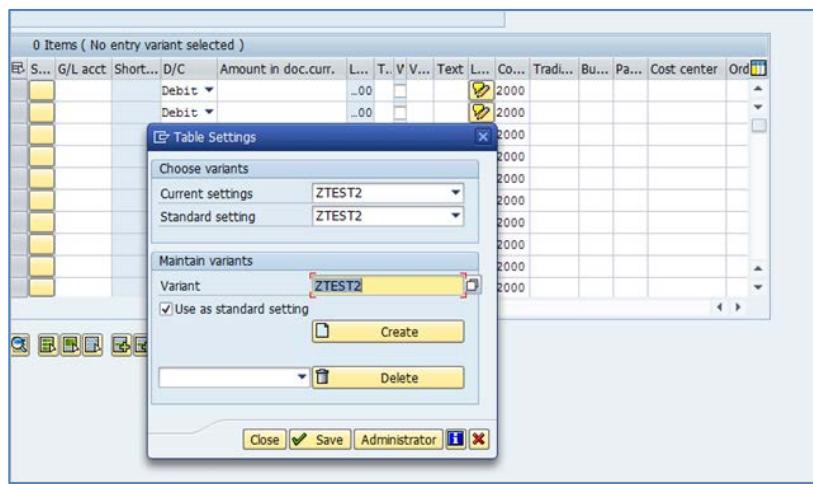
Maintain Accounting Configuration: Posting Keys - Detail screen shot

| Maintain Accounting Configuration : Posting Keys - Detail Screen | |
|--|---|
| Maintain Field Status |  |
| Posting Key | 31 Invoice |
| Debit/credit indicator | <input type="radio"/> Debit <input checked="" type="radio"/> Credit |
| Account type | <input type="radio"/> Customer <input checked="" type="radio"/> Vendor <input type="radio"/> G/L account <input type="radio"/> Assets <input type="radio"/> Material |
| Other attributes | <input checked="" type="checkbox"/> Sales-related <input type="checkbox"/> Special G/L <input type="checkbox"/> Reversal posting key 22 <input type="checkbox"/> Payment transaction |

Maintain Field Status Group: Additional account assignments screen shot



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.

| Display View "Address Mappings: Custom": Overview | | | | | | | | | | | | |
|---|------------------------------|----------------------|----------|-------------------------------------|------|--------------|------------------|-------|--------------------------|-------------------------------------|--------------------------|-------------|
| S... | Route Name | Logical Address Type | Sort ... | A... | C... | CoCd | Address Source | Funct | B... | S... | M... | Description |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_AP | SHIP_TO | 100010 | <input checked="" type="checkbox"/> | * | * | OVERRIDE ADDRESS | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_AR | SELLER_PRIMARY | 111111 | <input type="checkbox"/> | * | * | | GS | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_AR | SHIP_FROM | 111111 | <input type="checkbox"/> | * | * | | GS | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_AR | SHIP_TO | 100010 | <input checked="" type="checkbox"/> | * | * | OVERRIDE ADDRESS | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_FI | SHIP_TO | 100010 | <input checked="" type="checkbox"/> | * | * | OVERRIDE ADDRESS | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_LIV | ORDER_ACCEPTANCE | 100010 | <input type="checkbox"/> | * | 3000 | VENDOR | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_LIV | ORDER_ORIGIN | 100010 | <input type="checkbox"/> | 3000 | COMPANY CODE | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Adding the ZZIDTOVAD Field to the MIRO PO Reference Tab



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 in order to change the ship to address at time of invoice, there is an entirely different process than what is outlined above due to the fact that 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 in order to add a new custom tab to the MIRO screen using **BAdl MRM_ITEM_CUSTFIELDS**. The instructions also refer to several other LIV BAdl 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.

| Component | Typing Method | Component Type | Data Type | Length | Decim... | Short Description | Group |
|-------------|---------------|-----------------|-----------|--------|----------|--|-------|
| _INCLUDE | Types | ▼ CI_DRSEG_CUST | ... | 0 | | 0 IDT Override Field for MIRO PO Ref Tab | |
| ZIDT_ADDRNR | Types | ▼ ADRNR | CHAR | 10 | | 0 Address | |

General Configuration needs to match append in DRSEG
PO REF OVERRIDE ADDRESS 100001 ZIDT_ADDRNR

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.

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.

The screenshot shows the SAP MIRO transaction for an incoming invoice. The top navigation bar includes 'Show PO structure', 'Show worklist', 'Hold', 'Simulate', 'Messages', and 'Help'. The transaction is set to 'Invoice'. The 'Basic data' tab is selected, showing fields for 'Invoice date' (09/18/2014), 'Reference', 'Posting Date' (09/18/2014), 'Amount' (1,000.00- USD), 'Tax amount', 'Text', 'Paymt terms' (Due immediately), 'Baseline Date' (09/18/2014), and 'Company Code' (3000 IDES US INC New York). A red box highlights the 'Override Ad' tab in the tab bar below. The 'Override Ad' tab is active, showing a table for line items. The table has columns: Line, PO, Item, and Address n... (with a red arrow pointing to the 'Address n...' column). A single row is visible with values: 1, 4500002241, 10, and an empty address field.



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 new address field is optional and can be added by the system user via the instructions listed below.

With Integration release version 6.4.0.0, we are supporting tax calculation 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 elects to configure the unplanned delivery charges based on the various line items on 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:

Adding Address Field to Unplanned Delivery Function in MIRO Details Tab



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

Notice that in this image the new 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.

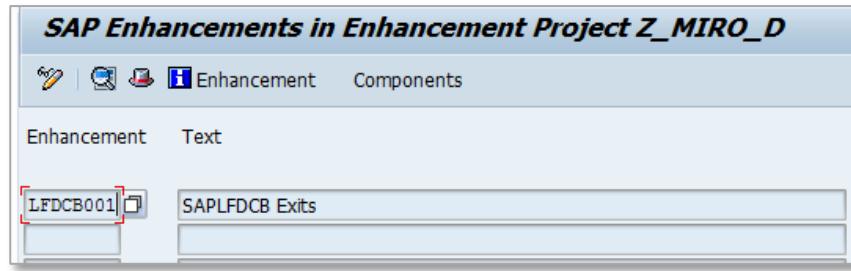
The screenshot shows the SAP interface for 'Enter Incoming Invoice: Company Code 3000'. The 'Details' tab is selected. At the bottom of the screen, there is a scrollable area containing several input fields. One of these fields, 'Address Number', is highlighted with a red box. The field is located in the lower-left quadrant of the scrollable area, below other fields like 'Assignment', 'Header Text', and 'SCB Ind.'

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

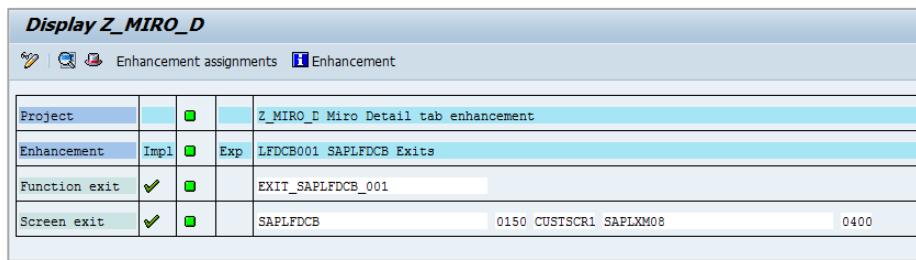
The screenshot shows the SAP Dictionary: Display Structure screen for the structure 'INVFO'. The 'Components' tab is selected. A table lists components and their details. The table has columns: Component, Typing Method, Component Type, Data Type, Length, Deci..., and Short Description. Two components are listed: '.APPEND' (Typing Method: Types, Component Type: ZZMIRO_DETAILS, Data Type: I, Length: 0, Deci...: 0, Short Description: Custom Fields on MIRO Details tab) and 'ZZADRNR' (Typing Method: Types, Component Type: AD_ADDRNUM, Data Type: CHAR, Length: 10, Deci...: 0, Short Description: Address number). The 'ZZADRNR' component is highlighted with a yellow background.

Adding Address Field to Unplanned Delivery Function in MIRO Details Tab

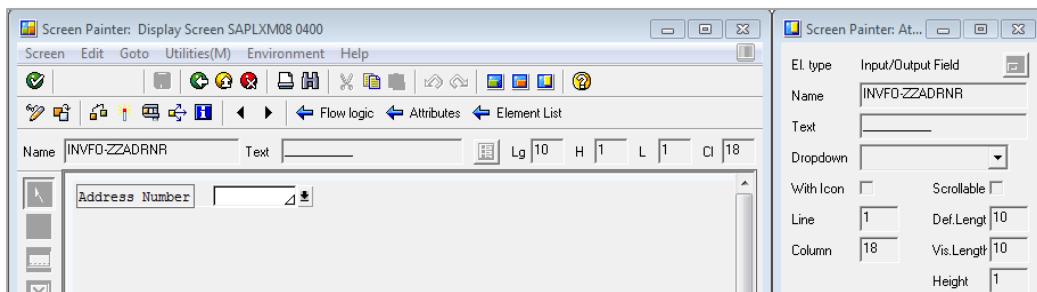
2) User exit LFDCB001 to enhance Details tab screen to add new field for address.



3) Create a Subscreen SAPLXM08-0400.



4) Put address field on the screen.



Adding Address Field to Unplanned Delivery Function in MIRO Details Tab

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.

