

Oil & Gas Enablement

Integration for SAP 6.4.4.0

As of release 6.4.4.0 Integration for SAP has included a few new features that will aid users that wish to take advantage of new functionality that is offered in Cloud Determination 2018.3. This release is not a full solution for Oil & Gas industry users but instead enables several of the new features offered in Determination. Oil & Gas users will need to do custom configuration between SAP and Determination to map fields, augment address types, address sources, Customer and Vendor master records, and Material Master records.

New fields that were added to Determination XSD

There is a list of new fields that Determination added to the XML structure that provides new Oil & Gas options for tax calculations, functions, etc. that are specific to the industry.

This list of new fields includes the following list.

Bill of Lading

1. Additional elements to the XML Input and Output schema
2. Ensure proper contract update occurs for Integrations:
 - o InputTransactionReferenceMap.xls
 - o OutputTransactionReferenceMap.xls
3. Add the following at both the Document and Line Level:
 - o Bill of Lading Number should be a STRING (numbers and/or letters) of at least 100 characters
 - o Bill of Lading date needs to include time, time can be optional. As per discussion, the field can have date, datetime or be null
 - o Bill of Lading tag set should only occur once at the document and once per line, this should be documented in the XSD, but the validation against that is currently not being issued on the service itself so multiple tags could exist, but only the last one will be utilized.
 - o <BILL_OF_LADING>
 - o <NUMBER>123456</NUMBER>
 - o <DATETIME>YYYY-MM-DD HH:MI:SS</DATETIME>
 - o </BILL_OF_LADING>
4. If top level tags exist, number and timestamp must also exist

5. Fields should be audited as part of the transaction at the specific level
Document/Line

Bill of Lading Number:

Separate value from the Invoice Num(could be a large number/string), This needs to be at the document and line level

Bill of Lading Date/time:

Used for tax calculation, could override the Invoice Date, document and line level. Time can be distinctive in some situations at the line level so this field must include time to at least seconds. These values should be used when doing specific tax calculations that required a BOL time (for instance Oregon movement)

Transportation Type

Use of current Mode of Transportation field:

Within Oil/Gas we have a need to identify the Mode of Transport in a transaction, as this is a defining factor in the tax calculation. The existing Mode of transport field is currently used outside of the US for VAT reporting on the Intrastat Report. This field is currently only utilized for that, within reporting and is keying off of a VAT tax type.

Assumptions:

Oil/Gas uses the term 'Mode of Transport' to represent motor fuel was moved/transported. This is the typical industry term for this. The ideal then is to utilize this field for this purpose.

As an oil/gas user I need the ability to specify a Mode of Transport and have the tax engine determine whether or not that Mode of Transport is bulk based on Transportation types, so that I can get the proper tax calculation within an oil/gas transaction.

As an oil/gas user I need the ability for the system to utilize the free form text field combined with the enumeration of the Transportation Types within the system (custom and standard content) values to determine the Bulk Designation of a transaction.

As an oil/gas user I need the bulk designation to be part of the output so that I know the value used in the calculation.

Adding logic to the mode of transportation field so that it triggers a tax change for oil and gas scenario only

Determination table addition:

Table name:

TB_TRANSPORTATION_TYPES

Columns:

TRANSPORTATION_TYPE_ID required, number, length=22, precision=10
TRANSPORTATION_TYPE_UUID required, string, length=36
NAME required, string, length=100
DESCRIPTION optional, string, length=300
BULK_FLAG required, string, length=1 ("Y" or "N")
MERCHANT_ID required, number, length=22, precision=10
MERCHANT_UUID required, string, length=36
MERCHANT_DATACONTEXT required, string, length=4 ("CONT" or "CONF")
START_DATE required, date, length=7
END_DATE optional, date, length=7
CREATION_DATE required, date, length=7
LAST_UPDATE_DATE optional, date, length=7
SYNCHRONIZATION_TIMESTAMP optional, timestamp(6), length=11
LAST_UPDATED_BY optional, number, length=22, precision=10
CREATED_BY required, number, length=22, precision=10

Bulk Designation

Same as Bulk Flag as noted in Determination table for transportation types.

BULK DESIGNATION should be an output value within the OUTPUT XML. Any field added to Audit but that is not part of input should be added to the output, for instance the BILLED_QUANTITY_TYPE as part of NET/GROSS quantity selection should be on the output. In this case, the BULK_DESIGNATION selected should be in the output

Exchange Indicator

Indicates whether or not a transaction within Oil/Gas is an Exchange vs a Sale/Movement. In the case of an Exchange there are reporting challenges if the indication of Exchange does not exist. In the case of an exchange, likely the ST=S and there is no movement and only a sale.

