ONESOURCE™ SAP COMMERCE INTEGRATION

SETUP, CONFIGURATION AND USER GUIDE

VERSION NUMBER (FROM TARGET)

Document Version 1

THOMSON REUTERS®

DOCUMENT HISTORY

VERSION NUMBER	VERSION DATE	SUMMARY
V1	November 2020	Initial release

TABLE OF CONTENTS

INTEGRATION OVERVIEW

This is an overview of the TAX calculation integration between SAP Commerce and ONESOURCE Indirect Tax Determination (Determination). Thomson Reuters ONESOURCE™ Indirect Tax offers a comprehensive, cloud-based transaction tax management solution that seamlessly integrates for accurate sales tax calculation, easy document management, effortless filing and remittance.

This integration will support the following events from the SAP Commerce Platform.

- Estimate Tax Calculation Call for Carts and Orders
- Audited Tax Calculation Call for Orders

As of version 1.0.0.0, the solution supports the following countries for tax integration:

- United States
- Canada

Prerequisites and Steps

For a successful implementation of the ONESOURCE IDT Integration with SAP Commerce, ensure you have procured the necessary packages and followed the implementation steps mentioned below:

PLATFORM	COMPONENT	DESCRIPTION
SAP Commerce	Local environment	• ONESOURCE IDT Integration extensions to be added to the project setup • Building and updating the project to activate the extensions. • Maintenance of system configuration for connecting to the ONESOURCE IDT system.
One Source Determination	Configuration	• The External Company ID setup that uniquely identifies the tax request within the Thomson Reuters solution for audit, reporting, and returns purposes. • Username and Password required to send tax calculation requests

Benefits

Integration

ONESOURCE Indirect Tax Integration extensions seamlessly connect your SAP Commerce system to ONESOURCE Determination for tax calculations and appropriate return of tax results. The extensions 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 your existing e-commerce systems using our open integration architecture. Tax calculation calls can be easily inserted into existing system workflows and processes to deliver real-time solutions with accurate tax results. This integration is fully embedded in the SAP Commerce architecture. It can easily be deployed in SAP Public Cloud environments and can be configured and troubleshot on its native user interfaces like Administration Console, Backoffice and Kibana.

In case you have specific requirements, it is also possible to easily extend it just like out of the box SAP extensions by your own project resources or Thomson Reuters Professional Services team.

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 life cycle 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.

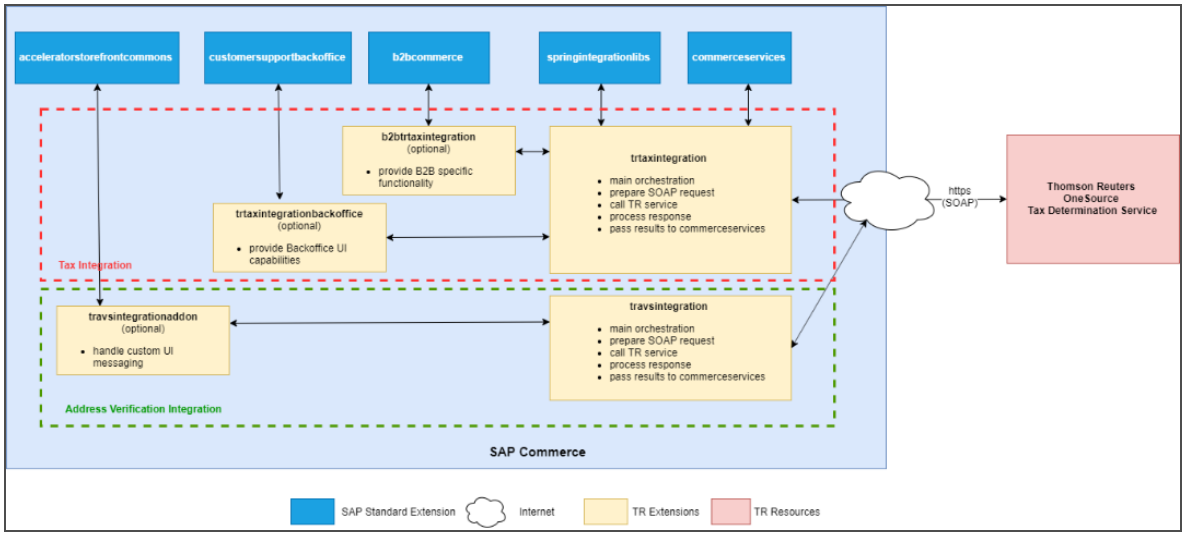
Reporting

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 with each state's unique electronic filing requirements. The software also goes beyond borders to include the returns required for tax compliance in Canada.

Architecture and Design Overview



In order to enable external tax calls to ONESOURCE IDT, Thomson Reuters provides following extensions to be included in your SAP Commerce setup.

For tax integration:

- **trtaxintegration**: This extension is mandatory for the integration and it supports both B2C and B2B scenarios. It requires the following SAP extensions: commerceservices and springintegrationlibs
- **b2btrtaxintegration**: This extension is optional and should only be used if you have a B2B Storefront. It requires the following SAP extension: b2bcommerce
- **trtaxintegrationbackoffice**: This extension is optional and should only be used if you would like to use Backoffice UI enhancements provided within the package. It requires the following SAP extension: customersupportbackoffice

In addition, if you plan to activate address verification integration, the following extensions should also be included.

- **travsintegration**: This extension is mandatory for the integration and it supports both B2C and B2B scenarios. It requires the following SAP extensions as prerequisites: commerceservices, commercefacades and springintegrationlibs
- **travsintegrationaddon**: This extension is optional and should only be used if you have Accelerator-based storefronts on the UI layer. It requires the following SAP extensions as prerequisites: acceleratorstorefrontcommons, addonsupport

For all integration touchpoints, the call is always triggered and managed by the backend layer (SAP Commerce application servers). The solution provides no frontend (Javascript) integration.

Assumptions and Known Limitations

- ONESOURCE IDT Integration for SAP Commerce is only supported for SAP Public Cloud Deployments. Although it may be assumed that the extensions provided would work in an on-premise environment, the functionality has not been tested in an on-premise cluster.
- The integration supports sending multiple ship-to addresses in a cart/order. Since this functionality is not supported by the out-of-the-box accelerators, it is assumed that the custom implementation in the client supports necessary checkout steps to get this address info from the user and persists this information on the cart. Details can be found in the following sections.
- Only net pricing is supported.
- The tax integration is only limited to the checkout pages of the storefront. No tax calls will be triggered on other pages of the storefront.
- Address Verification Integration only supports verification of United States addresses.
- The address validation integration is only limited to the storefront. There is such integration in Backoffice UI as part of this integration package.

INSTALLATION

Local Environment Installation

- Extract all extension folders (trtaxintegration, trtaxintegrationbackoffice and b2btrtaxintegration) into the “$HYBRIS_HOME/bin/custom” folder
- Add the following lines to your localextensions.xml file. ’trtaxintegration’ is mandatory for the integration while the others are optional.
    - *<extension name=’trtaxintegration’ />*
    - *<extension name=’b2btrtaxintegration’ />*
    - *<extension name=’trtaxintegrationbackoffice’ />*
- For address verification, add the following lines to your localextensions.xml file. *’travsintegration’* is mandatory for the integration while the other is optional.
    - *<extension name=’travsintegration ’ />*
    - *<extension name= travsintegrationaddon />*
- If you added *travsintegrationaddon*, run the following ant command to install the addon for the required storefront extension.

Ex: ant addoninstall -Daddonnames="travsintegrationaddon" -DaddonStorefront.yacceleratorstorefront="{{myextension}}" (Please replace {{myextension}} with the name of the storefront extension in your setup)

- Build your system with “ant all"
- Update your system with “ant updatesystem”.

Activate External Tax Calls

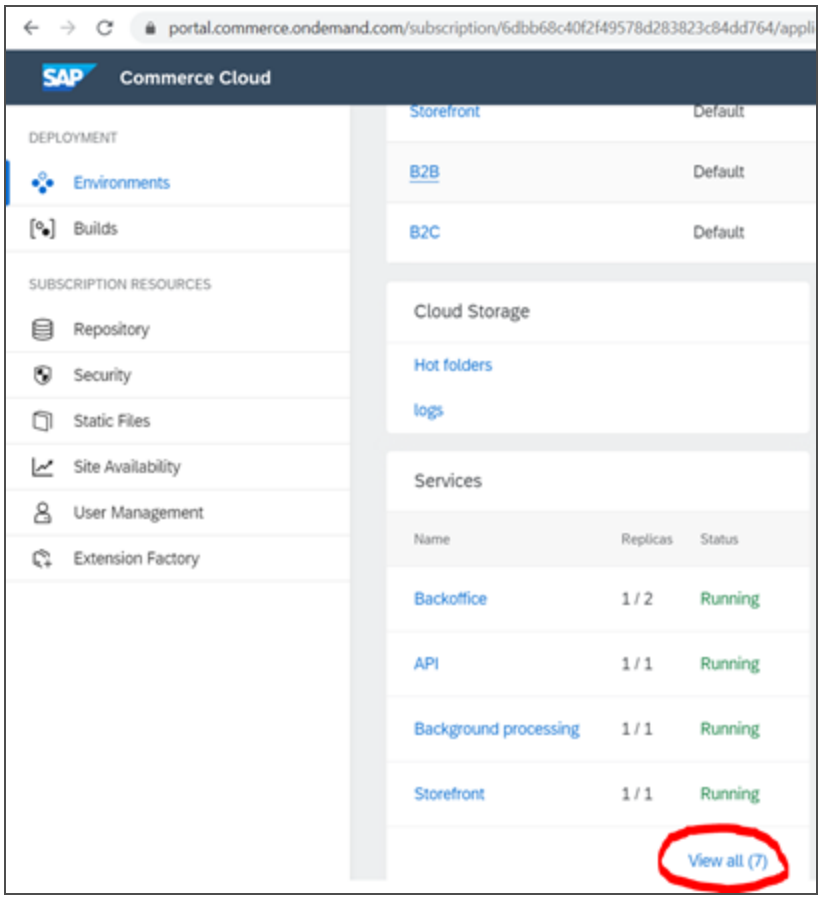
For ONESOURCE IDT external tax call to ONESOURCE IDT to be triggered in the system, the following prerequisites should be met:

- External Tax Calling should be activated for the base store (BaseStore.externalTaxEnabled should be true)

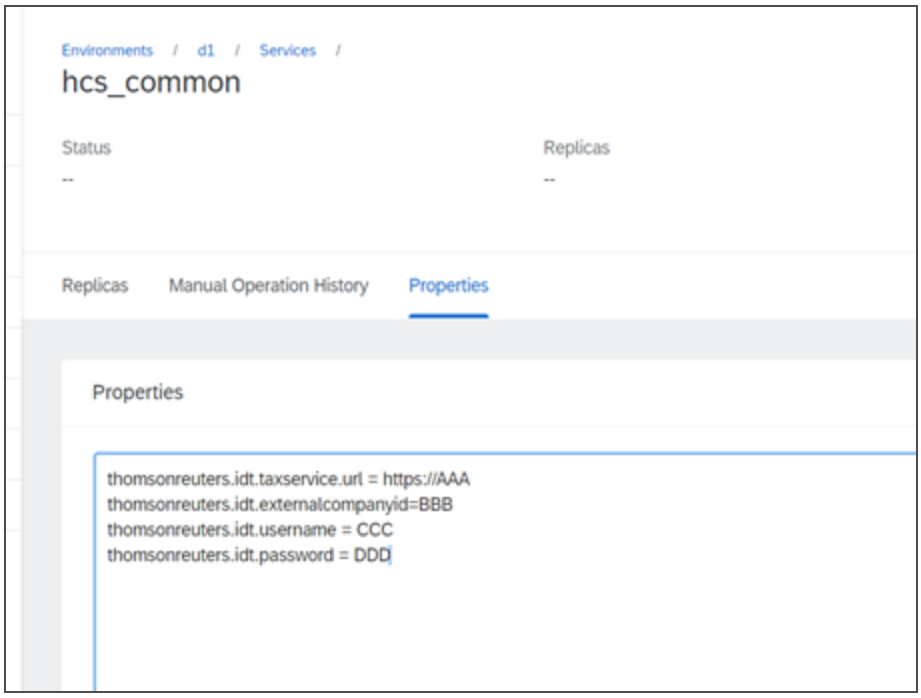
- External Company ID should be defined either in system configuration or in Base Store page in Backoffice. If a value is found in the base store settings, this value will override the system configuration parameter.
- Net pricing should be active for the base store (BaseStore.net field should be true)
- The cart should have a delivery address and a delivery mode assigned
- In case of a pickup in-store scenario, a point of service address should be maintained.