ABAP Editor: Display Include ZXM08001

```

Include ZXM08001 Active
4  *->-----*
5  *&     Module RECEIVE_DATA  OUTPUT
6  *-----*
7  *     text
8  *-----*
9  MODULE receive_data_0400 OUTPUT.
10
11  data: mv_tax_code type mwskz.
12  FIELD-SYMBOLS : <mv_udc_tax_code> TYPE MWSKZ.
13
14  import mv_tax_code to mv_tax_code FROM MEMORY ID 'UDC_TAX_CODE'.
15  IF mv_tax_code is NOT INITIAL.
16  ASSIGN ('(SAPLFDCB)INVFO-MWSKZ_BNK') TO <mv_udc_tax_code>.
17  IF <mv_udc_tax_code> IS ASSIGNED AND <mv_udc_tax_code> NE space.
18  <mv_udc_tax_code> = mv_tax_code.
19  ENDIF.
20
21  ENDIF.
22
23  ENDMODULE.          " RECEIVE DATA  OUTPUT

```

6) In the PAI method pass address number.

ABAP Editor: Display Include ZXM08001

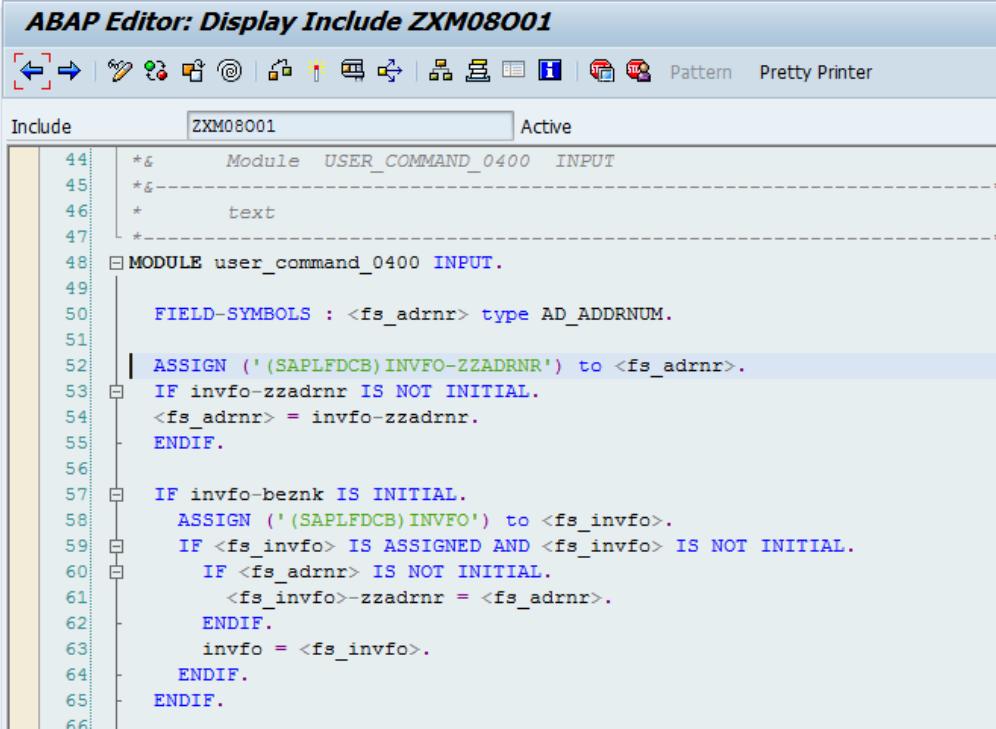
```

Include ZXM08001 Active
22  ENDMODULE.          " RECEIVE DATA  OUTPUT
23
24  *->-----*
25  *&     Module RECEIVE_ACTUAL_DATA  INPUT
26  *-----*
27  *     text
28  *-----*
29  MODULE receive_actual_data_0400 INPUT.
30
31  FIELD-SYMBOLS: <fs_invfo> TYPE invfo.
32
33  IF invfo-beznk IS INITIAL.
34  ASSIGN ('(SAPLFDCB)INVFO') to <fs_invfo>.
35  IF <fs_invfo> IS ASSIGNED AND <fs_invfo> IS NOT INITIAL.
36  IF invfo-zzadrnr IS NOT INITIAL.
37  <fs_invfo>-zzadrnr = invfo-zzadrnr.
38  ENDIF.
39  invfo = <fs_invfo>.
40  ENDIF.
41
42  ENDMODULE.          " RECEIVE ACTUAL DATA  INPUT

```

7) Put below code in user command module.

ABAP Editor: Display Include ZXMO8001



```

44  *-& Module USER_COMMAND_0400 INPUT
45  *-----
46  *     text
47  *-----
48  MODULE user_command_0400 INPUT.
49
50     FIELD-SYMBOLS : <fs_adrnr> type AD_ADDRNUM.
51
52     ASSIGN ('(SAPLFDCB)INVFO-ZZADRNR') to <fs_adrnr>.
53     IF invfo-zzadrnr IS NOT INITIAL.
54         <fs_adrnr> = invfo-zzadrnr.
55     ENDIF.
56
57     IF invfo-beznk IS INITIAL.
58         ASSIGN ('(SAPLFDCB)INVFO') to <fs_invfo>.
59         IF <fs_invfo> IS ASSIGNED AND <fs_invfo> IS NOT INITIAL.
60             IF <fs_adrnr> IS NOT INITIAL.
61                 <fs_invfo>-zzadrnr = <fs_adrnr>.
62             ENDIF.
63             invfo = <fs_invfo>.
64         ENDIF.
65     ENDIF.
66

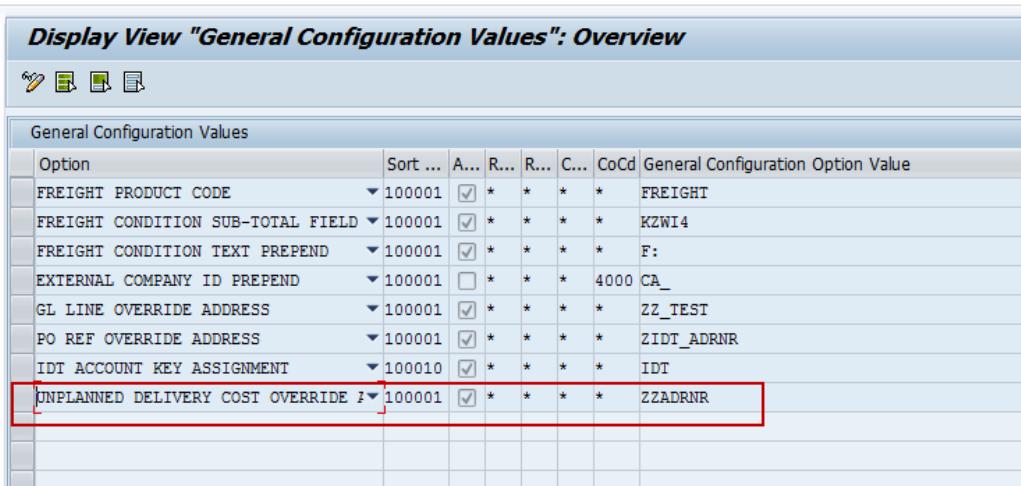
```



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.

Display View "General Configuration Values": Overview



| General Configuration Values | | | | | | | |
|-----------------------------------|----------|-------------------------------------|------|------|------|------|------------------------------------|
| Option | Sort ... | A... | R... | R... | C... | CoCd | General Configuration Option Value |
| FREIGHT PRODUCT CODE | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | FREIGHT |
| FREIGHT CONDITION SUB-TOTAL FIELD | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | KZWI4 |
| FREIGHT CONDITION TEXT PREPEND | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | F: |
| EXTERNAL COMPANY ID PREPEND | ▼ 100001 | <input type="checkbox"/> | * | * | * | * | 4000 CA_ |
| GL LINE OVERRIDE ADDRESS | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | ZZ_TEST |
| PO REF OVERRIDE ADDRESS | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | ZIDT_ADRNR |
| IDT ACCOUNT KEY ASSIGNMENT | ▼ 100010 | <input checked="" type="checkbox"/> | * | * | * | * | IDT |
| UNPLANNED DELIVERY COST OVERRIDE | ▼ 100001 | <input checked="" type="checkbox"/> | * | * | * | * | ZZADRNR |

Adding IDT Address Field to Service Entry Sheet Line Item

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.

| Change View "Address Mappings: Custom": Overview | | | | | | | | | | |
|---|------------------------------|----------------------|----------|-------------------------------------|------|------|------------------------|--------------------------|--------------------------|--------------------------|
|       | | | | | | | | | | |
| Address Mappings: Custom | | | | | | | | | | |
| S... | Route Name | Logical Address Type | Sort ... | A... | C... | CoCd | Address Source | Funct | B... | S... |
| <input type="checkbox"/> | /IDT/ROUTE_NON_GROUP_DOC_LIV | SHIP_TO | 121111 | <input checked="" type="checkbox"/> | * | * | LIV UNPLANNED ADDRE... | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | /IDT/ROUTE_NON_GROUP_DOC_SRS | BUYER_BRTMADV | 100001 | <input type="checkbox"/> | * | * | 1200 BUSINESS PLACE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