In this case, the Order Origin or Order Acceptance as well as the ST/S location should be utilized in the tax calculation.

Input/Output XML element, Audited, at DOCUMENT and LINE level following process defined: TEC-906

```
# Is_Exchange
#* Boolean -(Default to N)- [Removed default value as it was causing unwanted results in audit]
#* Not required
```

```
# Ensure value is audited in TB_INVOICE* tables as necessary
# Use process outlined in TEC-906
```

Blend Indicator

Input/Output XML element, Audited, at LINE level ONLY following process defined:
[TEC-906](#)

1. IS_BLENDED
 - o Boolean [Default to N](#) [Removing default to N as it will cause unwanted results in audit]-NVJ
 - o Not required
2. Ensure value is audited in TB_INVOICE* tables as necessary
3. Element should come directly after Commodity or Product code passed

Latitude and Longitude

Please consider address elements at the LINE and DOCUMENT level, as well as audited address values for all of the below requirements. Both Input and Output XML elements should be added to the SOAP interface for calculation. Audit database should capture all data on the Input XML.

Process documented as part of [TEC-906](#) should be followed for this story

After discussion the type of Number (31,10) should account for both Latitude and Longitude precision

1. Add Latitude with the most precise data type available in Oracle/Java combined to each address element. Latitude can be -90 and max +90 and any value within those ranges with a precision varying in degrees (smaller float values). 1cm precision of accuracy is achieved at 6 decimal beyond the '.', millimeter precision is achieved at 8 decimals beyond the dot.
 - o Not a required element for the address
 - o If Latitude is present Longitude must be present or we would not be able to utilize the coordinates
 - o Decimal Degree format is the only format we are supporting with this release
2. Add Longitude with the most precise data type available in Oracle/Java combined to each address element. Longitude can be any number between -180 and +180 within those ranges with a precision varying in degrees. for 1 cm precision 10 decimal places after the '.'.
 - o Not a required element for an address

- If Longitude is present latitude must be present or we would not be able to utilize the coordinates, the system should provide an error when this occurs of missing required information
- Decimal degree format is the only format we are supporting with this release

3. An address element at the Document Level should be the value used for all lines created unless the values are defined at the lines which should override the document level (see address country field as it currently works this way) **Added 3/21/2018 for clarification**

Note from PM after market research

After confirmation from Tax Research (Meara and Coco) regarding customer requests for Lat/Long conversions, Deloitte (Michael Welch), PS with some tentative Oil/Gas E&P (upstream) customers, the degree decimal is format of choice due to the need to easily programmatically account for coordinates. We are choosing to utilize the decimal degree (number only) format.

Net quantity and Billed Quantity type

Ensure the calculation considers Net quantity instead of GROSS/Default quantity value if that value is specified as the 'BILLED_QUANTITY_TYPE'

1. Add a new 'Type' xml element as a child under the current 'Quanity' element at the line level
 - It should not be required (not defined = gross)
 - If it exists it can only be: GROSS|NET
 - Ensure 'amount' field accepts decimal(31,10) values **Updated 3/20/18 to match other amount columns**
2. XML Output should reflect the indata passed through the input XML
3. **5 new additional audit fields should exists in the TB_INVOICE_LINES table:**
 1. NET_QTY_UOM_CODE (same as QTY_UOM_CODE)
 2. NET_QUANTITY (same definition as QUANTITY)
 3. GROSS_QTY_UOM_CODE (same as QTY_UOM_CODE)
 4. GROSS_QUANTITY (same definition as QUANTITY)
 5. BILLED_QUANTITY_TYPE (string to contain GROSS|NET)

Scenarios for NET vs GROSS quantities being added

1. If a 'NET' type is passed when no ('GROSS' or empty) type is passed:
 1. The NET quantity/uom should populate the current 'QTY_UOM_CODE' and 'QUANTITY' field as it is the only quantity type available AND the BILLED_QUANTITY_TYPE should be populated with NET