SAP Public Cloud Environment

- Make sure you committed extension folders in your code repository
- Update your manifest.json file to include the extensions in your setup. This file should be located in $HYBRIS_HOME folder. Also, make sure that your configuration properties described in section Configure SAP Commerce for ONESOURCE IDT Integration are included in your setup.
- Since it is a security best practice to not to keep any sensitive data in the code repository, you can provide the username and password necessary to connect to ONESOURCE IDT on the SAP Commerce Cloud Portal. Please carry out the following steps:
    - Login to https://portal.commerce.ondemand.com/ with your SAP credentials
    - Under Environments choose your environment
    - Click “View All” link in “Services” tile of your environment



- Choose “hcs_common” and “Properties” tab
- Provide your username and password in the following format:

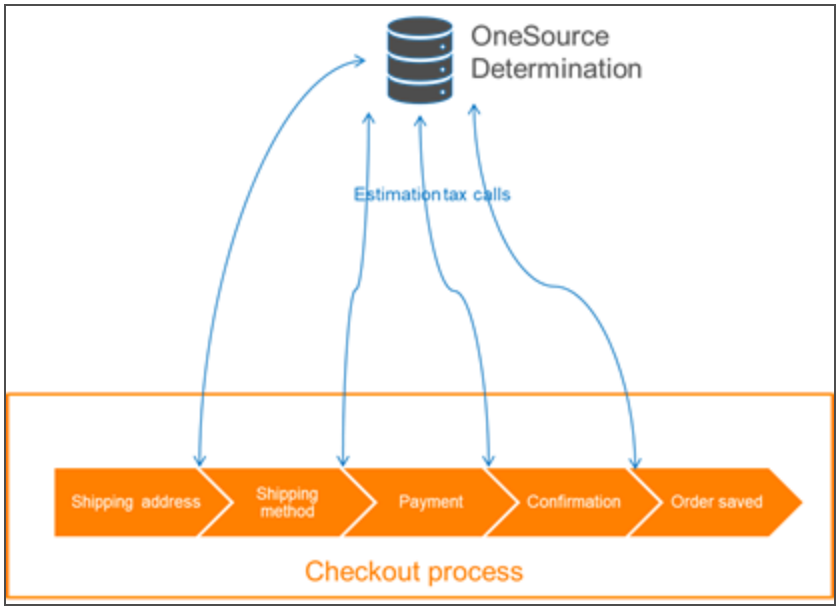


- Click "**Save**" button and then "**Apply Configurations**". Your application servers will be restarted with the provided values.

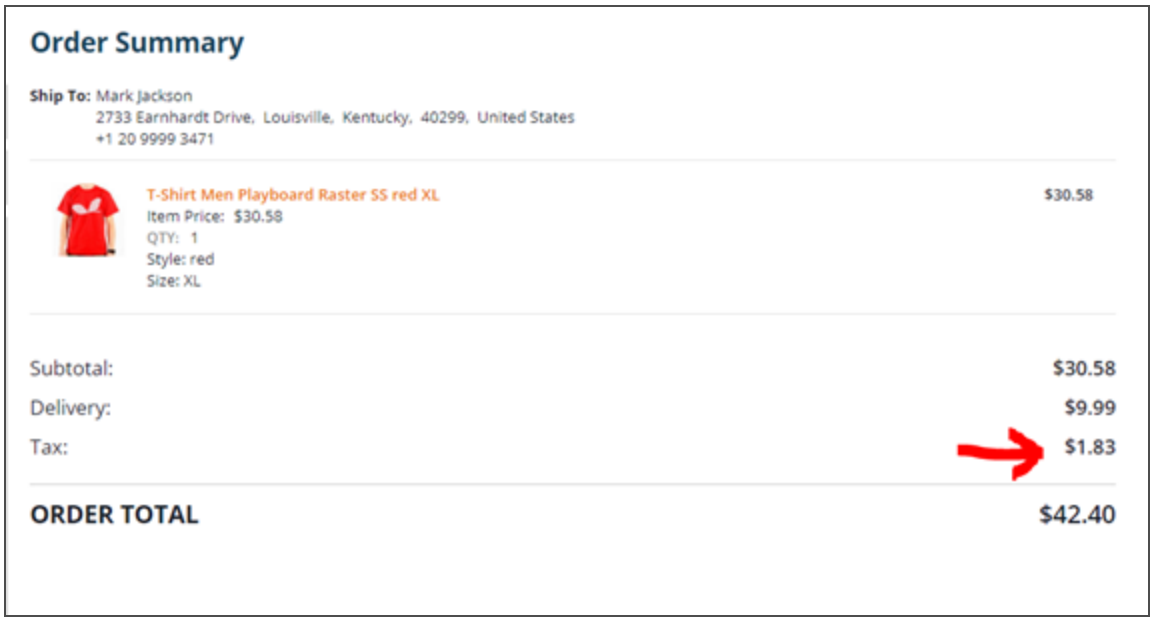
BUSINESS PROCESS DESCRIPTION

Main Business Processes

Checkout



During a checkout process both in B2C and B2B storefronts, several quote tax requests are sent to the Determination Engine in order to calculate the exact sales tax. After each call Determination response is read, summarized if necessary and persisted on the SAP Commerce cart object. The tax call occurs when the user clicks “Next” to proceed to the following step. The result for the total tax value is displayed in the “Order Summary” section of the next checkout step.

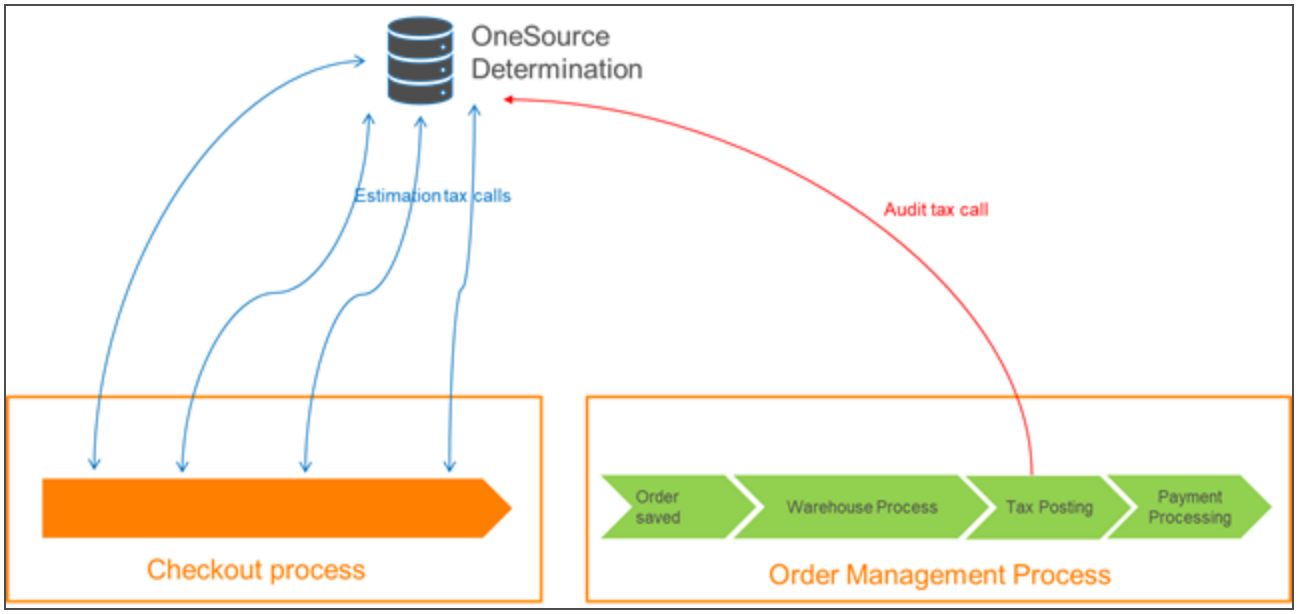


In this integration, the tax call is only triggered if any tax critical information is updated between checkout steps. If nothing is changed, no calls will be initiated.

Here are the major fields that affect tax calculation:

- Add or remove cart items
- Changes in existing cart items
    - Quantity
    - Price
    - Product
    - Delivery address
    - Delivery mode
    - Pickup store
    - Delivery address
    - Delivery mode
    - Delivery cost
    - Billing address
    - Header promotion

Order Processing



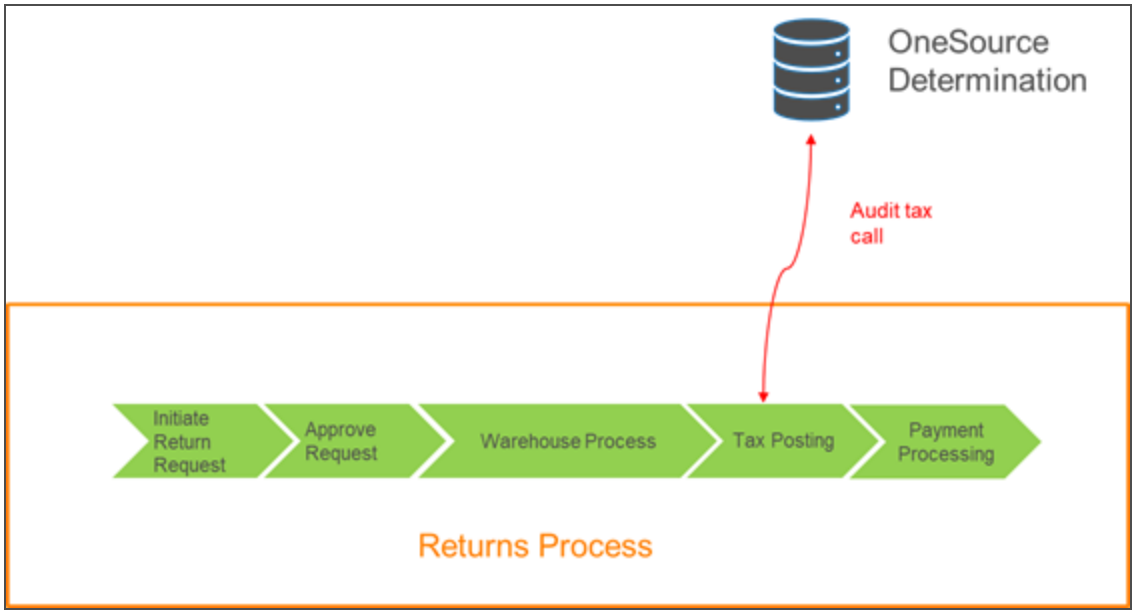
If the order fulfillment process is managed by SAP Commerce, it is possible to do the tax auditing for ONESOURCE Determination. On the other hand, if the order is sent to an external system for processing, this step may not be necessary.

Once the order is saved in SAP Commerce, an Order Business Process is triggered associated with this order which orchestrates all the tasks that need to be carried out for fulfilling the order. The major steps include e-mail notifications, assignment of orders to warehouses, warehouse process, tax postings and payment processing.

In this process flow, at the tax posting step, ONESOURCE Determination engine is called once again. Compared to the calls in the checkout process, the following differences should or might be observed in the integration:

- The tax call is audited, which means the document is persisted on the ONESOURCE Determination database.
- The tax call request is prepared with the last version of the order. Any change that might happen on the order from the first save is taken into account such as partial cancellations, order quantity updates or pricing / shipment cost changes.

Returns



If the return process is managed by SAP Commerce, it is possible to do the tax auditing for ONESOURCE Determination. On the other hand, if the returns process is not activated in SAP Commerce or the return is transferred to an external system for processing, this step may not be necessary.

Initiating the return request may happen in two channels: Storefront via returns self-service page or Customer Support agent using Customer Service Backoffice role. In both cases, a Return Request document is created in SAP Commerce and a Return Business Process is started in the background. Once the acceptance of the goods is completed in the warehouse, the necessary information to prepare a tax call becomes available.

Unlike the order process, a single call is made for a Return Request, which is an audit call. In this step, the taxes are calculated and persisted on the return request which can be used in the following refund payment steps. The record in the Determination Database is also created.

Other Functionality:

Canada Specific Functionality

For installations supporting Canada, it is possible to define buyer and seller registration numbers in the system.