ADDING IDT ADDRESS FIELD TO SERVICE ENTRY SHEET LINE ITEM

/IDT/BADI_SET_FREIGHT_LIV_UDC

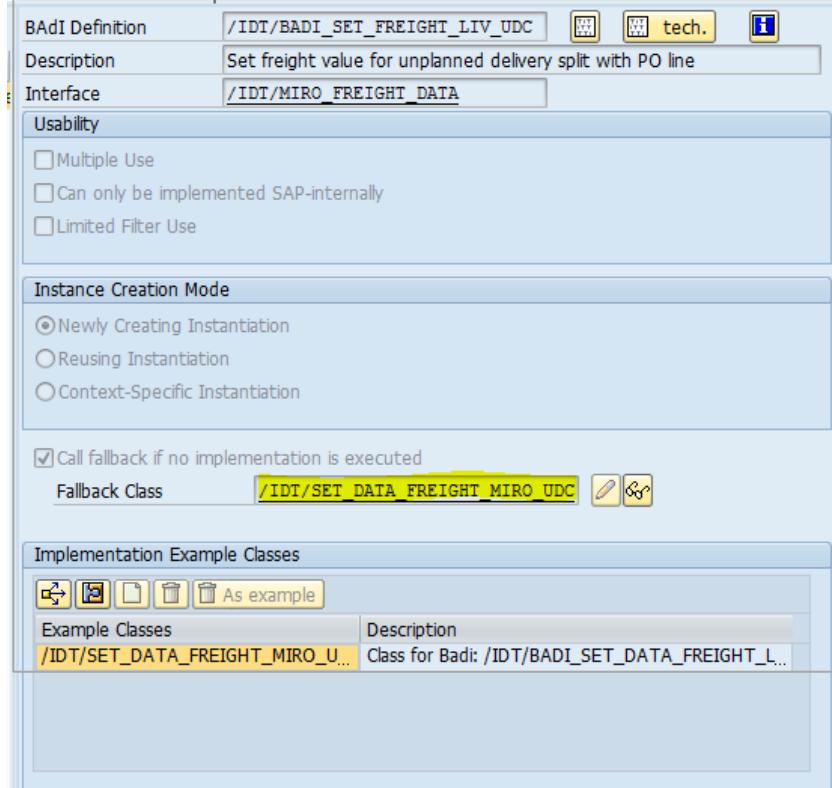
The purpose of the BAdI is to determine freight value only when it is entered at the MIRO details tab as an unplanned delivery charge and system configuration is set to option 0 or 1 which splits it up and includes it as part of the PO line item amount. Every business may have different logic to determine 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.

Adding MIRO Multiple Account Assignment Address Overrides

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_value = 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 ASSIGNMENT ADDRESS OVERIDES

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 Subscreen
- 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.

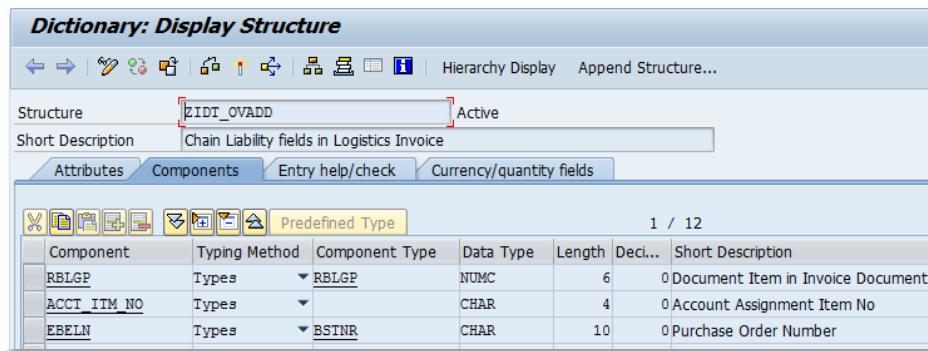
1. DRSEG (DRSEG_CI).

| Dictionary: Display Structure | | | | | | |
|---|---------------|---|-----------|------------------|---------|--|
| Structure | | DRSEG_CI | | Active | | |
| Short Description | | Transfer to BAdI for Customer's Own Invoice Item Data | | | | |
| Attributes | | Components | | Entry help/check | | Currency/quantity fields |
|        | |        | | Predefined Type | | 1 / 13 |
| Component | Typing Method | Component Type | Data Type | Length | Deci... | Short Description |
| <u>C_RBLGP</u> | Types | ▼ RBLGP | NUMC | 6 | 0 | Document Item in Invoice Document |
| <u>C_KOART</u> | Types | ▼ KOART | CHAR | 1 | 0 | Account Type |
| <u>C_MATNR</u> | Types | ▼ MATNR | CHAR | 18 | 0 | Material Number |
| <u>.INCLUDE</u> | Types | ▼ CI_DRSEG_CUST | OO | 0 | 0 | IDT Override Field for MIRO PO Ref Tab |
| <u>ACCT_ITM_NO</u> | Types | | CHAR | 4 | 0 | |
| <u>ZZ_MWSKZ</u> | Types | ▼ MWSKZ | CHAR | 2 | 0 | Tax on sales/purchases code |
| <u>ZZ_KOSTL</u> | Types | ▼ KOSTL | CHAR | 10 | 0 | Cost Center |
| <u>ZZ_AUFNR</u> | Types | ▼ AUFNR | CHAR | 12 | 0 | Order Number |
| <u>ZZ_PSP_PNR</u> | Types | ▼ PSP_PNR | CHAR | 24 | 0 | Work breakdown structure element (WBS element) |
| <u>ZZ_NPLNR</u> | Types | ▼ NPLNR | CHAR | 12 | 0 | Network Number for Account Assignment |
| <u>ZZ_ANLN1</u> | Types | ▼ ANLN1 | CHAR | 12 | 0 | Main Asset Number |
| <u>ZZ_ANLN2</u> | Types | ▼ ANLN2 | CHAR | 4 | 0 | Asset Subnumber |
| <u>ZIDTADRNR</u> | Types | ▼ ADRNR | CHAR | 10 | 0 | Address |

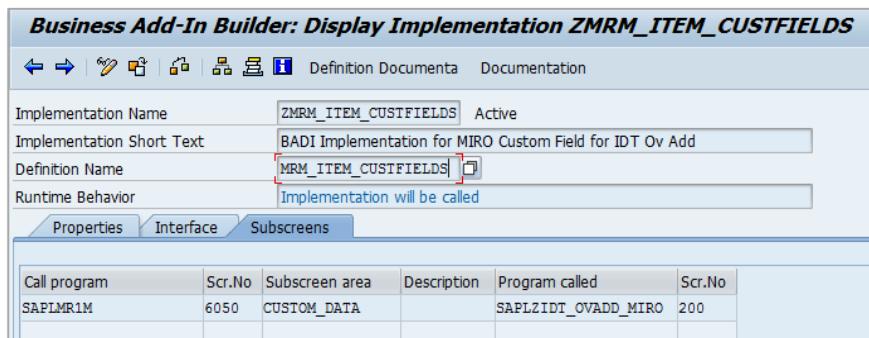
2. ZWA_TC_WKA – This structure has been used in screen enhancement code.

| Dictionary: Display Structure | | | | | | |
|-------------------------------|---------------|---|-----------|--------|------------------|-----------------------|
| Structure | | [ZWA_TC_WKA] | | Active | | |
| Short Description | | Chain Liability fields in Logistics Invoice | | | | |
| Attributes | | Components | | | Entry help/check | |
| Currency/quantity fields | | | | | | |
| | | | | | | Predefined Type |
| 1 / 12 | | | | | | |
| Component | Typing Method | Component Type | Data Type | Length | Deci... | Short Description |
| RBLGP | Types | <u>RBLGP</u> | NUMC | 6 | 0 | Document Item in Inv |
| ACCT_ITM_NO | Types | <u>ACCT_ITM_NO</u> | CHAR | 4 | 0 | Account Assignment |
| EBELN | Types | <u>EBELN</u> | CHAR | 10 | 0 | Purchase Order Num |
| EBELP | Types | <u>EBELP</u> | NUMC | 5 | 0 | Item Number of Purch |
| ZZ_MWSKZ | Types | <u>ZZ_MWSKZ</u> | CHAR | 2 | 0 | Tax on sales/purchase |
| ZZ_KOSTL | Types | <u>ZZ_KOSTL</u> | CHAR | 10 | 0 | Cost Center |
| ZZ_AUFNR | Types | <u>ZZ_AUFNR</u> | CHAR | 12 | 0 | Order Number |
| ZZ_PSP_PNR | Types | <u>ZZ_PSP_PNR</u> | CHAR | 24 | 0 | Work breakdown stru |
| ZZ_NPLNR | Types | <u>ZZ_NPLNR</u> | CHAR | 12 | 0 | Network Number for |
| ZZ_ANLN1 | Types | <u>ZZ_ANLN1</u> | CHAR | 12 | 0 | Main Asset Number |
| ZZ_ANLN2 | Types | <u>ZZ_ANLN2</u> | CHAR | 4 | 0 | Asset Subnumber |
| ZIDT_ADDRNR | Types | <u>ZIDT_ADDRNR</u> | CHAR | 10 | 0 | Address number |