2. If more than one 'NET' type is passed when no ('GROSS' or empty) type is passed:
 1. The first NET quantity/uom that is passed should populate the current 'QTY_UOM_CODE' and 'QUANTITY' field as it is the only quantity type available AND the BILLED_QUANTITY_TYPE should be populated with NET AND the NET_* should be populated with the values
3. If multiple 'NET' types are passed and one ('GROSS' or empty) type is passed OR more than one 'GROSS' type is passed:
 1. The first GROSS amount/uom that is passed should populate the current 'QTY_UOM_CODE' and 'QUANTITY' field AND GROSS_QUANTITY/GROSS_QTY_UOM_CODE should be populated with the value AND the BILLED_QUANTITY_TYPE = GROSS AND NET_QTY_UOM_CODE/NET_QUANTITY should be populated with the first 'NET' value.
 4. If a default flag is passed, whichever type/qty/value is associated with the default flag should be used in the current 'QTY_UOM_CODE' and 'QUANTITY' field AND would populate that specific 'GROSS' or 'NET' *_QTY_UOM_CODE and *_QUANTITY field. If both GROSS and NET are passed, then both *_QTY_UOM_CODE and *_QUANTITY would be populated.
 5. Quantities at the line level should allow multiple quantities of different type
 6. XML Output should have additional elements for the Net/Gross/Billed_quantity_type values

Description:

Within oil/gas there are 3 defined 'types' of quantities: Gross/Net/Billed

In any given transaction, the user could pass both a Gross and/or Net quantity. Net quantity is typically calculated in ERP based on what the quantity would be within a constant 60deg temperature environment. Then either the NET or GROSS would be passed in as the 'Billed' quantity.

The 'BILLED_QUANTITY_TYPE' can be either NET or GROSS depending upon the customer requirements.

We currently have a 'Quantities' element:

```

<QUANTITIES>
  <QUANTITY>
    <AMOUNT>1</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
<QUANTITIES>

```

For oil/gas we need a new 'type' attribute to quantity that would designate something other then the gross type (default/current behavior)
 Currently no type distinction, typically determined based on transaction to either

be Gross or Net, not typically specified, but definitely reported on for oil/gas depending upon the type specified.

New addition would look like:

```
<QUANTITIES>
  <QUANTITY>
    <DEFAULT>true</DEFAULT> (will audit to BILLED_QUANTITY_TYPE)
    based on defined type
    <TYPE>NET</TYPE>
    <AMOUNT>1</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
  <QUANTITY>
    <TYPE>GROSS</TYPE> OR if type is missing Gross is default
    <AMOUNT>2</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
<QUANTITIES>
```

SCENARIO 1: Additional NET type added to quantities [GROSS type used for calculation]

```
<QUANTITIES>
  <QUANTITY>
    <TYPE>NET</TYPE>
    <AMOUNT>1</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
  <QUANTITY>
    <AMOUNT>2</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
<QUANTITIES>
```

AUDITED DATA:

NET_QTY_UOM_CODE = EACH
NET_QUANTITY = 1
GROSS_QTY_UOM_CODE=EACH (populated based on the selected value or non-typed value being specified)
GROSS_QUANTITY= 2 (populated based on the selected value or non-typed value being specified)
BILLED_QUANTITY_TYPE = GROSS (use the gross type by default for calculation)
QUANTITY = 2
QTY_UOM_CODE = EACH

SCENARIO2: Typical current use case [GROSS type used for calculation]

```
<QUANTITIES>
  <QUANTITY>
```

```
<AMOUNT>2</AMOUNT>
<UOM>each</UOM>
</QUANTITY>
<QUANTITIES>
```

AUDITED DATA:

NET_QTY_UOM_CODE = null
NET_QUANTITY = null
GROSS_QTY_UOM_CODE=EACH (populated based on the selected value or non-typed value being specified)
GROSS_QUANTITY= 2 (populated based on the selected value or non-typed value being specified)
BILLED_QUANTITY_TYPE = GROSS
QUANTITY = 2
QTY_UOM_CODE = EACH

Scenario: mutliple gross [GROSS type used for calculation]

```
<QUANTITIES>
  <QUANTITY>
    <TYPE>GROSS</TYPE>
    <AMOUNT>1</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
  <QUANTITY>
    <AMOUNT>2</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
<BILLED_QUANTITY_TYPE>NET</BILLED_QUANTITY_TYPE>
<QUANTITIES>
```

AUDITED DATA:

NET_QTY_UOM_CODE = null
NET_QUANTITY = null
GROSS_QTY_UOM_CODE=EACH
GROSS_QUANTITY= 1
BILLED_QUANTITY_TYPE = GROSS
QUANTITY = 1 (current logic selects the 1st value specified as the value to use, populated based on GROSS type being passed and being selected)
QTY_UOM_CODE = EACH