- Seller Registration Numbers: “thomsonreuters.idt.registration.seller” system configuration has to be set as true. Multiple registration numbers can be defined for a base store in Backoffice. For this reason, a custom field is introduced to the Base Store model (tridtTaxRegistrationNumbers).
- Buyer Registration Numbers: “thomsonreuters.idt.registration.buyer” system configuration has to be set as true. Multiple registration numbers can be defined for a B2B Unit in Backoffice. For this reason, a custom field is introduced to the B2BUnit model (tridtTaxRegistrationNumbers).
- In a tax request, for all Canadian addresses, <PROVINCE> tag is used instead of <STATE>.

Taxes for the Delivery Cost

The ONESOURCE Determination integration supports tax calculation for delivery costs. This delivery cost can be sent in two different ways:

- Single delivery cost as an additional item : For each tax request an additional item is added to the order for the delivery.
    - AbstractOrder.deliveryCost field is used as the gross amount for this item.
    - The product/commodity code for the delivery cost item is determined by the “thomsonreuters.idt.request.shipping.productcode” system property.
    - Quantity will always be 1.
    - The tax results are kept as TaxValue’s in cart/order header.

- Distributed delivery cost per item: For each delivery related item in cart/order, an additional item is inserted in the request.
  - AbstractOrder.deliveryCost field is distributed to each delivery item proportionally based on their item value.
  - The product/commodity code and quantities are maintained the same as the first method.
  - The delivery tax items are associated with their original items so that the same tax calculation logic is applied both to the cart/order item and its respective delivery.
  - Pickup in store items are excluded from this logic since they are not delivery related. No delivery item is created for them and they are not taken into account in delivery cost distribution.
  - The tax results are kept as TaxValues in items rather than in the header. A prefix is added to the tax code (Delivery-State) to differentiate delivery cost tax values from product tax values. The summarization option valid in the system also applies for the delivery cost taxes.

Single delivery cost method is the default method in delivery cost tax calculation and should be preferred in most cases. On the other hand, the distributed delivery cost method allows more accurate tax calculation but comes with its own complexity.

Summarization Options for Tax Authority Details

The SAP Commerce ONESOURCE Integration includes details of all the Tax Authorities (Tax Blocks) associated with a line in the Determination response. If this level of details is kept as is, there will be several tax lines for each item in the cart/order and in the header (in case of a shipping cost). For this reason, it is possible to apply a summarization operation to these tax values and group them by using one of the available criteria, hence reducing the number of tax items in carts/orders in SAP Commerce.

Ex: Suppose the following taxes are calculated and returned by the Determination System for a cart having no delivery cost:

CART ENTRY NO:	TAX AUTHORITY	TAX VALUE ($)	ERP TAX CODE	AUTHORITY TYPE	ZONE LEVEL
1	A1	2.00	T1	State Sales/Use	State
1	A2	3.00	T1	County Sales/Use	County
2	A1	4.00	T1	State Sales/Use	State
2	A2	5.00	T1	County Sales/Use	County

2	A3	6.00	T2	County Sales/Use	County
3	A1	7.00	T1	State Sales/Use	State
3	A2	8.00	T1	County Sales/Use	County
3	A3	9.00	T1	County Sales/Use	County
3	A4	10.00	T3	County Rental	County

You can choose one of the following options for summarization and set to thomsonreuters.idt.response.summarization system property:

- **SummaryByErpCode**: The tax values are grouped by <TAX><ERP_TAX_CODE> tag in the response. In other words, the tax values having the same ERP code are consolidated into a single tax value.

Ex: Here is the tax value structure you'll find in the cart in SAP Commerce:

CART ENTRY NO:	TAX CODE	TAX VALUE ($)
1	T1	5.00
2	T1	9.00
2	T2	6.00
3	T1	24.00
3	T3	10.00

- **SummaryByAuthority**: The tax values are grouped by <TAX><AUTHORITY_TYPE> tag in the response. In other words, the tax values having the same authority types are consolidated into a single tax value.

Ex: Here is the tax value structure you'll find in the cart in SAP Commerce:

CART ENTRY NO:	TAX CODE	TAX VALUE ($)
1	State Sales/Use	2.00
1	County Sales/Use	3.00
2	State Sales/Use	4.00
2	County Sales/Use	11.00

3	State Sales/Use	7.00
3	County Sales/Use	17.00
3	County Rental	10.00

- **SummaryByZone**: The tax values are grouped by <TAX><EFFECTIVE_ZONE_LEVEL> tag in the response. In other words, the tax values having the same zone level are consolidated into a single tax value.

Ex: Here is the tax value structure you'll find in the cart in SAP Commerce:

CART ENTRY NO:	TAX CODE	TAX VALUE ($)
1	State	2.00
1	County	3.00
2	State	4.00
2	County	11.00
3	State	7.00
3	County	27.00

- **FullDetails**: The tax values are not grouped at all and converted to TaxValue's as is.

Ex: Here is the tax value structure you'll find in the cart in SAP Commerce:

CART ENTRY NO:	TAX CODE	TAX VALUE ($)
1	A1	2.00
1	A2	3.00
2	A1	4.00
2	A2	5.00
2	A3	6.00
3	A1	7.00
3	A2	8.00

3	A3	9.00
3	A4	10.00

Handling Returns Process

With the integration package, a return specific business process definition (tridt-return-process) is provided that enhances the standard returns business process definition (return-process). In order to use this process base store setting should be updated. The relevant field is Return Request Process code (BaseStore.createReturnProcessCode).

Compared to return-process, tridt-return-process executes the tax call before the payment refund process. This way tax values can be taken into account when determining the refund amount. As a result, the following fields on the return request is updated:

- “Total tax” in Return Request Header: This value includes all product taxes and delivery cost related taxes (if exists)
- “Tax” in every Return Request Entry: This value is only for the product tax.

Due to the SAP Return Request data model, the detailed tax values are not being kept in the document, unlike orders. For this reason, summarization options (Ex: SummarizeByZone) that are available for orders, are not valid for return requests.

Note: SAP standard refund amount calculation logic does not include return request tax values into the total refund amount (class CaptureRefundAction). This implementation is not included in this package since the payment domain is out of this product’s scope. It is advised that the clients who would like to use this functionality, extend refund calculation logic to take taxes into account when calculating the totals.

Refunding the Delivery Cost in Returns

As per standard SAP Commerce logic, it is possible to refund the delivery cost in a return (Refund Delivery Cost field on Return Request). In this case, the tax calculation needs to be carried out for the delivery cost. In this process:

- Single delivery cost as an additional item method is used for the delivery cost.
- For technical reasons, the system performs two calls one after the other. First, an estimation call is made for the whole sales order. The results are used to prepare the tax request for the return and the return call (either estimation or an audit) immediately follows the first call.
- There is no summarization option for response processing in returns. The delivery cost tax is included in the total tax calculation and is not kept separately in the return request document.

Backoffice Enhancements

The enhancement in Backoffice UI within the ONESOURCE Determination Integration package is listed as follows:

Backoffice Administrator Role

- “Recalculate order totals” and “Calculate with promotions” action buttons in Orders page are deactivated if external taxing is activated in the base store assigned to the order.
- Two new buttons are introduced instead of the two actions above as “Recalculate order totals with external tax” and “Calculate with promotions with external tax” that will call Determination and update tax values in the order and respective order entries after executing out-of-the-box calculation logic.

Customer Support Agent Role

- The following actions existing on the Order Details page are deactivated as they are not relevant for this integration.
    - Manual Tax Void Action
    - Manual Tax Commit Action
    - Manual Tax Requote Action
    - Manual Delivery Cost Commit Action
- The following actions are added on the Order Details page are which are specific to ONESOURCE Determination integration.
    - Refresh Taxes Action (It the order is audit relevant, the request will be audited in Determination, otherwise it will be an estimation call. All Tax Values in the order and respective order entries will be refreshed and the total tax field in the order header will be recalculated.)

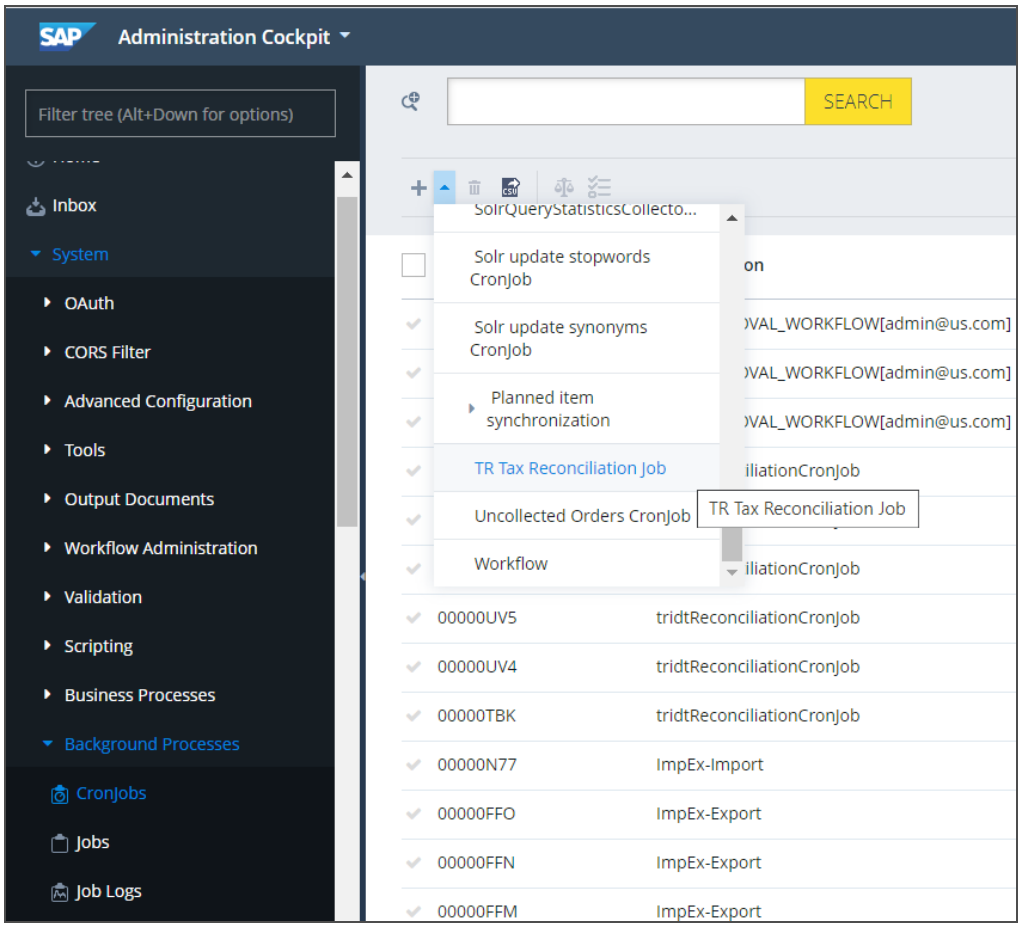
- Following actions existing on the Return Request Details page are deactivated as they are not relevant for this integration
    - Manual Tax Reverse Action
- The following actions are added on the Return Request page are which are specific to ONESOURCE Determination integration.
    - Refresh Taxes Action (The system will send an audit call to the Determination System. Tax fields in the return request and refund entries will be recalculated.)

Reconciliation Report

The integration package provides a reconciliation report that can be used in month-end closing. By running a cronjob, users can download a list of audited orders and returns in a specified period for a specified base store and compare the results with ONESOURCE Determination reports in order to identify inconsistencies between two systems.

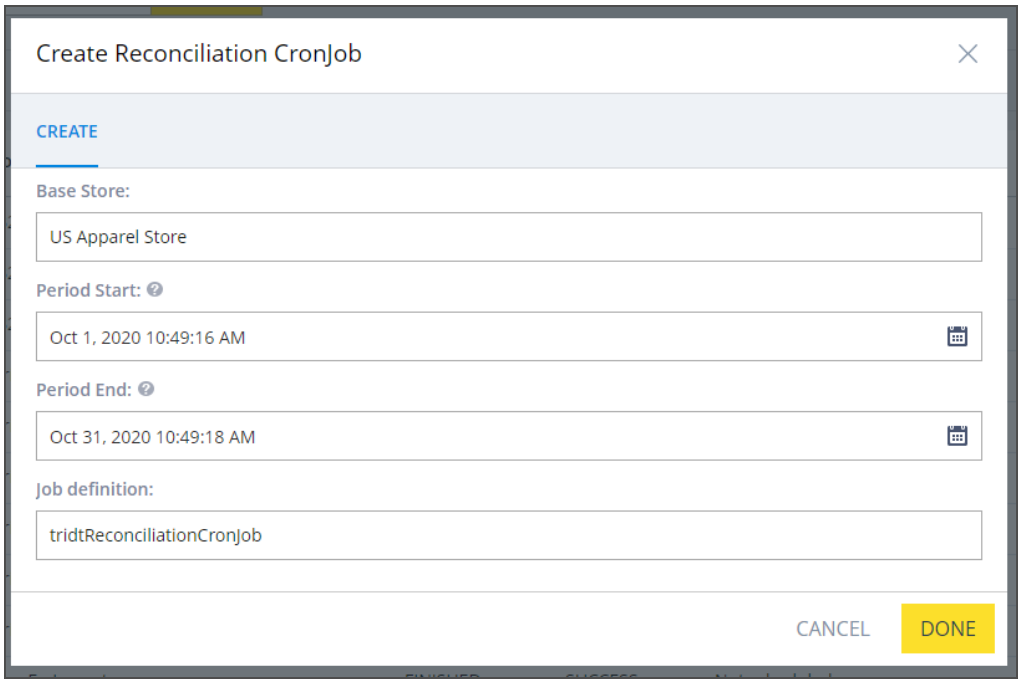
In order to initiate the cronjob:

- In Backoffice UI, go to System/ Background Processes / Cronjobs
- Create a new cronjob by clicking on the little triangle near the "Create new Cronjob" button and select TR Tax Reconciliation Job.