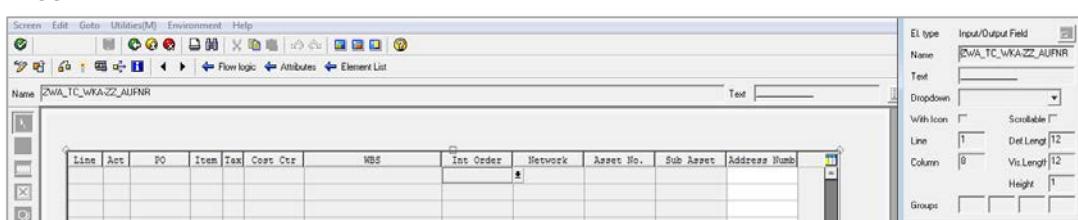
3. Create a very similar structure with the name - ZIDT_OVADD



Implement BAdI MRM_ITEM_CUSTFIELDS



Screen 200



Adding MIRO Multiple Account Assignment Address Overrides

Create the Subscreen

Module pool code for this screen is attached below –

Screen 200 – Code

Flow Logic –

Screen Painter: Display Screen for SAPLZIDT_OVADD_MIRO

Screen number 200 Active

Attributes Element list Flow logic

```

1  PROCESS BEFORE OUTPUT.
2
3  MODULE badi_pbo.
4
5  *-&SPWIZARD: PBO FLOW LOGIC FOR TABLECONTROL 'TC_WKA_TAB'
6  MODULE tc_wka_tab_change_tc_attr.
7  *-&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_COL_ATTR.
8  [LOOP AT tab_tc_wka
9    INTO zwa_tc_wka
10   WITH CONTROL tc_wka_tab
11   CURSOR tc_wka_tab-top_line.
12   *   MODULE TC_WKA_TAB_GET_LINES.
13   MODULE tc_wka_tab_set_attributes.
14  *-&SPWIZARD:   MODULE TC_WKA_TAB_CHANGE_FIELD_ATTR
15  ENDLOOP.
16
17  PROCESS AFTER INPUT.
18
19  *-&SPWIZARD: PAI FLOW LOGIC FOR TABLECONTROL 'TC_WKA_TAB'
20  [LOOP AT tab_tc_wka.
21    CHAIN.
22    FIELD zwa_tc_wka-rblgp.
23    FIELD zwa_tc_wka-ebeln.
24    FIELD zwa_tc_wka-ebelp.
25    FIELD zwa_tc_wka-zz_mwskz.
26    FIELD zwa_tc_wka-zz_kostl.
27    FIELD zwa_tc_wka-zz_psp_pnr.
28    FIELD zwa_tc_wka-zz_aufnr.
29    FIELD zwa_tc_wka-zz_nplnr.
30    FIELD zwa_tc_wka-zz_anln1.
31    FIELD zwa_tc_wka-zz_anln2.
32    FIELD zwa_tc_wka-acct_itm_no.
33    FIELD zwa_tc_wka-zidt_adrnr.
34    MODULE tc_wka_tab_modify ON CHAIN-REQUEST.
35    ENDCHAIN.
36    * To keep track of the table control line count
37    MODULE tc_wka_tab_set_line_count.
38  ENDLOOP.
39
40  MODULE tc_wka_fields_check.
41  MODULE tc_wka_tab_user_command.
42  *-&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_TC_ATTR.
43  *-&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_COL_ATTR.
44
45
46  MODULE badi_pai.

```

Screen Painter: Display Screen for SAPLZIDT_OVADD_MIRO

Screen number 200 Active

Attributes Element list Flow logic

```

23  FIELD zwa_tc_wka-ebeln.
24  FIELD zwa_tc_wka-ebelp.
25  FIELD zwa_tc_wka-zz_mwskz.
26  FIELD zwa_tc_wka-zz_kostl.
27  FIELD zwa_tc_wka-zz_psp_pnr.
28  FIELD zwa_tc_wka-zz_aufnr.
29  FIELD zwa_tc_wka-zz_nplnr.
30  FIELD zwa_tc_wka-zz_anln1.
31  FIELD zwa_tc_wka-zz_anln2.
32  FIELD zwa_tc_wka-acct_itm_no.
33  FIELD zwa_tc_wka-zidt_adrnr.
34  MODULE tc_wka_tab_modify ON CHAIN-REQUEST.
35  ENDCHAIN.
36  * To keep track of the table control line count
37  MODULE tc_wka_tab_set_line_count.
38  ENDLOOP.
39
40  MODULE tc_wka_fields_check.
41  MODULE tc_wka_tab_user_command.
42  *-&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_TC_ATTR.
43  *-&SPWIZARD: MODULE TC_WKA_TAB_CHANGE_COL_ATTR.
44
45
46  MODULE badi_pai.

```

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-ZIDTADRNR.
      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.

```

Write code in the PAI/PBO modules of the screen

MODULE BAdI_PAII

PBO module

ABAP Editor: Display Include LZIDT_OVADD_MIRO001

```

Include LZIDT_OVADD_MIRO001 Active

1  * 
2  ***INCLUDE LIDNL_WKAO01 .
3  *
4  *-
5  *& Module badi_pbo  OUTPUT
6  *-
7  * text
8  *-
9  MODULE badi_pbo  OUTPUT.
10
11  DATA: wa_drseg TYPE mmcr_drseg,
12       wa_drseg_custom TYPE drseg_ci,
13       wa_ekpo TYPE ekpo,
14       s_co      TYPE mmcr_drseg_co,
15       lv_index  TYPE sy-index,
16       lv_acct_lines TYPE sy-index,
17       lv_badi_lines TYPE sy-index.|
18
19
20  * IF custfield_modify IS INITIAL.
21  CALL METHOD cl_exithandler->get_instance_for_subscreens
22      CHANGING
23          instance = custfield_modify
24      EXCEPTIONS
25          OTHERS    = 6.
26  * ENDIF.
27
28  CALL METHOD custfield_modify->invoice_data_get
29      ...
30
Scope \MODULE badi_pbo

```

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_ABPPSP_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.

```

Adding MIRO Multiple Account Assignment Address Overrides

```

MOVE-CORRESPONDING WA_DRSEG_CUSTOM TO ZWA_TC_WKA.
MOVE WA_DRSEG-ZIDTADRNR TO ZWA_TC_WKA-ZIDTADRNR.
ELSE.
  IF AKT_TYP = 'A'.
    MOVE-CORRESPONDING WA_DRSEG TO ZWA_TC_WKA.
  ELSE.
    CLEAR ZWA_TC_WKA-ZIDTADRNR.
  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-ZIDTADRNR TO ZWA_TC_WKA-ZIDTADRNR.
  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-ZIDTADRNR = S_CO-ZIDTADRNR.
    ELSE.
      CLEAR ZWA_TC_WKA-ZIDTADRNR.
    ENDIF.
  ENDIF.
ENDIF.

```

```

        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_ABPPSP_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_ATTR

```

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-ZIDTADRNR IS NOT INITIAL.
    DATA VADRNR TYPE ADRC.
    SELECT SINGLE * FROM ADRC INTO VADRNR
      WHERE ADDRNUMBER = ZWA_TC_WKA-
ZIDTADRNR.

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

```

```

ENDLOOP.