Scenario: Default flag used with multiple types specified [NET type used for calculation]

```
<QUANTITIES>
  <QUANTITY>
    <TYPE>GROSS</TYPE>
    <AMOUNT>1</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
```

```
<QUANTITY>
  <TYPE>NET</TYPE>
  <DEFAULT>true</DEFAULT>
  <AMOUNT>2</AMOUNT>
  <UOM>each</UOM>
</QUANTITY>
<BILLED_QUANTITY_TYPE>NET</BILLED_QUANTITY_TYPE>
<QUANTITIES>
```

AUDITED DATA:

NET_QTY_UOM_CODE = EACH
NET_QUANTITY = 2
GROSS_QTY_UOM_CODE=EACH (is still audited even though it is not used in calculation)
GROSS_QUANTITY= 1
BILLED_QUANTITY_TYPE = NET
QUANTITY = 2 (populated based on the 'default' flag existing which will use the NET type even though gross type is first)
QTY_UOM_CODE = EACH

Address Types

The new Address Types brought by Oil and Gas are:

Origin
Ship From
Destination
Ship To
Sold To
Purchaser
Carrier
Consigner
Stock Holder
Exchange Stock
Diversion Destination

To support TEC 799, DetC backend needs to support new scenarios such that both buyer and seller have licenses. Whether licenses exist and/or their types will determine if tax can be exempted.

The new implementation can be referenced to "Registration" with the note that Registrations are associated to Company (MerchantID). This relationship serves VAT

Registration fine as VAT exemptions are supposed to take place within a group. This may not be suitable for Licenses.

One of the most significant components involved in any tax determination transaction is the location of the parties involved in the transaction.

In the US, each transaction involves up to five location types:

Ship From: Where the goods came from prior to receipt by the customer (for example, a warehouse or store).

Order Origin: Where the order for the merchandise was placed (for example, a store or a trade show).

Order Acceptance: The location where the order was accepted (for example, a call center where an order was taken).

Ship To: The location where the customer is receiving the merchandise (for example, the customer's address).

Supply: For services transactions, the location at which the service is supplied.

Each of these locations may have an impact on the calculation of transaction taxes for a particular transaction.

In addition to the location types above, international transactions may also use the following location types:

Bill To: A separate billing address for the customer.

Middleman: The location of a middleman who handles the paperwork for the buyer and the seller in a transaction.

Buyer Primary: The buyer's primary, or home, address.

Seller Primary: The seller's home, or primary, address.

Facility ID, IRS Registered Facility, Bulk Storage Facility

1. New elements should exist at the LINE and DOCUMENT level, as well as audited (sbxaud) address values for all of the below requirements. See **TB_INVOICE_ADDRESSES**, **TB_INVOICE_LINE_ADDRESSES**
2. Both Input and Output XML elements should be added to the SOAP interface for calculation.
3. Audit database should capture all data on the Input XML.
4. Add 'Facility ID' field to each address element (similar to Branch ID)
 - o Can be a string of numbers or characters, 100 characters field length
 - o Not a required element for an address
5. Add 'IRS Registered Facility' value to each address element (similar to bonded)
 - o Boolean y/n
 - o Not a required element for an address
 - o Default to N
6. Add 'Bulk Storage Facility' field to each address element (similar to bonded)

- Boolean Y/N
- Not a required element for an address
- Default to N

7. An address element at the Document Level should be the value used for all lines created unless the values are defined at the lines which should override the document level (see address country field as it currently works this way)

License types

A good starting point would be LicenseCondition.java

This will be the base for JE once its logic is decided.

Licenses will need to be categorized as

1. Vendor vs Customer
2. Self vs Business Partner

A possible solution can be such that CustomerLicense (re-use) and VendorLicense(new) to track licenses for business entities that do not have MerchantId (aka business partners of TR customer); MerchantLicense to track licenses for TR customer. Depends on how license will be used, MerchantLicense may need to be categorized as Seller/Buyer.

In a tax calculation, when the company role is seller, system will look for seller(vendor) license in MerchantLicense and buyer(customer) license in CustomerLicense.

Or, in another tax calculation, when the company role is buyer, system will look for buyer(customer) license in MerchantLicense and seller(vendor) license in VendorLicense.

Middleman is not in scope.