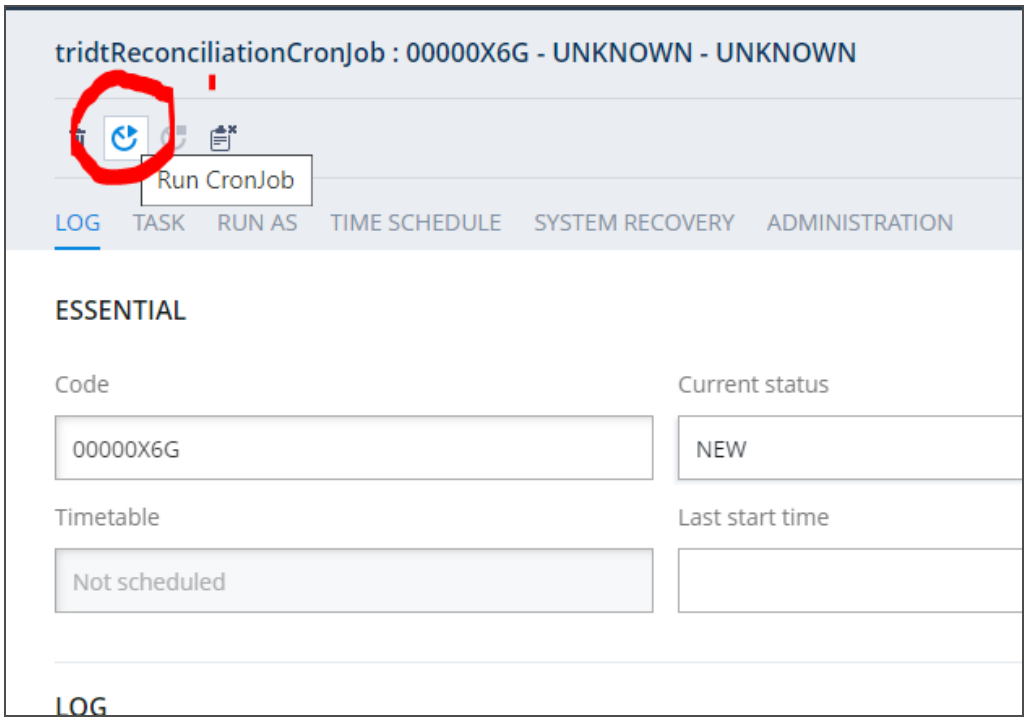


- Select a base store, period start and end dates for the interval. Then choose tridtReconciliationCronJob for job definition.
    - When selecting dates, please keep in mind that, the report will only consider the date portion of the selected value. The times will be adjusted to the start of the day for the period start and end of the day for the period end after the cronjob is started.
    - Since the dates are kept in the system timezone in SAP Commerce, the date chosen by the user is assumed to be in the system timezone.

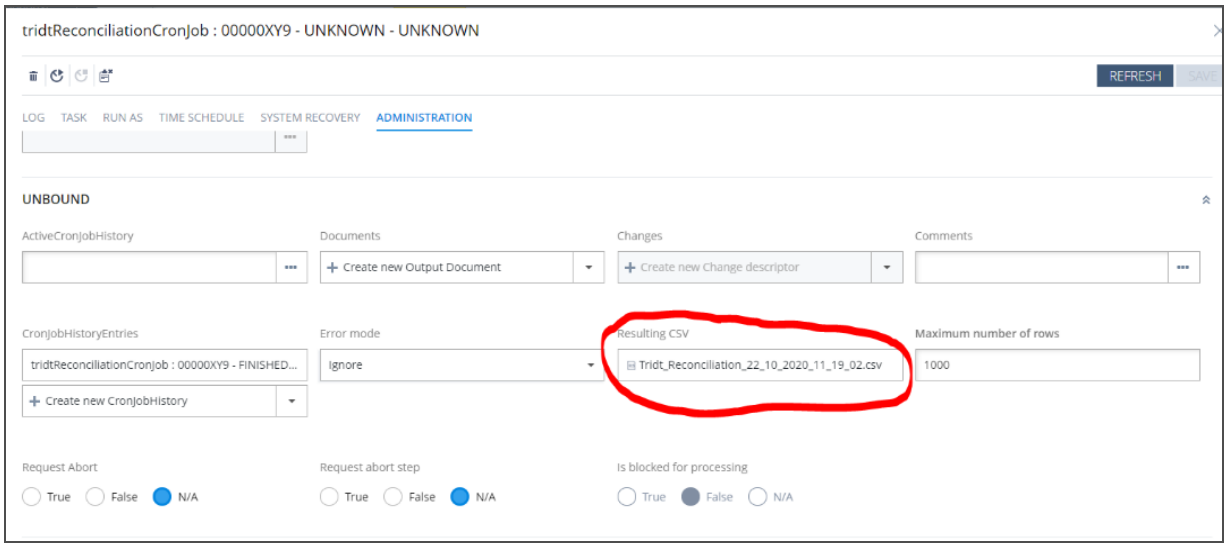
Ex: For a user in EST (browser timezone) on a system in CST, Oct 1, 2020 10:49:16 AM EST is adjusted as Oct 1, 2020 12:00:00 AM CST for a period start date.



- After clicking on Done, a new cronjob will be created but not run yet. In order to run the job, find the newly created cronjob and start it.



- If the job is completed successfully, you'll find a csv file in Resulting CSV field under Administration tab.



- You can double click on the file to download it to your local environment.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	External Company ID	Host System	Calling System	Company Role	ERP Transaction ID	Document Number	Document Type	Document Description	Document Partner Name	Document Partner Number	ERP Period	Document Date	Fiscal Date	Gross Amount	Tax Amount	Docume
2	INTSAPDEV_USCommerce	DEV	master	S	45000	45000	Invoice		Pronto Services	PS0002	Oct-20	6-Oct-20	6-Oct-20	233.5	20.09	USD
3	INTSAPDEV_USCommerce	DEV	master	S	46001	46001	Invoice		Pronto Services	PS0002	Oct-20	7-Oct-20	7-Oct-20	106	3.8	USD
4	INTSAPDEV_USCommerce	DEV	master	S	46003	46003	Invoice		Pronto Services	PS0002	Oct-20	7-Oct-20	7-Oct-20	154.5	9.66	USD
5	INTSAPDEV_USCommerce	DEV	master	S	46005	46005	Invoice		Pronto Services	PS0002	Oct-20	7-Oct-20	7-Oct-20	623	62.33	USD
6	INTSAPDEV_USCommerce	DEV	master	S	46007	46007	Invoice		Pronto Services	PS0002	Oct-20	8-Oct-20	8-Oct-20	308.5	18.65	USD
7	INTSAPDEV_USCommerce	DEV	master	S	52002	52002	Invoice		US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	490	35.54	USD
8	INTSAPDEV_USCommerce	DEV	master	S	52004	52004	Invoice		US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	558	40.46	USD
9	INTSAPDEV_USCommerce	DEV	master	S	52006	52006	Invoice		US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	988	71.64	USD
10	INTSAPDEV_USCommerce	DEV	master	S	55002	55002	Invoice		Pronto Services	PS0002	Oct-20	22-Oct-20	22-Oct-20	425	28.73	USD
11	INTSAPDEV_USCommerce	DEV	master	S	4000	4000	Return	Return for order 00045000	Pronto Services	PS0002	Oct-20	6-Oct-20	7-Oct-20	233.5	0.00E+00	USD
12	INTSAPDEV_USCommerce	DEV	master	S	4006	4006	Return	Return for order 00046001	Pronto Services	PS0002	Oct-20	7-Oct-20	7-Oct-20	43.5	3.8	USD
13	INTSAPDEV_USCommerce	DEV	master	S	4007	4007	Return	Return for order 00046003	Pronto Services	PS0002	Oct-20	7-Oct-20	7-Oct-20	139.5	9.66	USD
14	INTSAPDEV_USCommerce	DEV	master	S	4008	4008	Return	Return for order 00046005	Pronto Services	PS0002	Oct-20	7-Oct-20	8-Oct-20	613	61.3	USD
15	INTSAPDEV_USCommerce	DEV	master	S	6000	6000	Return	Return for order 00052006	US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	978	0.00E+00	USD
16	INTSAPDEV_USCommerce	DEV	master	S	6001	6001	Return	Return for order 00052002	US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	262	19	USD
17	INTSAPDEV_USCommerce	DEV	master	S	6002	6002	Return	Return for order 00052004	US Company	US_Company	Oct-20	21-Oct-20	21-Oct-20	279	0.00E+00	USD
18	INTSAPDEV_USCommerce	DEV	master	S	7000	7000	Return	Return for order 00055002	Pronto Services	PS0002	Oct-20	22-Oct-20	22-Oct-20	207.5	0.00E+00	USD
19																
20																
21																

- If the job can not find any orders or returns for the given parameters, the result of the cronjob will be set as ERROR.

CONFIGURATION OPTIONS

It is possible to change the default integration behavior by providing certain system properties in local.properties file for local installations and manifest.json file SAP Public Cloud Systems. Each option is described below in detail:

KEY	DEFAULT VALUE	DESCRIPTION	SCENARIO
Connection			
thomsonreuters.idt.taxservice.url	Empty	The value will be provided by Thomson Reuters	B2C/B2B
thomsonreuters.idt.externalcompanyid	Empty	The value will be provided by Thomson Reuters. The value in this parameter may be overridden if an External Company ID is defined for the base store of the cart/order.	B2C/B2B
thomsonreuters.idt.username	Empty	The value will be provided by Thomson Reuters	B2C/B2B
thomsonreuters.idt.password	Empty	The value will be provided by Thomson Reuters	B2C/B2B
thomsonreuters.idt.hostsystem	DEV	String that represents the role of the system. Ex: DEV, QA, PROD etc.	B2C/B2B
thomsonreuters.idt.callingsystemnumber	master	String that represents the calling system sending the transaction.	B2C/B2B
Address			

thomsonreuters.idt.address.postalcodeIncludesGeoCode	true	If it is true, XXXXX-YYYY format is expected on Address.postalCode field: where the first part will be used for <POSTAL_CODE> tag and the second part will be used for the <GEOCODE> tag in the request XML. If no hyphen is detected, the value will be used just for the <POSTAL_CODE> tag and <GEOCODE> will be left empty. If it is false Address.postalCode field is just used for the <POSTAL_CODE> tag and thomsonreuters.idt.address.geocode will be used to generate <GEOCODE> content in the request XML.	B2C/B2B
thomsonreuters.idt.address.sendGeoCode	false	If it is true, geocode will be included in the request. The value will depend on thomsonreuters.idt.address.postalcodeIncludesGeoCode and thomsonreuters.idt.address.geocode parameters.	B2C/B2B
thomsonreuters.idt.address.geocode	Empty	Any String typed fieldname (SAP standard or custom) on “Address” type can be provided as a parameter for this property	B2C/B2B
thomsonreuters.idt.address.county	Empty	Any String typed fieldname (SAP standard or custom) on “Address” type can be provided as a parameter for this property	B2C/B2B
thomsonreuters.idt.address.city	town	Any String typed fieldname (SAP standard or custom) on “Address” type can be provided as a parameter for this property	B2C/B2B