ENDMODULE.          " TC_WKA_FIELDS_CHECK INPUT

```

Module TC_WKA_TAB_USER_COMMAND

```

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_PA1

```

MODULE BADI_PA1 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_PA1  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.

| Quantity/Percent | Net Value | Cost Ctr | G/L Account | BsAr | Earmarked... | Item | Profit Center |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 50.0 | 500.00 | 3000 | 475000 | 9000 | | 3000 | |
| 50.0 | 500.00 | 3001 | 476000 | 9000 | | 3000 | |
| | 0.00 | | 417000 | | | | |
| | 0.00 | | 417000 | | | | |

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 –

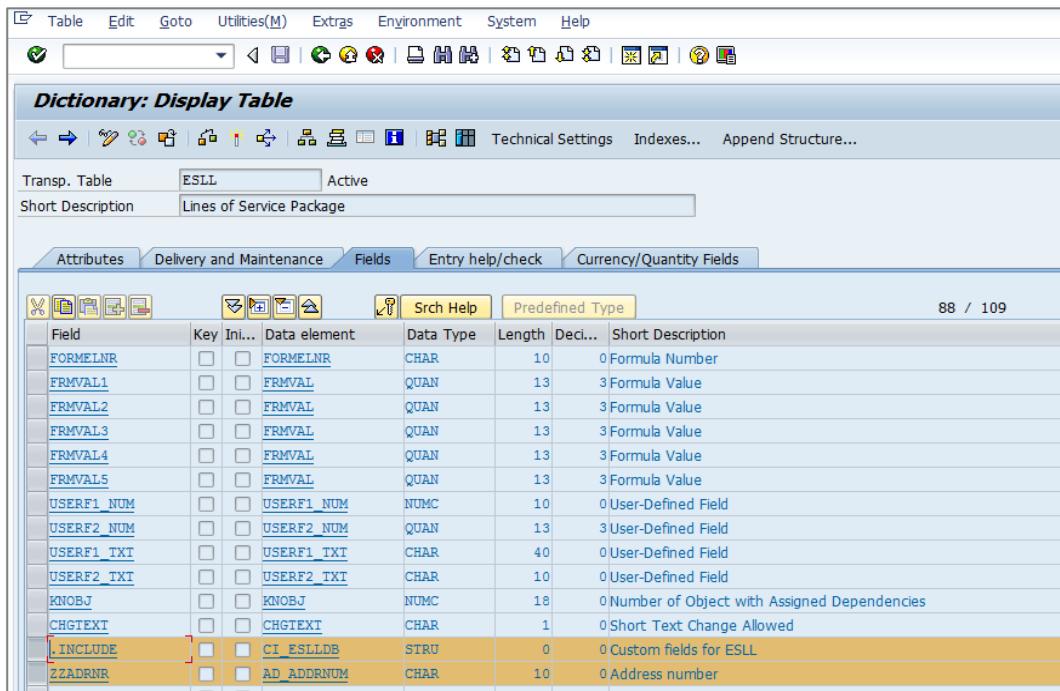
```
INCLUDE /IDT/SERVICES_MAA_ZEKKN_TAX.
```

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.

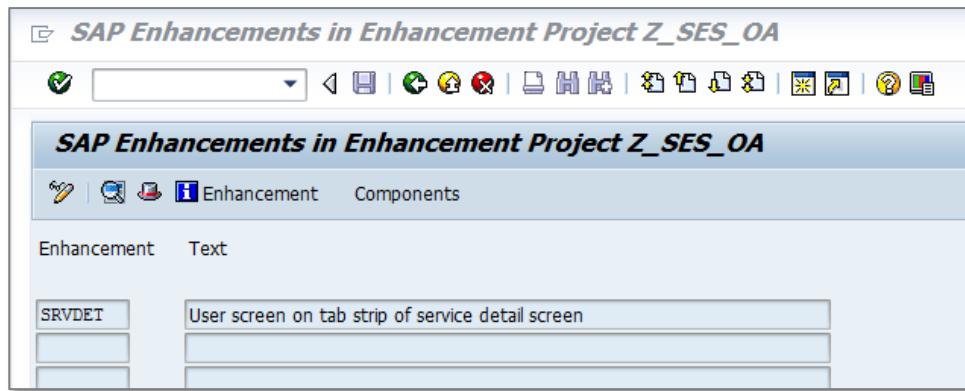
1. Add the new address number field ZZADRNR to the append structure CI_ESLLDB that is part of ESLL table (Example shown below for reference)



| Field | Key | Inh... | Data element | Data Type | Length | Deci... | Short Description |
|------------|--------------------------|--------------------------|--------------|-----------|--------|---------|---|
| FORMELNR | <input type="checkbox"/> | <input type="checkbox"/> | FORMELNR | CHAR | 10 | 0 | Formula Number |
| FRMVAL1 | <input type="checkbox"/> | <input type="checkbox"/> | FRMVAL | QUAN | 13 | 3 | Formula Value |
| FRMVAL2 | <input type="checkbox"/> | <input type="checkbox"/> | FRMVAL | QUAN | 13 | 3 | Formula Value |
| FRMVAL3 | <input type="checkbox"/> | <input type="checkbox"/> | FRMVAL | QUAN | 13 | 3 | Formula Value |
| FRMVAL4 | <input type="checkbox"/> | <input type="checkbox"/> | FRMVAL | QUAN | 13 | 3 | Formula Value |
| FRMVAL5 | <input type="checkbox"/> | <input type="checkbox"/> | FRMVAL | QUAN | 13 | 3 | Formula Value |
| USERF1_NUM | <input type="checkbox"/> | <input type="checkbox"/> | USERF1_NUM | NUMC | 10 | 0 | User-Defined Field |
| USERF2_NUM | <input type="checkbox"/> | <input type="checkbox"/> | USERF2_NUM | QUAN | 13 | 3 | User-Defined Field |
| USERF1_TXT | <input type="checkbox"/> | <input type="checkbox"/> | USERF1_TXT | CHAR | 40 | 0 | User-Defined Field |
| USERF2_TXT | <input type="checkbox"/> | <input type="checkbox"/> | USERF2_TXT | CHAR | 10 | 0 | User-Defined Field |
| KNOBJ | <input type="checkbox"/> | <input type="checkbox"/> | KNOBJ | NUMC | 18 | 0 | Number of Object with Assigned Dependencies |
| CHGTEXT | <input type="checkbox"/> | <input type="checkbox"/> | CHGTEXT | CHAR | 1 | 0 | Short Text Change Allowed |
| INCLUDE | <input type="checkbox"/> | <input type="checkbox"/> | CI_ESLLDB | STRU | 0 | 0 | Custom fields for ESLL |
| ZZADRNR | <input type="checkbox"/> | <input type="checkbox"/> | AD_ADDRNUM | CHAR | 10 | 0 | Address number |

Support of Override Address - Address Number at Service Level in PO and ML81N

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

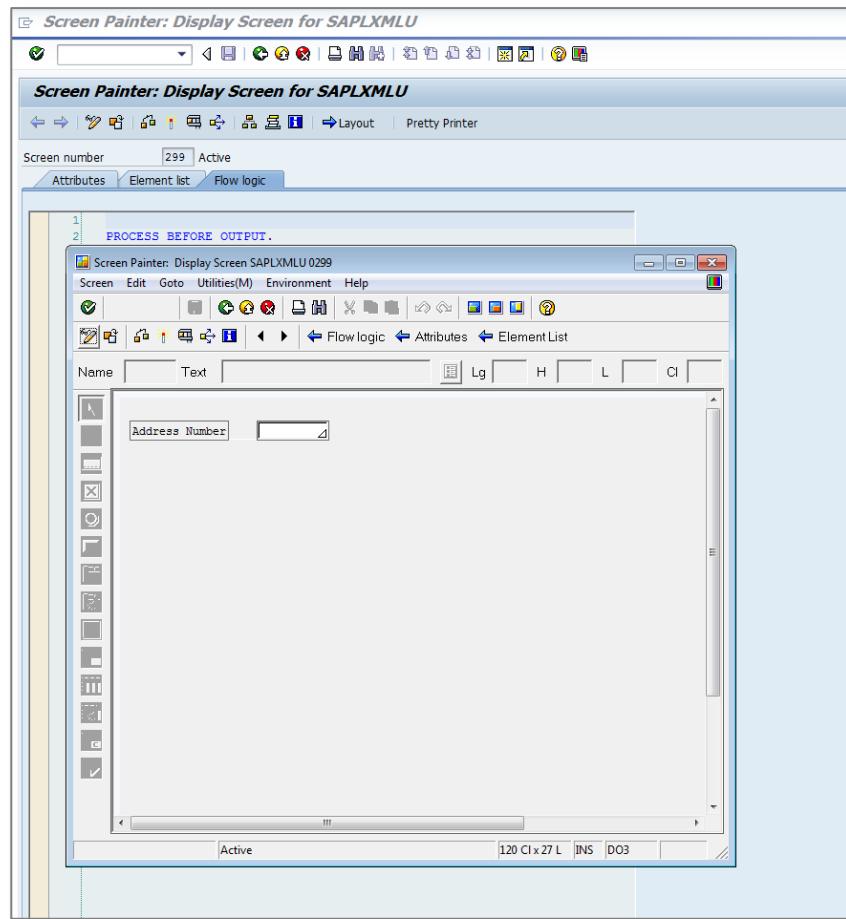


3. Implement the following exits within the enhancement. This includes the screen changes to add the custom field for override address. Sample code changes are also provided below to show how this works as a reference. This can be implemented in different ways too according to customer needs.

| Display Z_SES_OA | | | | |
|-------------------------|------|-----|--|--|
| Enhancement assignments | | | | |
| Project | Impl | Exp | Z_SES_OA Project for Including Address Number at Service level in PO | |
| Enhancement | Impl | Exp | SRVDET User screen on tab strip of service detail screen | |
| Function exit | ✓ | ✓ | EXIT_SAPIMLSP_040 EXIT_SAPIMLSP_041 | |
| Screen exit | ✓ | ✓ | SAPIMLSP 0220 SUSCUSER SAPIXMLU 0299 | |
| Include tables | ✓ | ✓ | CI_ESLLDB | |

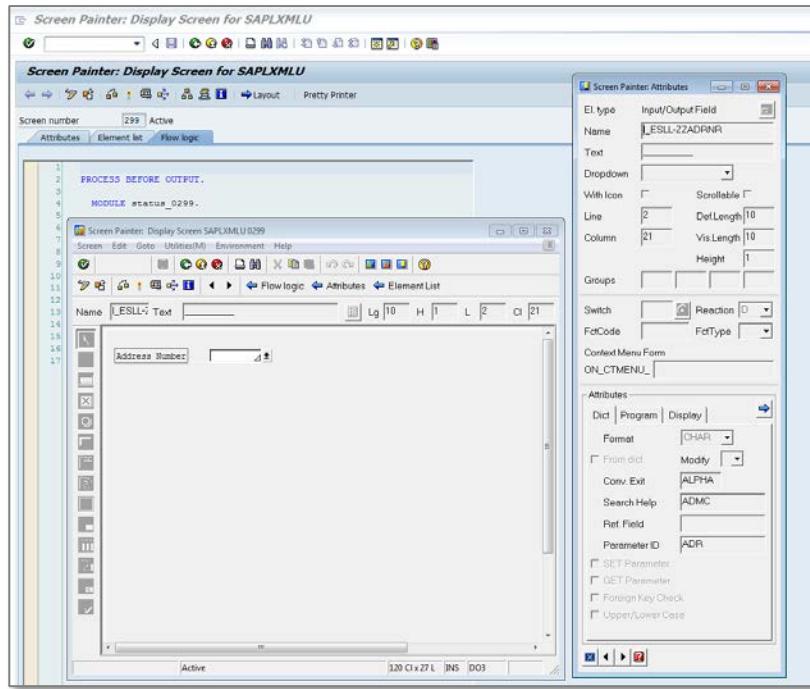
Support of Override Address - Address Number at Service Level in PO and ML81N

Screen shot of Screen Painter: Display Screen for SAPLXMLU



Support of Override Address - Address Number at Service Level in PO and ML81N

Screen shot of Screen Painter Attributes panel.



Sample code inside EXIT_SAPLMLSP_040 in include ZXMLUU23

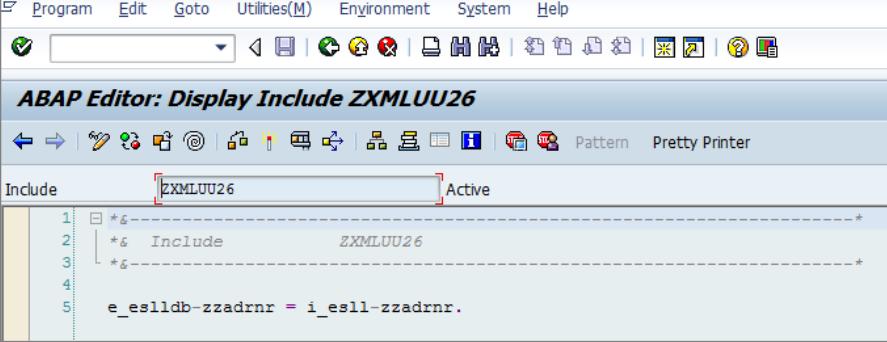
```