See jira TEC-799 for attached requirements document for License Type logic

TEC-968

1. Add support for Operating License Types in Determination for Content
 1. Operating License Types should support the following information
 - Name (50 char) (unique within date range) required
 - Associated Merchant (merchant_id) required as part of content
 - Description (1000 char)
 - Associated Zone (zone id) Linked to which location supports this license type
 - Start Date (part of uniqueness) Required
 - End Date
 2. Operating License Types should have the appropriate fields to facility associations or data uniqueness content functions
Lastupdated/lastcreated/UID/Data_context/indexes/etc.

3. Operating License Types should be recorded as other content is within Application Event History
4. Determination needs to support the content extract/triggers for this content
5. Code updated so that it supports SDI import/export
6. Determine cost to support custom OLT's and communicate to PM
7. No performance degradation on DetC

```
TEC-1026
<OPERATING_LICENSES>
  <ASSUME_BUYER_FULLY_LICENSED>true</ASSUME_BUYER_FULLY_LICENSED>
  <BUYER>
    <LICENSE_TYPE>LADDEC</LICENSE_TYPE>
    <LICENSE_NUMBER>12345</LICENSE_NUMBER>
  </BUYER>
  <BUYER> <!--10 max occurs of buyer)
    <LICENSE_TYPE>LAD</LICENSE_TYPE>
    <LICENSE_NUMBER>S12345</LICENSE_NUMBER>
  </BUYER>

  <ASSUME_SELLER_FULLY_LICENSED>true</ASSUME_SELLER_FULLY_LICENSED>
  <SELLER>
    <LICENSE_TYPE>LAPS</LICENSE_TYPE>
    <LICENSE_NUMBER>12E345</LICENSE_NUMBER>
  </SELLER>
  <SELLER> <!--10 max occurs of SELLER)
    <LICENSE_TYPE>LAGOV</LICENSE_TYPE>
    <LICENSE_NUMBER>125345</LICENSE_NUMBER>
  </SELLER>
</OPERATING_LICENSES>
```

FTA Code Support

Federation of Tax Administrator codes specifically created for Fuel Tax Administration are codes that are representative of existing products (mostly) within our UNSPSC tree (US product tree). These are different codes for the same value

1. User needs the system to accept a FTA code within the input xml elements available in the Enterprise SOAP service and calculate the proper taxability for this code, so that the users with ERP's that have FTA codes stored for their products don't have to do custom mappings to enable oil/gas transactions
 1. User needs the system to account for differences between the FTA code number scheme and the current US tree numbering scheme:
 - FTA code values currently can conflict with content in the US tree as there are FTA codes that are the same as UNSPSC codes that exist in the US tree.

- FTA Code values can have a variable percentage associated with the code itself, such as M0-M99 where M0,M1,M2,M3,etc. are all possible codes within a range from 0-99.
 - FTA Code
- 2. User needs the system to audit these codes in the audit database under a field similar to commodity code that exists at the document/invoice and line level, so that these FTA codes can be reported on at a later point in time.
- 3. User needs the system to display the audited value of the Input Element on the Output/Return from a calculation call
- 4. User needs the ability to map to the element from a Transeditor/rule qualifier/any other XML_ELEMENT mapping feature in the UI
- 5. User needs the ability to accept the same FTA code value from within the Model Scenario's UI for inclusion in a model scenario calculation

Validate that Exemption Certificates will not erroneously exempt a transaction from Excise tax

Without knowing the exact details of 'how' exemption certificates work in the engine, we need to ensure that the Excise taxes that occur within oil/gas will not be 'exempt' by a blanket exemption certificate that is intended for Sales/Use Tax.

Exemptions certificates do not exist for companies to be exempt from excise tax. Only Government bodies can be exempt, and even that is in certain cases.

Ensure that an exemption certificate can only apply to specific Tax Types or transactions and would not be considered as part of an oil/gas transaction with Excise or Motor Fuels Tax Types

10 New Date and Amount Attributes

1. At the Line and Document level in the INPUT and OUTPUT XML schema add the following 'Custom' attributes:
 - CUSTOM_DATE_ATTRIBUTE (1-5) (Date type) same date format as the invoice. invoice_date
 - CUSTOM_AMOUNT_ATTRIBUTE (1-5) (number type) number (31,5) see tb_invoices.total_tax_amount
2. These fields should be treated the same as another custom attribute within Input/Output/Data model/etc.