thomsonreuters.idt.address.district	district	Any String typed fieldname (SAP standard or custom) on “Address” type can be provided as a parameter for this property	B2C/B2B
thomsonreuters.idt.request.address.sendorderacceptanceaddress	false	If set to true, address details will be determined based on the tridtOrderAcceptanceAddress field on the Base Store. If the field is empty, the ship-from address is also used as the order acceptance address.	B2C/B2B
thomsonreuters.idt.request.address.sendorderoriginaddress	false	In B2C, if set to true, ship-to address will be copied to the order-origin address. In B2B, if set to true, address details will be determined based on the B2BUnit.contactAddress field. If the field is empty, similar to B2C, ship-to address will be copied to order origin address.	B2C/B2B
Product Code / Commodity Code Determination			
thomsonreuters.idt.request.item.sendCommodityCode	false	If set to true, <COMMODITY_CODE> tag will be sent for each line item. <PRODUCT_CODE> tag will be sent otherwise.	B2C/B2B

thomsonreuters.idt.request.item.taxcode.enhancedsearch.producthierarchy.enabled	false	If set to true, an enhanced search will be triggered to find product/commodity code starting from the product in cart and going upper levels if not found. E.g. First check the size variant's product master for a product/commodity code, if not found check the associated color variant and if not found check the base model.	B2C/B2B
thomsonreuters.idt.request.item.taxcode.enhancedsearch.categoryhierarchy.enabled	false	If set to true and if the search in product hierarchy doesn't provide any results, then an enhanced search will be triggered to find product/commodity code starting from the first level category and going upper levels if not found. E.g. First check the immediate categories assigned to the product in cart, if not found check the parent categories until a value is found.	B2C/B2B
thomsonreuters.idt.request.item.taxcode.fallbackCode	Empty	If no value is found for product/commodity code, this text will be used as a fallback code.	B2C/B2B
Delivery Cost			
thomsonreuters.idt.request.shipping.productcode	'FREIGHT'	This value will be used as the default product/commodity code for the delivery cost item.	B2C/B2B
thomsonreuters.idt.request.b2b.senddeliverymodeasincoterms	false	Only in B2B scenarios, the code of the DeliveryMode associated with the cart, can be sent in the <DELIVERY_TERMS>. This option can be used if Incoterm values are kept as DeliveryModes in SAP Commerce and the value chosen affects the tax calculation.	B2B

thomsonreuters.idt.request.shipping.sendmultipledeliverycostitems	false	If activated, a separate delivery cost item will be included in the request for each delivery related item in cart/order. Delivery cost on the header will be distributed to each delivery item based on the respective item value.	B2C/B2B
Canada specific			
thomsonreuters.idt.registration.seller	false	If activated, any registration number maintained in the base store will be included in the request as seller registration numbers.	B2C/B2B
thomsonreuters.idt.registration.buyer	false	If activated, any registration number maintained in the B2BUnit will be included in the request as buyer registration numbers.	B2B
Response Summarization			
thomsonreuters.idt.response.summarization	SummaryByZone	The value should be either of these four values: - SummaryByErpCode - SummaryByAuthority - SummaryByZone - FullDetails Details are described in section Summarization Options for Tax Authority Details	B2C/B2B
thomsonreuters.idt.response.shipping.prefix	'Delivery-'	This prefix will be used for naming every delivery cost related tax value.	B2C/B2B
Service Call Fallback			

thomsonreuters.idt.fallback.fallbackoncart	false	If activated, for cases where ONESOURCE Determination service is not reachable or integration call could not be completed due to a system error, a fallback rate will be used as an alternative in order not to stop the user from completing the checkout.	B2C/B2B
thomsonreuters.idt.fallback.fixedpercentage	Empty	The fixed-rate for the fallback process.	B2C/B2B
Logging			
thomsonreuters.idt.detailedlogging.request	true	If set to true, the details of the request XML will be saved in the logs.	B2C/B2B
thomsonreuters.idt.detailedlogging.response	true	If set to true, the details of the response XML will be saved in the logs.	B2C/B2B

DEVELOPMENT AND EXTENSIBILITY

Custom Attributes

ONESOURCE Determination enables companies to submit, and have returned, user-defined elements that are not directly supported in Determination. These elements can be used in conjunction with TransEditors (input filters) to enable custom tax calculations and are stored in the audit table for reporting purposes. Each <USER_ELEMENT> structure contains a single <NAME>/<VALUE> pair.

If you need to use these fields and send in the request payload, you need to extend the capabilities of trtaxintegration extension.

You can achieve this in five steps:

1. Decide on the hierarchy. Customer attributes can be sent either in:
    a. Header
    b. Product item
    c. Delivery cost items
2. In a custom extension add trtaxintegration as a required extension
3. In the custom extension, create a populator class implementing de.hybris.platform.converters.Populator with the following source and target parameters:
    a. For header attributes:
        i. source: de.hybris.platform.core.model.order.AbstractOrderModel
        ii. target: taxcalculationservice.wsdl.IndataType
    b. For product item attributes:
        i. source: com.thomsonreuters.idt.integrations.sapcommerce.models.TridtDiscountedOrderEntry
        ii. target: taxcalculationservice.wsdl.IndataLineType
    c. For delivery cost item attributes:
        i. source: de.hybris.platform.core.model.order.AbstractOrderModel
        ii. target: taxcalculationservice.wsdl.IndataLineType
4. Implement the business logic for each class. Add name/value pairs in taxcalculationservice.wsdl.UserElementType type to the respective structure. Currently restricted to the values ATTRIBUTE2 through ATTRIBUTE50. Note that ATTRIBUTE1 is already used in the standard logic in trtaxintegration extension.

5. Manage dependency injection in {{extension}}-spring.xml file
    a. Add a new bean for the populator. Ex:
        i. <alias name=**"{{populator bean ID}}"** alias=**"{{populator bean ID}}"** />
        ii. <bean id=**"{{populator bean ID}}"** class=**"{{Populator class path}}"**/>
    b. Add the populator bean to the converter bean as a dependency

For header attributes:

<alias name=**"{{header converter bean ID}}"** alias="tridtOrderConverter" />

<bean id=**"{{header converter bean ID}}"** parent="abstractPopulatingConverter">

<property name="targetClass" value="taxcalculationservice.wsdl.IndataType" />

<property name="populators">

<list>

<ref bean="tridtCommonOrderHeaderPopulator" />

<ref bean="tridtB2COrderHeaderPopulator" />

<ref bean=**"{{populator bean ID}}"** />

</list>

</property>

</bean>

For item attributes:

<alias name=**"{{item converter bean ID}}"** alias="tridtOrderItemConverter" />

<bean id=**"{{item converter bean ID}}"** parent="abstractPopulatingConverter">

<property name="targetClass" value="taxcalculationservice.wsdl.IndataLineType" />

<property name="populators">

<list>

<ref bean="tridtOrderItemPopulator" />

<ref bean=**"{{populator bean ID}}"** />

</list>

</property>

</bean>

For delivery cost item attributes:

<alias name=**"{{delivery cost item converter bean ID}}"** alias="tridtShippingItemConverter" />

<bean id=**"{{delivery cost item converter bean ID}}"** parent="abstractPopulatingConverter">

<property name="targetClass" value="java.util.ArrayList" />

<property name="populators">

<list>

<ref bean="tridtShippingItemPopulator" />

<ref bean=**"{{populator bean ID}}"** />

</list>

</property>

</bean>

Support for Omni Commerce Connect (OCC) REST API

The SAP Commerce ONESOURCE integration (estimation calls during checkout) can also be triggered by Omni Commerce Connect (OCC) web service calls. When an update on the cart is received via an OCC call, the prerequisites described in section Business Process Description - Checkout are evaluated to decide whether a new tax call is necessary just like for the call coming from an Accelerator Storefront.

Here are some examples that can trigger an external tax call:

- POST - /{baseSiteId}/users/{userId}/carts/{cartId}/addresses/delivery
- PUT - /{baseSiteId}/users/{userId}/carts/{cartId}/deliverymode
- POST/PUT - /{baseSiteId}/users/{userId}/carts/{cartId}/paymentdetails

LOGGING AND MONITORING

In the lifetime of a single tax call, several log messages are generated by the system and those can be displayed in the standard application log in SAP Commerce. The table below summarizes the logs that should be observed in the logs for an error-free tax call process.

STEP NO	LOG LEVEL	CLASS	TEXT	PARAMETERS	NOTES
1	INFO	DefaultTridtTaxCalculationService	Starting ONESOURCE Determination process...		
2	INFO	ClientFactory	Configuring system for company {1}	{1}: External Company ID	
3	DEBUG	TridtSoapLoggingService	Log ID: {1}, ONESOURCE Determination Request XML: {2}	{1}: Unique request ID that includes the cart id and the timestamp {2}: Request XML	The message is only activated if thomsonreuters.idt.detailedlogging.request=true and DEBUG log level is activated for class TridtSoapLoggingService

4	DEBUG	TridtSoapLoggingService	Log ID: {1}, ONESOURCE Determination Response XML: {2}	{1}: Unique request ID that includes the cart id and the timestamp {2}: Response XML	The message is only activated if thomsonreuters.idt.detailedlogging.response=true and DEBUG log level is activated for class TridtSoapLoggingService
5	INFO	DefaultTridtTaxCalculationService	Request stats for ID: {1} , Request Preparation Duration: #{2}#, Response Waiting Duration: ##{3}##, Response Processing Duration: ###{4}###	{1}: Unique request ID that includes the cart id and the timestamp {2}: Time passed while preparing the request {3}: Time passed in Determination System processing {4}: Time passed while processing the response	

In SAP Public Cloud Deployments these logs can be displayed in Kibana user interface. As a part of the integration package, three custom dashboards with several metrics and saved searches are provided for monitoring OneSource Determination integration. In your trtaxintegration extension, you will find a Kibana Saved Objects import file (Tridt_Kibana_Customizations.json) located in resources\kibana folder. You can import this file in Kibana/ Management / Saved Objects and start using your dashboards and saved searches in your SAP Commerce Public Cloud environment.

In order to activate DEBUG level logging for the XML messaging, include the following system properties to your local.properties file and restart the system:

```
log4j2.logger.tridtXMLLogging.name =
com.thomsonreuters.idt.integrations.sapcommerce.soap.client.impl.TridtSoapLog
gingService
log4j2.logger.tridtXMLLogging.level = DEBUG
log4j2.logger.tridtXMLLogging.appenderRef.stdout.ref = STDOUT
log4j2.logger.tridtXMLLogging.additivity = false
```

Performance Dashboard:

The dashboard includes several visualizations that show critical metrics about the overall status and performance of the integration in a specified time frame.

The time frame can be set in the upper right section. The dashboard includes the following information:

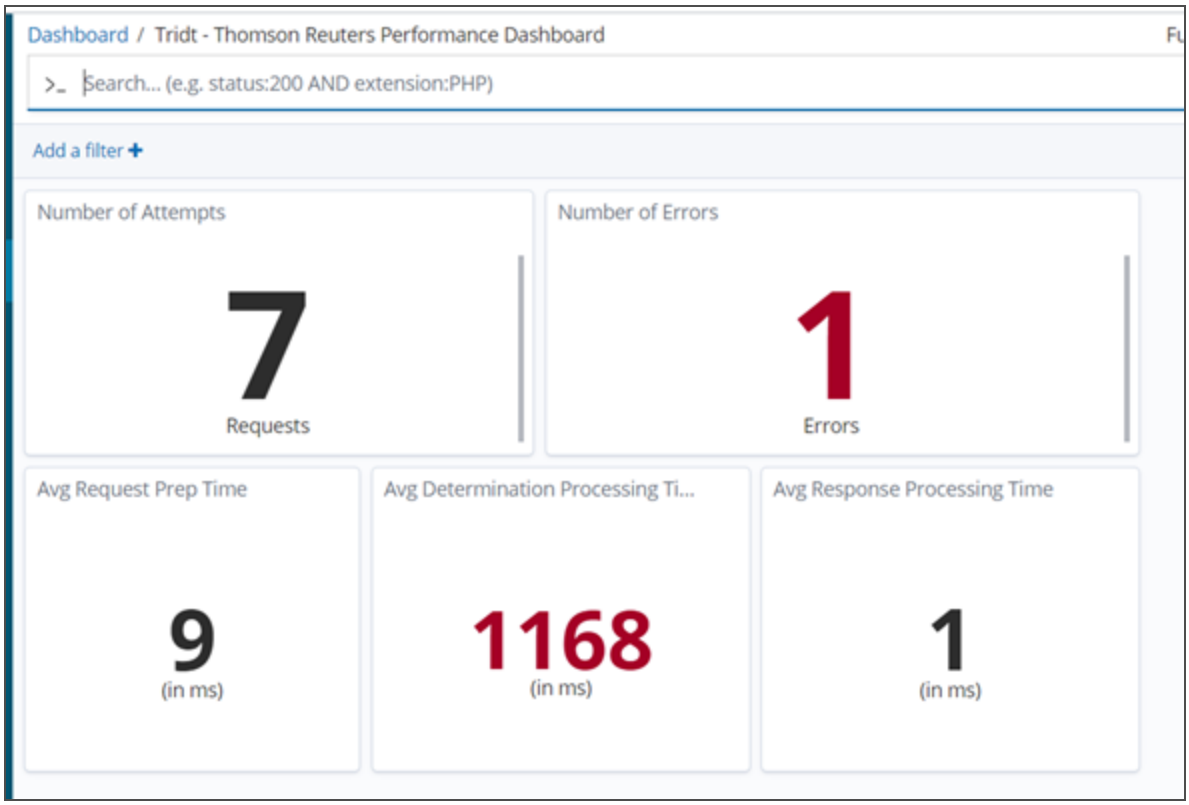
- The number of tax call attempts

- The total number of errors

- Average Request Preparation Time (SAP Commerce)

- Average Determination Processing Time (Connection + Determination Engine)

- Average Response Processing Time (SAP Commerce)



Errors Dashboard:

The dashboard includes several visualizations that focus on tax integration-related errors and their breakdowns in a specified time frame.