1  *->
2  *- Include      ZXMLUU23
3  *->
4
5  DATA: ms_esll TYPE esll.
6  FIELD-SYMBOLS: <fs_esll> TYPE esll.
7  UNASSIGN: <fs_esll>.
8  CLEAR: g_esll_zzadrnr,ms_esll.
9
10
11  ASSIGN ('(SAPLMLSP)ESLL') TO <fs_esll>.
12  IF sy-subrc EQ 0 AND <fs_esll> IS ASSIGNED.
13
14  IF <fs_esll>-zzadrnr IS NOT INITIAL.      "From Screen
15  g_esll_zzadrnr = <fs_esll>-zzadrnr.
16  ELSE.  "Check to see if source structure has data already in the override field. (Eg: Populate PO data to ML81N.)
17  IF i_esll-pln_packno IS NOT INITIAL AND i_esll-pln_introw IS NOT INITIAL.
18
19  SELECT SINGLE * FROM esll INTO ms_esll WHERE packno = i_esll-pln_packno
20  AND introw = i_esll-pln_introw.
21  IF sy-subrc EQ 0 AND ms_esll-zzadrnr IS NOT INITIAL.
22  g_esll_zzadrnr = ms_esll-zzadrnr.
23  ENDIF.
24  ELSEIF i_esll-zzadrnr IS NOT INITIAL.
25  g_esll_zzadrnr = i_esll-zzadrnr.
26  ENDIF.
27  ENDIF.
28  ENDIF.

```

Support of Override Address - Address Number at Service Level in PO and ML81N

Sample code inside EXIT_SAPLMLSP_041 on include ZXMLUU26

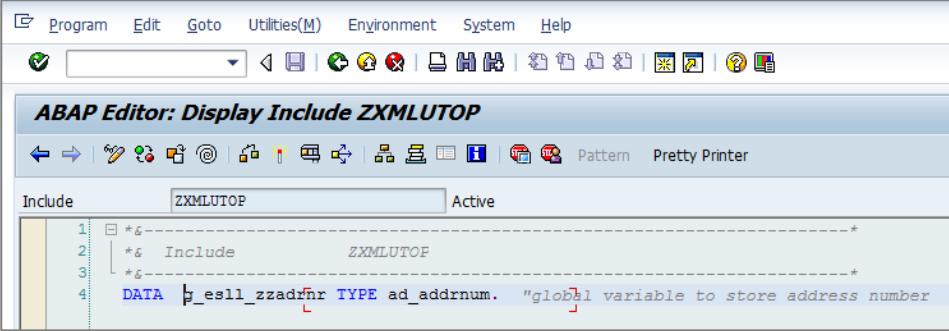


```

1  *&-----+
2  *&  Include      ZXMLUU26
3  *&-----+
4
5  e_es11db-zzadrnr = i_es11-zzadrnr.

```

Code inside include ZXMLUTOP for declaring the global variable.

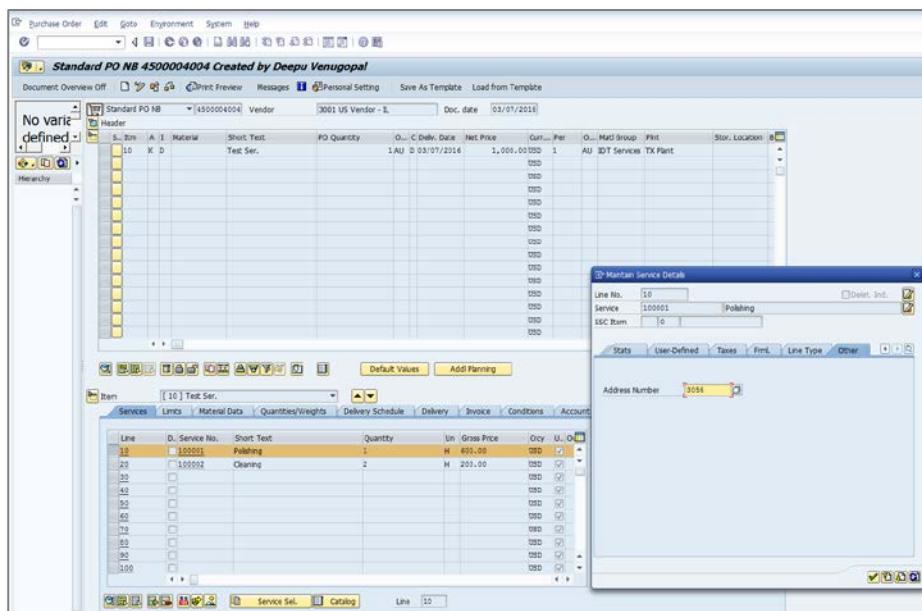


```

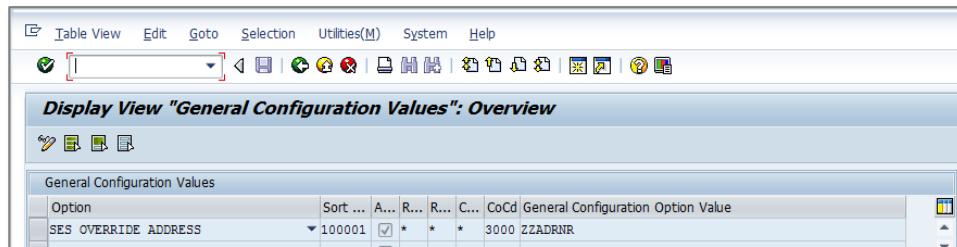
1  *&-----+
2  *&  Include      ZXMLUTOP
3  *&-----+
4  DATA b_es11_zzadrnr TYPE ad_addrnum. "global variable to store address number

```

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

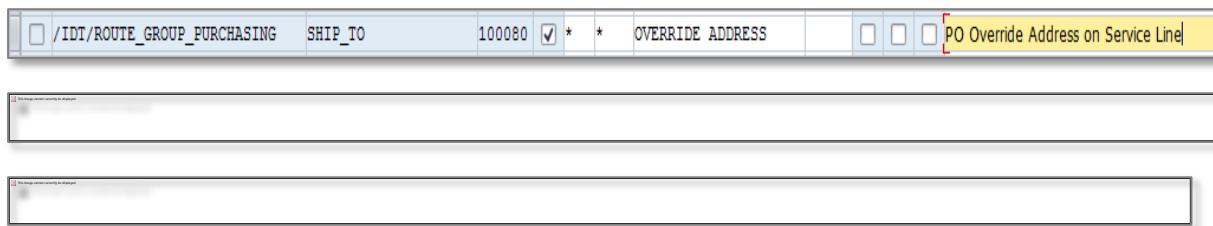


5. An entry for the new override address field for SES also needs to be added to the /IDT/D_GENERAL table as shown below.



| Option | Sort ... | A... | R... | R... | C... | CoCd | General Configuration Option | Value |
|----------------------|----------|-------------------------------------|------|------|------|------|------------------------------|-------|
| SES OVERRIDE ADDRESS | 100001 | <input checked="" type="checkbox"/> | * | * | * | 3000 | ZZADRNR | |

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.

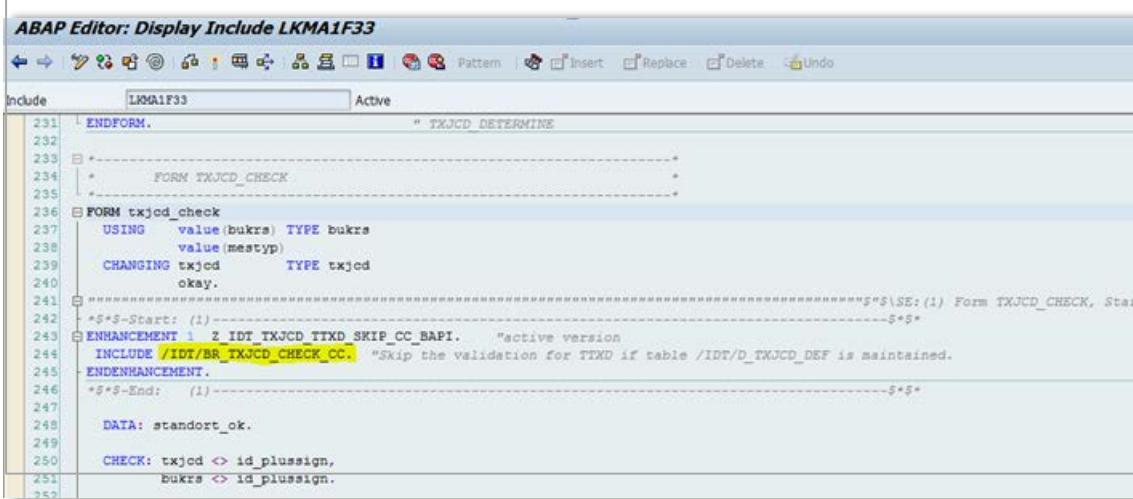
BRAZIL: TAX JURISDICTION CODE FOR COST CENTER CREATED THROUGH BAPI

This is an optional enhancement point that a user can elect to implement in order 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 our new 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 new implicit enhancement is required in the include LKMA1F33 to handle the BAPI call K_COSTCTR_BAPI_CREATEMULTIPLE.

India optional Hooks/Include Statements for Transaction J_1IG_INV

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.