The time frame can be set in the upper right section. The dashboard includes the following information:

- The number of tax call attempts

- The total number of errors

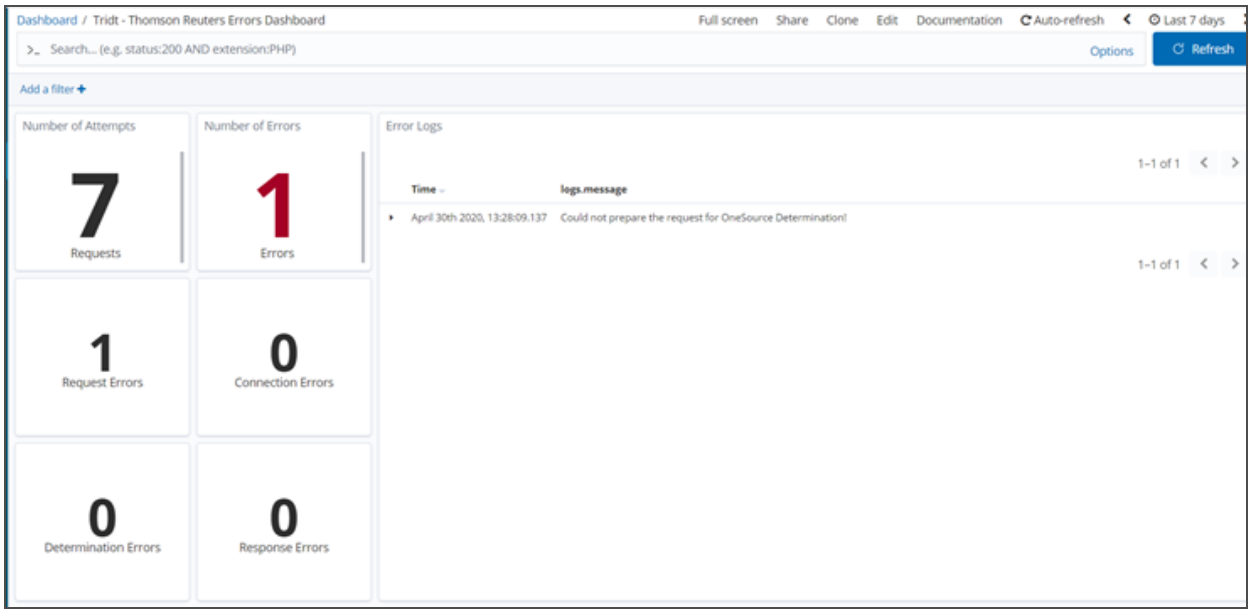
- The total number of request preparation related errors.

- The total number of connection errors.

- The total number of errors returned by ONESOURCE Determination.

- The total number of response processing related errors.

- List of all errors

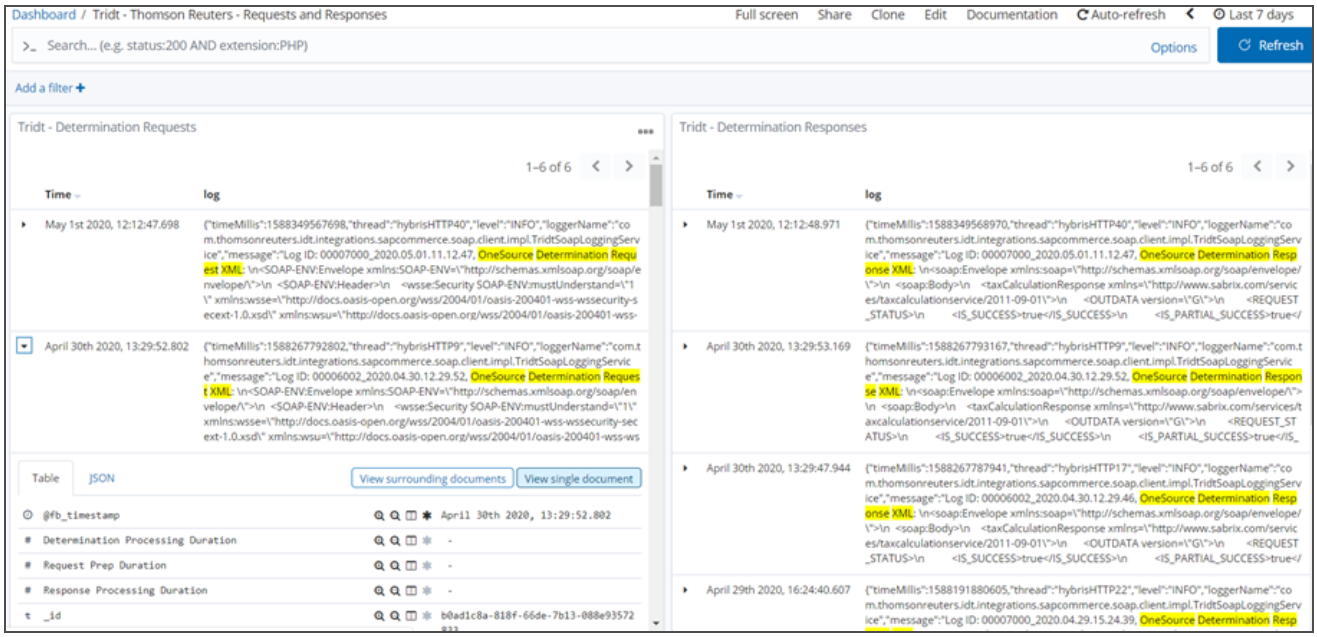


Requests and Responses Dashboard:

The dashboard includes two saved searches for request and response XMLs in a specified time frame. Please note that detailed logging should be activated for having XML messages to be written to the system log.

The time frame can be set in the upper right section.

Details of a single message can be displayed by expanding the details and clicking on the “View Single Document” button.



Tip: You can copy a log ID (Ex: 00007000_2020.04.29.15.24.39), go to the Discover page of Kibana and execute a search to display all tax integration relevant logs for a specific call. On the same page, you can also use wildcards to display all logs from multiple tax calls for a single cart (Ex: 00007000_*).

TROUBLESHOOTING SCENARIOS

Classification of Higher-Level Exceptions:

A typical tax process is composed of 4 different sub-processes and they run one after the other.

Request Preparation → Establishing a connection to Determination → Determination Engine's internal process → Response Processing

Search for the following keywords in the logs in order to understand which part of the process is causing the error.

Once you determine the problematic area, you should look for other exceptions/errors to determine the root cause.

KEYWORD TO SEARCH	SUBPROCESS THAT IS THROWING THE EXCEPTION	RESPONSIBLE SYSTEM	DETAILS
TridtRequestException	Request Preparation	SAP Commerce	Request Preparation Exceptions
TridtConnectionException	Establishing a connection to Determination	SAP Commerce	Connection Exceptions
TridtDeterminationException	Determination Engine's internal process	Determination	Determination Errors
TridtResponseException	Response Processing	SAP Commerce	Response Processing Exceptions
DefaultExternalTaxesService.saveOrder			
TaxValue.parseTaxValue			

Request Preparation Exceptions

You'll find the following exception in your logs: TridtRequestException – "Could not prepare the request for ONESOURCE Determination! Order/Cart/Return Request {1} cannot be converted to a request" where {1} is the document number.

This is a generic exception which indicates that the real problem is in request preparation. You need to investigate logs for further details. Here are some possible errors and their reasons:

EXCEPTION	TEXT	REASON
ConversionException	No External Company Id found for Thomson Reuters Tax integration	thomsonreuters.idt.externalcompanyid system property is missing
IllegalArgumentException	The default Point of Service assigned to the Base Store cannot be null.	BaseStore.defaultDeliveryOrigin field is empty.
IllegalArgumentException	No Ship-from address is assigned to the base store	BaseStore.defaultDeliveryOrigin.address field is empty.

NoSuchMethodException	Dynamic address field fetch failed!	Check thomsonreuters.idt.address.* properties. The value does not correspond to a field in AddressModel
IllegalArgumentException	Field XXX is not a String field!	Check thomsonreuters.idt.address.* properties. The value should be a String typed field in AddressModel

Response Processing Exceptions

You'll find one the following exceptions in your logs:

- **TridtResponseException** – "ONESOURCE Tax Determination response could not be processed! Response processing failed for {1}" where {1} is the document number. This exception indicates that response XML could not be converted into SAP's internal TaxValue structure
- Any exception in **de.hybris.platform.commerceservices.externaltax.impl.DefaultExternalTaxesService.saveOrder** indicates that converted TaxValue's could not be saved into the database.
- Any exception in **de.hybris.platform.util.TaxValue.parseTaxValue** indicates that converted TaxValue's could not be saved into the database.

Those are generic exceptions. You need to investigate logs for further details. Here are some possible errors and their reasons:

EXCEPTION	TEXT	REASON
ConversionException	Tax response is corrupted, conversion not possible	The response XML received, does not have OutdataInvoiceType inside.
NullPointerException	Cannot invoke "java.util.List.isEmpty ()" because "lines" is null	The response XML received, does not have OutdataLineType inside.
NullPointerException	Cannot invoke "java.util.List.stream ()" because "taxList" is null	There is no tax value info for a line in the response XML.

NullPointerException	element cannot be mapped to a null key	One of the tax values in the response message doesn't have the necessary key for summarization. Check thomsonreuters.idt.response.summarization system property and check why that field is not coming up from Determination.
DataIntegrityViolationException	ModelSavingException query; SQL []; String or binary data would be truncated.; nested exception is com.microsoft.sqlserver .jdbc.SQLServerException: String or binary data would be truncated.	Database table field length for CartEntries.taxValueInternal field is 255 characters long by design. This field is too short to keep all tax values for certain cases. SAP suggests changing the field length by running an "Alter Table" SQL query. In SAP Public Cloud deployments, customers can not perform this operation. You should open an incident to SAP Support, with component CEC-HCS-CCAZ-DBO, and ask SAP to change the length of field p_taxvaluesinternal in table cartentries to 1000 characters.
NumberFormatException	error parsing tax value string '<TV<XX-XXX #[ATT]sales [ATT]#4.07#true#4.07#USD>VT>' : java.lang.NumberFormatException: For input string: "[ATT]sales[ATT]"	One of the "tax code" for an item possible includes a special character that s misinterpreted by the SAP Commerce internal logic. Check the response XML for details.

Connection Exceptions

You'll find the following exception in your logs: **TridtConnectionException** – Could not call ONESOURCE Determination System!