```

ABAP Editor: Display Include LKMA1F33
Include LKMA1F33 Active
231 1 ENDFORM.                                     " TXJCD DETERMINE
232
233 2 *----- FORM TXJCD_CHECK -----*
234 3 | *----- FORM TXJCD_CHECK -----*
235
236 4 FORM txjcd_check
237 5   USING   value(bukrs) TYPE bukrs
238 6         value(mestyp)
239 7   CHANGING txjcd      TYPE txjcd
240 8   okay.
241 9 *----- $*S-Start: (1) -----$*S
242 10 *----- ENHANCEMENT 1 2 IDT TXJCD TTXD SKIP_CC_BAPI. "active version
243 11 INCLUDE /IDT/BR_TXJCD_CHECK_CC. "Skip the validation for TTXD if table /IDT/D_TXJCD_DEF is maintained.
244 12 ENDDOCUMENTATION.
245 13 *----- $*S-End: (1) -----$*S
246 14 DATA: standort_ok.
247
248 15 CHECK: txjcd <> id_plussign,
249 16           bukrs <> id_plussign.
250
251

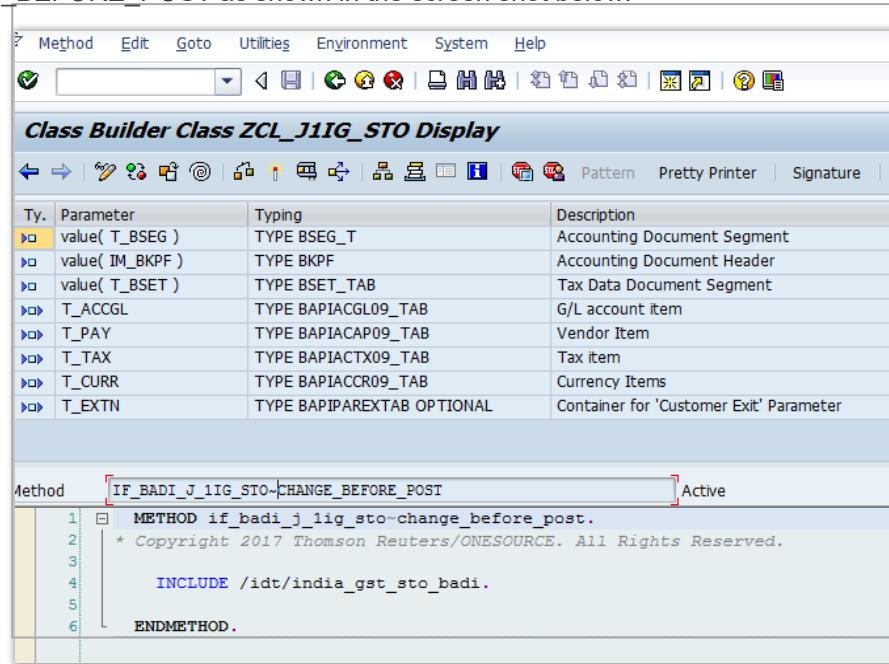
```

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.



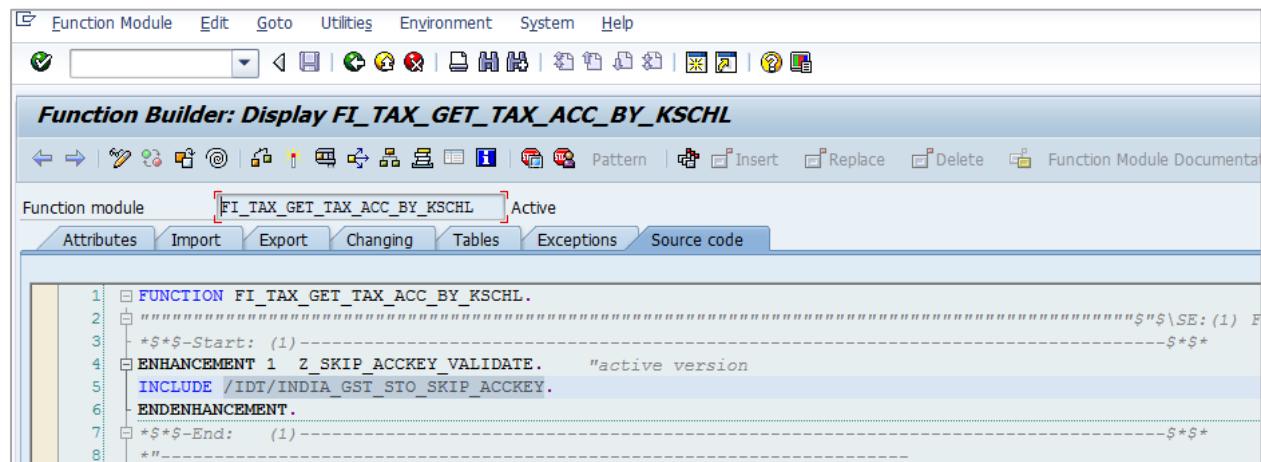
```

Method IF_BADI_J_1IG_STO~CHANGE_BEFORE_POST Active
1: METHOD if_badi_j_1ig_sto-change_before_post.
2: * Copyright 2017 Thomson Reuters/ONESOURCE. All Rights Reserved.
3:
4: INCLUDE /idt/india_gst_sto_badi.
5:
6: ENDMETHOD.

```

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.



```

Function module FI_TAX_GET_TAX_ACC_BY_KSCHL Active
Attributes Import Export Changing Tables Exceptions Source code

1: FUNCTION FI_TAX_GET_TAX_ACC_BY_KSCHL.
2: *$*$-Start: (1) -----$$$: (1) F
3: ENHANCEMENT 1 2_SKIP_ACCKEY_VALIDATE.      "active version
4: INCLUDE /IDT/INDIA_GST_STO_SKIP_ACCKEY.
5: ENDENHANCEMENT.
6: *$*$-End: (1) -----$$$*
7:
8: "

```

APPENDIX 1: 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. |
| /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. |
| /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_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. |
| /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 |

List of Journeys

| | |
|--------------------------------|--|
| | 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 |
| /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 <i>Configuration Guide</i> 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. |
| /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 in order 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. |

LIST OF ROUTES

Routes can be basically 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 in agreement. |
| /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. |
| /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. |

| Route | Description |
|--------------------------------|--|
| /IDT/ROUTE_NON_GROUP_DOC_DT | This Route is to handle the complexity of transactions with deferred tax. |
| /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. |

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 of 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 |
| | BKPF (FI/LIV) VBRK (SD) T001 for both processes | /IDT/JOURNEY_AUDIT_UPD_DB_GL /IDT/JOURNEY_AUDIT_UPD_DB_BILL |
| | 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, MAK, 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 |
| | BKPF (FI/LIV) VBRK (SD) | /IDT/JOURNEY_AUDIT_UPD_DB_GL /IDT/JOURNEY_AUDIT_UPD_DB_BILL |
| | 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 | All |

Source Bases

| | | |
|----------|---|---|
| | GC_VERSION – represents the XS | |
| 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 |
| 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 |
| | 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

| Operand | Description | Journey supported |
|-------------|--|---|
| CALC_HDR | ROLE, EXTERNAL_COMPANY_ID, UNIQUE_INVOICE_NUMBER, TAX_CATEGORY | /IDT/JOURNEY_HEADER_REQUEST |
| CALC_ITEM | DET_TAX_CODE, IS_EXEMPT, AMOUNT, QUANTITY, IS_CREDIT | /IDT/JOURNEY_ITEM_REQUEST, /IDT/JOURNEY_NG_ITEM_REQUEST, |
| PARTNER_TAB | Used to reference a partner address in | /IDT/JOURNEY_HEADER_REQUEST, |

Special Purpose Operands

| | | |
|------------|--|---|
| | field mappings. Can be used in combination with partner function type. | /IDT/JOURNEY_ITEM_REQUEST, /IDT/JOURNEY_NG_ITEM_REQUEST |
| 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. | DT/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 |
|---------|-------------|
|---------|-------------|

Special Purpose Operands in Tables

| | |
|---------------------|---|
| 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_PROXYSES | 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 |
| /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 |

| Table | Description |
|------------------|--|
| /IDT/D_PLNTS_ABD | Billing documents types that signify a plants abroad process |
| /IDT/D_PROXYIES | 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_CTRL | FI Control Process: Standard |
| /IDT/S_LOG_CONF | 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 |
| /IDT/V_AS_PRIVAT | SAP Non-Partner Function Address Sources: Standard |
| /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_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 |

| Table | Description |
|------------------|---|
| /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 |

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 |

| Route | Description |
|-------|---|
| | 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 |
| 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 |

| Transaction code | Used for |
|----------------------|--|
| /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 |
| /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 |

| Transaction code | Used for |
|---------------------|---|
| /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 |
| /IDT/US_TAX_REPORT | Copy of US Tax 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

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-BKXTXT |
| 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 |
| Document Date | Document date in format : DD-MON-YYYY, E.g. 12-Aug-2012 | ETXDCI-ZZINVDATE if not NULL, else ETXDCI-TAX_DATE |
| 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 |

| Output field name | SAP field name | Table-fieldname |
|-------------------|--|-----------------|
| Document currency | Document currency key | BKPF-WAERS |
| Optional UDF 1 | Custom field based on custom BAPI implementation | |
| Optional UDF 2 | | |
| Optional UDF 3 | | |
| Optional UDF 4 | | |
| Optional UDF 5 | | |

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 | |
| Fiscal Year | BKPF-GJAHR | Parameter | R | |
| Posting Date | BKPF-BUDAT | Select options | O | |
| Posting Period | BKPF-MONAT | Select options | O | |
| Tax Code | BSEG-MWSKZ | Select options | O | |
| Select Zero/Exempt tax records | | Check Box | O | X |
| Application / Local server path | | Radio Buttons | O | Application server |
| Application Server | RLGRAP-Filename | Parameter | O | TVARV variable value |
| Presentation Server | RLGRAP-Filename | Parameter | O | User parameter value |

| Description | Field Name | Type | Optional / Required | Default value |
|----------------------|------------|-----------|---------------------|---------------|
| Company Code Prepend | | Parameter | O | |
| Package Size | | Parameter | R | 10000 |

ONESOURCE INDIRECT TAX TRANSPORT OBJECTS

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 |
| /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 | BAdl definition | BAdl definition to add logic for 5 user defined fields |
| /IDT/EXTRACT_OUTPUT | Structure | Output file structure |
| /IDT/INT_RECON_EXTRACT | Interface | BAdl Interface |
| /IDT/CL_RECON_EXTRACT | Class | Class implementing the BAdl interface |
| /IDT/EXTRACT_UDF | Structure | BAdl 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

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**.

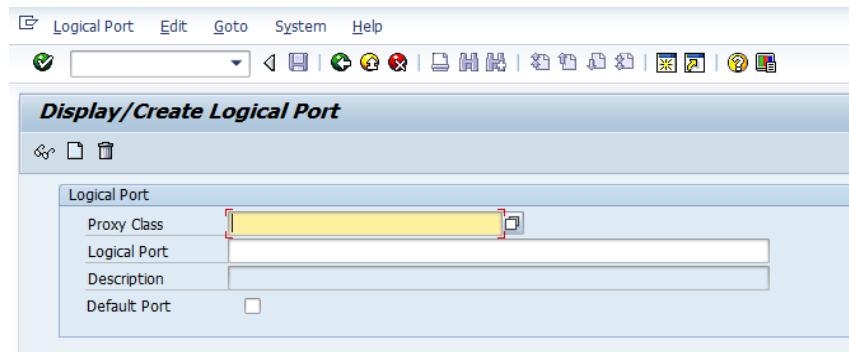
APPENDIX 2: LOGICAL PORT

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.

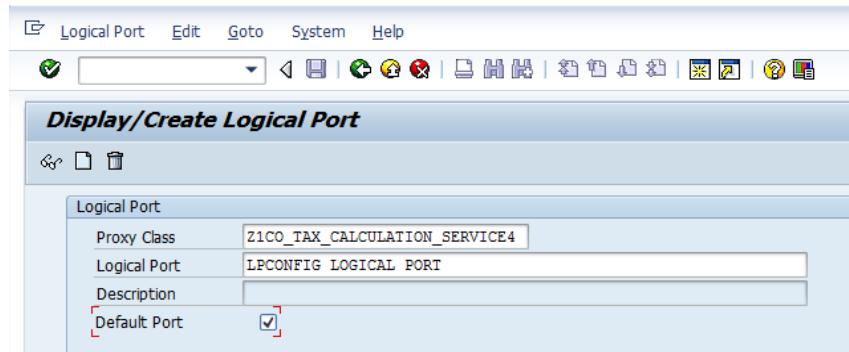


We have not extensively tested this method, but provide below setup screens as a courtesy in case you would like to use this alternate way of setup. We don't recommend using it as a production setup.

1. Transaction Code: **LPCONFIG**



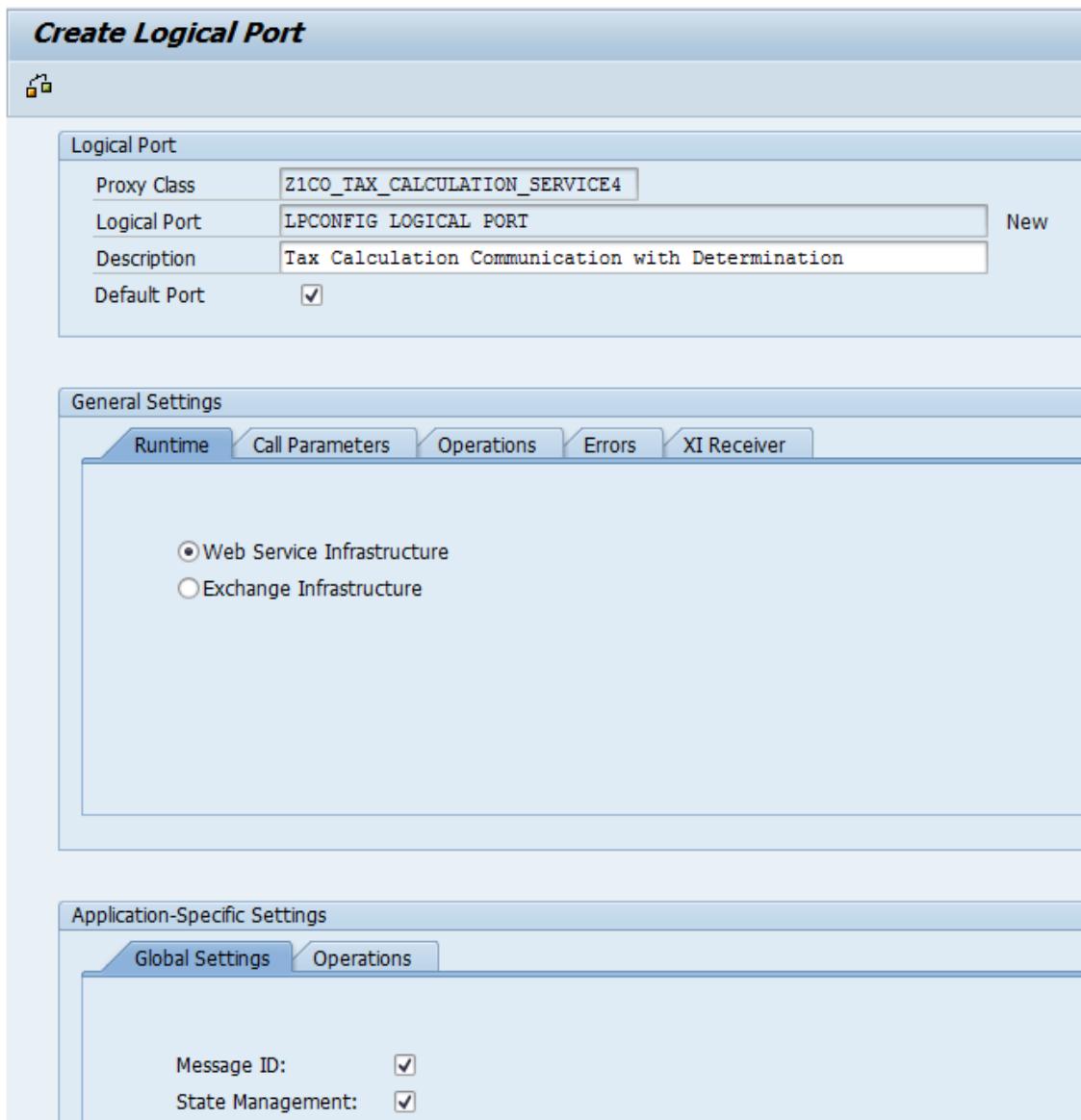
2. Use the Proxy Name from [Creating an SAP Proxy](#) section and create a Logical Port Name. Then Check “Default Port”.



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`

Create Logical Port

Logical Port

| | |
|--------------|--|
| Proxy Class | Z1CO_TAX_CALCULATION_SERVICE4 |
| Logical Port | LPCONFIG LOGICAL PORT |
| Description | Tax Calculation Communication with Determination |
| Default Port | <input checked="" type="checkbox"/> |

General Settings

Runtime Call Parameters Operations Errors XI Receiver

HTTP Destination

 URL

 Local Path Prefix

Binding Type

Application-Specific Settings

Global Settings Operations

Message ID:
 State Management:



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.