This is a generic exception which indicates that the real problem occurred while trying to connect to Determination. You need to investigate logs for further details. Here are some possible errors and their reasons:

This kind of errors mostly occur when the SOAP Request is malformed or contains invalid authentication credentials. The SOAP Response would contain a <soap: Fault> element with two (2) child elements <faultcode> and <faultstring> as shown below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
        <soap:Fault>
        <faultcode
xmlns:ns1="http://ws.apache.org/wss4j">ns1:SecurityError</faultcode>
        <faultstring>A security error was encountered when verifying the
message</faultstring>
        </soap:Fault>
        </soap:Body>
        </soap:Envelope>
```

Here are some examples:

EXCEPTION	TEXT	REASON
SocketException	Connection timed out (Read failed)	Determination failed to send any response. This indicates an internal problem in Determination System.
WebServiceTransportException	404	Determination system could not be found. Check thomsonreuters.idt.url
SoapFaultClientException	A security error was encountered when verifying the message	Username and password can be wrong. Check thomsonreuters.idt.username and thomsonreuters.idt.password system properties
WebServiceIOException	I/O error: Permission denied: connect: {domain}; nested exception is java.net.SocketException: Permission denied: connect: {domain} where {domain} is the URL domain of Determination Service	Network communication error during integration, check network settings are correct

WebServiceIOException	I/O error: Connection reset; nested exception is javax.net.ssl.SSLException: Connection reset	Network communication error during integration, check network settings are correct

Determination Errors

You'll find one of the following exceptions in your logs:

TridtDeterminationException – "ONESOURCE Tax Determination returned an error! Determination error for id {1} : {2} " where {1} is the cart number and {2} is the error text returned by the Determination.

This is a generic exception which indicates that the real problem is in request preparation. You need to investigate logs for further details.

Determination Request Errors indicates that the Request was unsuccessful and is indicated on the <IS_SUCCESS> child element of the <REQUEST_STATUS> element found in the <OUTDATA> structure. The <IS_SUCCESS> contains a Boolean value (either true or false) indicating the status of the request. When the <IS_SUCCESS> value is false, the Integration should indicate to the calling application the request was unsuccessful. Below is a sample Determination Response where the Request is not successful:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
        <taxCalculationResponse
xmlns="http://www.sabrix.com/services/taxcalculationservice/2011-09-01">
        <OUTDATA version="G">
        <REQUEST_STATUS>
        <IS_SUCCESS>false</IS_SUCCESS>
        <IS_PARTIAL_SUCCESS>false</IS_PARTIAL_SUCCESS>
        <ERROR>
        <CODE>COMPANY_NOT_FOUND</CODE>
        <DESCRIPTION>Unknown Company.</DESCRIPTION>
        <ERROR_LOCATION_PATH>/OUTDATA/INVOICE[INVOICE_NUMBER='155']</ERROR_
LOCATION_PATH>
        </ERROR>
        </REQUEST_STATUS
```

The values of <CODE>, <DESCRIPTION> elements within the <ERROR> structure provides further details about the root cause of the error.

For the list of possible Determination Errors, please refer to the following page: List of Determination errors

Other Problems:

Sometimes there are no exceptions visible in logs but the tax values are supposed to have a different value. For those cases most of the time you need to focus on Request Preparation and you need to make sure that all relevant tags are included in the message and the values in the request XML are correct. Here are some examples:

AREA	SYMPTOM	SOLUTION
Address	Geocode tag is missing in the request XML	thomsonreuters.idt.address.postalcodeIncludesGeoCode should be set to false and thomsonreuters.idt.address.geocode should point to a String typed field in AddressModel
Address	Format of postal code is wrong	If thomsonreuters.idt.address.postalcodeIncludesGeoCode is true, postalCode field in AddressModel is expected to be in XXXXX-YYYY format where X's are for postal code and Y's are for Geocode
Address	County, city or district fields are missing in request XML	Check properties starting with thomsonreuters.idt.address.county (or city or district). The value should point to a String typed field in AddressModel.
Address	No bill-to address in the request for a B2B customer	The source of Bill-To address depends on how cart is paid. If it is a credit card payment, the bill-to address is searched in the PaymentInfo associated with the cart. If it can't be found, bill-to address is assumed to be the same with the ship-to address and no separate address is added to the tax request. If it is an account payment, the B2BUnit associated with the B2BCustomer is found and B2BUnit.billingAddress field is used to populate address fields. Please check there is no problem in those fields.
Header	Customer number is wrong for a B2B customer	The B2BUnit associated with the B2BCustomer is found and B2BUnit.billingAddress field is used for customer name and id. Please check there is no problem in those fields.

Response	Wrong summarization level or no summarization carried out at all.	Check thomsonreuters.idt.response.summarization property. Value may be wrong.

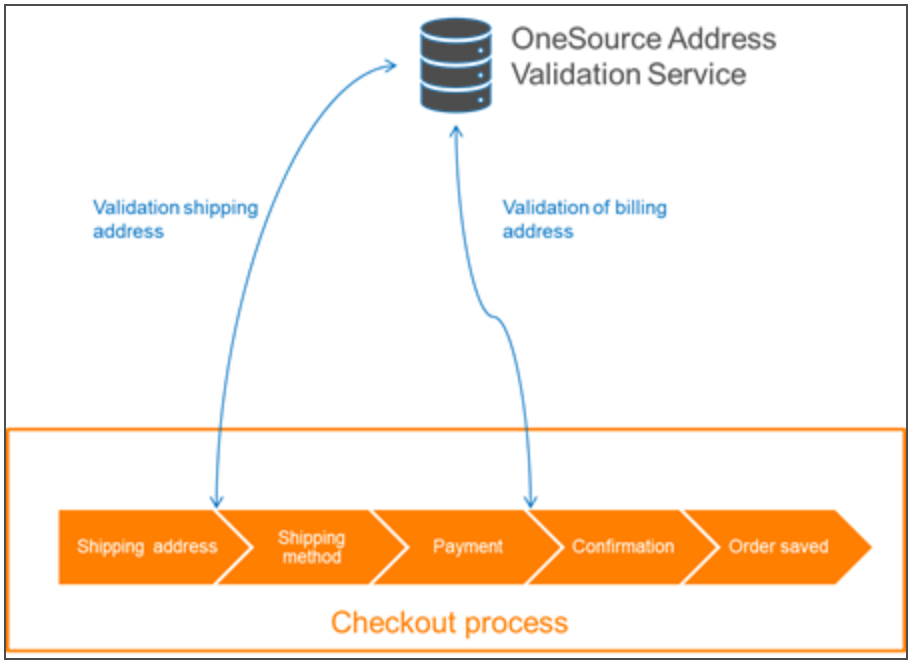
ADDRESS VERIFICATION SERVICE

For external address validation call to ONESOURCE AVS to be triggered in the system, the following prerequisites should be met:

- AVS Calls should be activated in system configuration (thomsonreuters.avs.integration.active=true)
- External Company ID should be defined in system configuration
- The user enters a US address in the storefront and hit Next

AVS Business Process Description

Checkout



During a checkout process both in B2C and B2B storefronts, two address validation requests can be sent to the Address Validation Service in order to validate the address details provided by the user. In this process, the user fills the address form to create a new address in the system and clicks “Next” to proceed to the next step. At this stage, AVS service is called with the form fields. The service can return three different results:

- Address confirmed: In this case, the address is saved on the cart (and optionally on the address book of the user) and the next checkout step is displayed afterward.
- Address corrected: AVS returns a similar address with some fields corrected. The new address is displayed as a popup to the user. The user can then accepts the new address or continue with the old version.
- Address rejected: When the address can not be identified by the service, an error is returned by the system together with a reason. The user stays on the address step to make the necessary corrections and the error and the reason are displayed as a notification.

In the checkout step, if the user selects an address from the address book (previously used and stored addresses) the integration will not be triggered.

My Account / Address Book

In the accelerators of SAP Commerce, users can edit their stored addresses in Address Book page under My Account section. When adding a new address or editing an existing one, the same AVS call mentioned above is triggered and the results are processed in a similar way.

AVS Configuration Options

It is possible to change the default integration behavior by providing certain system properties in local.properties file for local installations and manifest.json file SAP Public Cloud Systems. Each option is described below in detail:

KEY	DEFAULT VALUE	DESCRIPTION
thomsonreuters.avs.integration.active	true	If true, integration will be active
thomsonreuters.avs.addressvalidationservice.url	Empty	The value will be provided by Thomson Reuters.
thomsonreuters.avs.externalcompanyid	Empty	The value will be provided by Thomson Reuters

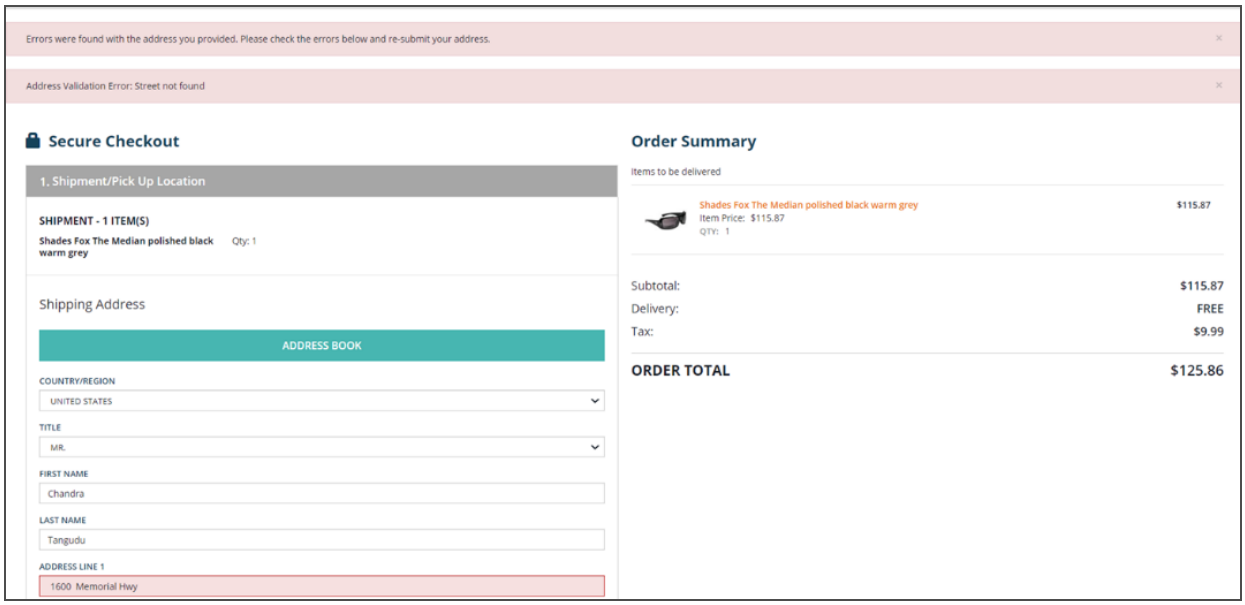
thomsonreuters.avs.password	Empty	The value will be provided by Thomson Reuters
thomsonreuters.avs.username	Empty	The value will be provided by Thomson Reuters
thomsonreuters.avs.detailedlogging.request	false	If true, the XML message for the request will be logged
thomsonreuters.avs.detailedlogging.response	false	If true, the XML message for the response will be logged
thomsonreuters.avs.integration.connection.errorallowed	false	If true, in case of connection or system errors, users will be allowed to proceed to the next checkout step
thomsonreuters.avs.integration.connection.errormessage	String	Address Verification Service is down.

AVS UI Messages

In case of an error returned by the service, the error text is passed to the UI to enable the user to make the necessary corrections.

Here are some possible error messages that can be returned by the service:

- Street number or box number out of range
- Please enter a valid address number.
- Street not found
- Please input a street address.
- Address not found
- Multiple addresses match.



AVS Logging and Monitoring

In the lifetime of a single address validation call, several log messages are generated by the system and those can be displayed in the standard application log in SAP Commerce. The table below summarizes the logs that should be observed in the logs for an error-free address validation call process.

STEP NO	LOG LEVEL	CLASS	TEXT	PARAMETERS	NOTES
1	INFO	TravsAddressVerification Service	Calling Addres Verification Service		
2	INFO	ClientFactory	Configuring system for company {1}	{1}: External Company ID	

3	INFO	TravsSoapLoggingService	Log ID: {1}, Address Verification Service Request XML: {2}	{1}: Unique request ID that includes the cart id and the timestamp {2}: Request XML	The message is only activated if thomsonreuters.avs.detailedlogging.request =true and DEBUG log level is activated for class TravsSoapLoggingService
4	INFO	TravsSoapLoggingService	Log ID: {1}, Address Verification Service Response XML: {2}	{1}: Unique request ID that includes the cart id and the timestamp {2}: Response XML	The message is only activated if thomsonreuters.avs.detailedlogging.response =true and DEBUG log level is activated for class TravsSoapLoggingService
5	INFO	TravsAddressVerification Service	Request stats for AddressLine1: {1} , Response Waiting Duration: ##{2}##	{1}: ID for the AVS call {2}: Time passed while waiting for the response from the external service	

In SAP Public Cloud Deployments these logs can be displayed in Kibana user interface. As a part of the integration package, three custom dashboards with several metrics and saved searches are provided for monitoring ONESOURCE Determination integration.

If an exception occurs during the service call, the root cause will be logged together with this error message: "**Address Verification Service call failed and not completed**".

For troubleshooting in Kibana, you can use this text to find the error logs and investigate the accompanying exception.

APPENDIX 1: DATA MAPPING

The table below describes the logic behind how SAP Commerce internal data structures are used to generate the Determination Request XML and how Determination Response XML is mapped back to SAP Commerce internal structures.

ONESOURCE DETERMINATION		SAP COMMERCE		
Level	Data Element	Type	Population Logic	Notes
Request				
Header - UsernameToken	Username	System Property	thomsonreuters.idt.username	
Header - UsernameToken	Password	System Property	thomsonreuters.idt.password	Will be masked in logs
INDATA	version (attribute of INDATA)	Fixed value	Default as 'G'	
INVOICE	EXTERNAL_COMPANY_ID	Variable value or System Property	BaseStore.tridtExternalCompanyId or thomsonreuters.idt.externalcompanyid	
INVOICE	HOST_SYSTEM	System Property	thomsonreuters.idt.hostsystem	

INVOICE	CALLING_SYSTEM_NUMBER	System Property	thomsonreuters.idt.callingsystemnumber	
INVOICE	CALCULATION_DIRECTION	Fixed value	Default is 'F'	
INVOICE	COMPANY_ROLE	Fixed value	Default is 'S'	
INVOICE	CURRENCY_CODE	Variable value	AbstractOrder.currency.isocode	
INVOICE	INVOICE_DATE	Variable value	For carts: Get the current date For Orders: Order.date For returns: ReturnRequest.order.date	
INVOICE	INVOICE_NUMBER	Variable value	For carts: AbstractOrder.code+Timestamp (Timestamp is in yyyy.MM.dd.HH.mm.ss format) For orders: AbstractOrder.code For returns: ReturnRequest.code	

INVOICE	IS_AUDITED	Variable value	true/false	For carts always false, for orders only true if delivery status is SHIPPED, for returns always true.
INVOICE	IS_CREDIT	Fixed value	true	Only for return requests
INVOICE	ORIGINAL_DOCUMENT_ID	Variable value	Associated order for the return request	Only for return requests
INVOICE	ORIGINAL_INVOICE_DATE	Variable value	The date for the associated order	Only for return requests
INVOICE	TAX_DETERMINATION_DATE	Variable value	The date for the associated order	Only for return requests
INVOICE	TRANSACTION_TYPE	Fixed value	Default as 'GS'	
INVOICE	CUSTOMER_NUMBER	Variable value	In B2C: AbstractOrder.user.uid In B2B: B2BUnit is found based on AbstractOrder.user and B2BUnit.uid is sent	
INVOICE	CUSTOMER_NAME	Variable value	B2BUnit is found based on AbstractOrder.user and B2BUnit.name is sent	Only valid in B2B scenarios

INVOICE	DELIVERY_TERMS	AbstractOrder	deliveryMode.code	Only valid in B2B scenarios Will be used only if thomsonreuters.idt.request.b2b.senddeliverymodeasincoterms=true
INVOICE	SHIP_FROM	Variable value	BaseStore.defaultDeliveryOrigin.address	The AddressModel found will be the basis to fill the address fields described below
INVOICE	SHIP_TO	Variable value	AbstractOrder.deliveryAddress	The AddressModel found will be the basis to fill the address fields described below
INVOICE	BILL_TO	Strategy	In B2C scenarios: Check AbstractOrder.paymentInfo.billingAddress, if not null use this address. If null, no Bill-To element is sent In B2B scenarios: Check AbstractOrder.paymentInfo, if it is a Credit Card, then the same as B2C, otherwise find B2BUnit associated with the user and send B2BUnit.billingAddress	The AddressModel found will be the basis to fill the address fields described below
INVOICE	ORDER_ACCEPTANCE	Strategy	Check system property thomsonreuters.idt.request.address.sendorderacceptanceaddress If the property is true, BaseStore.tridtOrderAcceptanceAddress will be sent.	The AddressModel found will be the basis to fill the address fields described below

INVOICE	ORDER_ORIGIN	Strategy	If activated by thomsonreuters.idt.request.address.sendorderoriginaddress: In B2C scenarios, will be the same as SHIP_TO. In B2B scenarios, B2BUnit.contactAddress will be sent if found. If not, it will be the same as SHIP_TO.	The AddressModel found will be the basis to fill the address fields described below
INVOICE	REGISTRATIONS / SELLER_ROLE	Variable value	BaseStore.tridtTaxRegistrationNumbers	
INVOICE	REGISTRATIONS / BUYER_ROLE	Variable value	B2BUnit.tridtTaxRegistrationNumbers	
ADDRESS	COUNTRY	Variable value	Address.country.isoCode	
ADDRESS	STATE	Variable value	Address.region.isoCodeShort	
ADDRESS	STATE	Variable value	Address.region.isoCodeShort	Only for Canadian addresses
ADDRESS	COUNTY	Variable value	Address.(field)	field is determined based on thomsonreuters.idt.address.county
ADDRESS	CITY	Variable value	Address.(field)	field is determined based on thomsonreuters.idt.address.city

ADDRESS	DISTRICT	Variable value	Address.(field)	field is determined based on thomsonreuters.idt.address.district
ADDRESS	POSTCODE	Variable value	Address.postalcode	field value may be stripped based on thomsonreuters.idt.address.postalcodeIncludesGeoCode
ADDRESS	GEOCODE	Variable value	Address.postalcode or Address.(field)	Please check Configuration Options for the following system properties: thomsonreuters.idt.address.postalcodeIncludesGeoCode thomsonreuters.idt.address.geocode
LINE	ID	Variable value	AbstractOrderEntry.entryNumber	
LINE	LINE_NUMBER	Variable value	AbstractOrderEntry.entryNumber	
LINE	PRODUCT_CODE / COMMODITY_CODE	Strategy	-	Please check Configuration Options for the following system properties: thomsonreuters.idt.request.item.sendCommodityCode thomsonreuters.idt.request.item.taxcode.enhancedsearch.producthierarchy.enabled thomsonreuters.idt.request.item.taxcode.enhancedsearch.categoryhierarchy.enabled thomsonreuters.idt.request.item.taxcode.fallbackCode

LINE	PART_NUMBER	Variable value	AbstractOrderEntry.product.code	
LINE	DESCRIPTION	Variable value	AbstractOrderEntry.product.name	
LINE	GROSS_AMOUNT	Variable value	AbstractOrderEntry.TotalPrice + AbstractOrderEntry.DiscountValues	
LINE	DISCOUNT_AMOUNT	Variable value	Propertionally distributed AbstractOrder.DiscountValues + AbstractOrderEntry.DiscountValues	
LINE	AMOUNT	Variable value	AbstractOrderEntry.quantity	
LINE	UOM	Variable value	AbstractOrderEntry.unit	
LINE	SHIP_TO	Variable value	AbstractOrderEntry.deliveryPointOfService.address or AbstractOrderEntry.deliveryAddress	Only if it is a pickup item or if the item delivery address is not empty
LINE	SHIP_FROM	Variable value	AbstractOrderEntry.deliveryPointOfService.address	Only if it is a pickup item
LINE	DELIVERY_TERMS	Variable value	AbstractOrderEntry.deliveryMode.code	Only valid in B2B scenarios Will be used only if item delivery mode is not empty and thomsonreuters.idt.request.b2b.senddeliverymodeasincoterms=true

LINE (DELIVERY)	ID	Strategy	Last entry number + 1	
LINE (DELIVERY)	LINE_NUMBER	Strategy	Last entry number + 1	
LINE (DELIVERY)	PRODUCT_CODE / COMMODITY_CODE	System Property	thomsonreuters.idt.deliveryitem	Please check Configuration Options for the following system properties: thomsonreuters.idt.request.item.sendCommodityCode thomsonreuters.idt.request.shipping.productcode
LINE (DELIVERY)	GROSS_AMOUNT	Variable value	AbstractOrder.deliveryCost or a calculated value if thomsonreuters.idt.request.shipping.sendmultipledelivery costitems=true	
LINE (DELIVERY)	OVERRIDE_AMOUNT	Variable value	With Country, State, County, City, District, Postal code breakdown	Used only if the delivery cost is being refunded in return request
LINE (DELIVERY)	AMOUNT	Fixed value	Default as '1'	
LINE (DELIVERY)	RELATED_LINE_NUMBER	Variable value	Associated entry number (Only if thomsonreuters.idt.request.shipping.sendmultipledelivery costitems)	Not used for Returns

LINE (DELIVERY)	USER_ELEMENT/ATTRIBUTE 1	Fixed value	Default is 'X'	Flag to differentiate shipment lines from product lines, used in response processing
Response				
INVOICE/LINE/TAX	AUTHORITY_NAME / ERP_TAX_CODE / AUTHORITYTYPE / EFFECTIVEZONELEVEL	ExternalTaxDocument /target/lineItemTaxes/ TaxValue	code	ONESOURCE field is selected based on thomsonreuters.idt.response.summarization
INVOICE/LINE/TAX	TAX_AMOUNT/AUTHORITY_AMOUNT	ExternalTaxDocument /target/lineItemTaxes/ TaxValue	value	Value is calculated based on thomsonreuters.idt.response.summarization. If summarization is requested several TAX_AMOUNT/AUTHORITY_AMOUNTs are summed up based on the summarization level
INVOICE/LINE/TAX	-	ExternalTaxDocument /target/lineItemTaxes/ TaxValue	absolute	Always false. SAP Commerce does not accept rates as external tax
INVOICE/LINE/TAX	TAX_AMOUNT/AUTHORITY_AMOUNT	ExternalTaxDocument /target/lineItemTaxes/ TaxValue	appliedValue	Same as value

INVOICE	CURRENCY_CODE	ExternalTaxDocument/target/lineItemTaxes/TaxValue	currency	
INVOICE/LINE/TAX	AUTHORITY_NAME / ERP_TAX_CODE / AUTHORITYTYPE / EFFECTIVEZONELEVEL	ExternalTaxDocument/target/shippingCostTaxes/TaxValue	code	ONESOURCE field is selected based on thomsonreuters.idt.response.summarization, thomsonreuters.idt.response.shipping.prefix is added as a prefix to the determined value.
INVOICE/LINE/TAX	TAX_AMOUNT/AUTHORITY_AMOUNT	ExternalTaxDocument/target/shippingCostTaxes/TaxValue	value	Value is calculated based on thomsonreuters.idt.response.summarization. If summarization is requested several TAX_AMOUNT/AUTHORITY_AMOUNTs are summed up based on the summarization level
INVOICE/LINE/TAX	-	ExternalTaxDocument/target/shippingCostTaxes/TaxValue	absolute	Always false. SAP Commerce does not accept rates as external tax
INVOICE/LINE/TAX	TAX_AMOUNT/AUTHORITY_AMOUNT	ExternalTaxDocument/target/shippingCostTaxes/TaxValue	appliedValue	Same as value
INVOICE	CURRENCY_CODE	ExternalTaxDocument/target/shippingCostTaxes/TaxValue	currency	

APPENDIX 2: AVS DATA MAPPING

The table below describes the logic behind how SAP Commerce internal data structures are used to generate the Determination Request XML and how Determination Response XML is mapped back to SAP Commerce internal structures.

ONESOURCE DETERMINATION		SAP COMMERCE		
Level	Data Element	Type	Attribute	Notes
Request				
Header - UsernameToken	Username	System Property	thomsonreuters.avs.username	
Header - UsernameToken	Password	System Property	thomsonreuters.avs.password	Will be masked in logs
ValidateAddressRequest	ExternalCompanyId	System Property	thomsonreuters.avs.externalcompanyid	
Address	Address1	Variable		Address field form filled by the user
Address	Address2	Variable		Address field form filled by the user
Address	City	Variable		Address field form filled by the user
Address	Region	Variable		Address field form filled by the user

Address	Postal Code	Variable		Address field form filled by the user
Address	Country	Variable		Address field form filled by the user
Response				
RequestStatus	Status	SUCCESS or ERROR		Status of the call
AddressResponse	Status	CONFIRMED, CORRECTED or ERROR		Status of the validation
ResponseAddress	Address1	Variable		Corrected value by the service
ResponseAddress	Address2	Variable		Corrected value by the service
ResponseAddress	City	Variable		Corrected value by the service
ResponseAddress	Region	Variable		Corrected value by the service

ResponseAddress	Postal Code	Variable		Corrected value by the service
ResponseAddress	Country	Variable		Corrected value by the service