3. Ensure these fields are audited in the TB_INVOICE* table
4. Follow process prescribed
5. <USER_ELEMENT>
6. <NAME>ATTRIBUTE1</NAME>
7. <VALUE>MY CUSTOM STRING</VALUE>
8. </USER_ELEMENT>
9. <USER_ELEMENT>
10. <NAME>ATTRIBUTE2</NAME>
11. <VALUE>MY CUSTOM STRING</VALUE>
12. </USER_ELEMENT>
13. <USER_ELEMENT_AMOUNT>
14. <NAME>ATTRIBUTE1</NAME>
15. <VALUE>(ONLY NUMBERS ALLOWED HERE)</VALUE>
16. </USER_ELEMENT_AMOUNT>
17. <USER_ELEMENT_DATE>
18. <NAME>ATTRIBUTE1</NAME>
19. <VALUE>(ONLY DATES ALLOWED HERE)</VALUE>
20. </USER_ELEMENT_DATE>

As there are a few fields that are necessary for Oil/Gas that are of a specific type: Date or Number(amount), we need to account for these but not be specific about them for our customers.

There are named amounts and named dates that oil/gas transactions from a reporting perspective may require depending upon the form that Determination cannot explicitly pass through a calculation and maintain the type.

Amount examples in OG:

- Amount paid with tax
- Amount paid without Tax
- Tax collected/tax due/tax outstanding/tax paid/tax unpaid
- Quantity non-taxable
- Quantity Taxable

Dates examples in OG:

- Receipt Date (date product was received)
- Import Verification Assigned Date

XSD New Fields in XML structure

```
<INDATA version="G">
<SCENARIO_NAME>Brazil Cent Rounding</SCENARIO_NAME>
```

```

<INVOICE>
  <CALCULATION_DIRECTION>F</CALCULATION_DIRECTION>
  <COMPANY_ID>2392</COMPANY_ID>
  <COMPANY_NAME>DTest</COMPANY_NAME>
  <COMPANY_ROLE>B</COMPANY_ROLE>
  <CURRENCY_CODE>USD</CURRENCY_CODE>
  <DOCUMENT_TYPE></DOCUMENT_TYPE>
  <EXTERNAL_COMPANY_ID>DTest</EXTERNAL_COMPANY_ID>
  <INVOICE_DATE>2015-09-16</INVOICE_DATE>
  <IS_AUDITED>false</IS_AUDITED>
  <IS_BUSINESS_SUPPLY>true</IS_BUSINESS_SUPPLY>
  <IS_REPORTED>false</IS_REPORTED>
  <IS_REVERSED>false</IS_REVERSED>
<MODE_OF_TRANSPORT>MOT</MODE_OF_TRANSPORT><!-- Current field would be modified to
use the enumerations that exist within the Transportation Types table to key off
of for Oil/Gas in the US (TEC-866), also currently used for VAT intrastat
reporting-->
<POINT_OF_TITLE_TRANSFER>I</POINT_OF_TITLE_TRANSFER>
  <IS_EXCHANGE>true</IS_EXCHANGE><!-- Optional, max occurs 1 per INVOICE
entity, boolean (Tec-900)-->
  <REGISTRATIONS>
    <BUYER_ROLE>12312312312321</BUYER_ROLE>
    <BUYER_ROLE>1234567890123</BUYER_ROLE>
    <SELLER_ROLE>456456456456</SELLER_ROLE>
    <SELLER_ROLE>45645645645645</SELLER_ROLE>
  </REGISTRATIONS>
  <OPERATING_LICENSES><!--Optional, max occurs 1 per document/invoice,
wrapper for license types (TEC-1026)-->
    <BUYER_LICENSE_TYPE>LADDEC</BUYER_LICENSE_TYPE><!--Optional, max occurs
1 per operating license entity, varchar(50) represents the license type code
value (will be in content and custom) -->
    <SELLER_LICENSE_TYPE>LAPS</SELLER_LICENSE_TYPE><!--Optional, max occurs
1 per operating license entity, varchar(50) represents the license type code
value (will be in content and custom) -->
  </OPERATING_LICENSES>
  <BILL_OF_LADING><!--Optional, max occurs 1 per document/invoice, wrapper
for bol info (TEC-868)-->
    <NUMBER>123456</NUMBER><!--Optional, max occurs 1 per bill of lading,
varchar 100-->
    <DATETIME>YYYY-MM-DD HH:MI:SS</DATETIME><!--Optional, max occurs 1 per
bill of lading, date and time, time is optional-->
  </BILL_OF_LADING>
<SELLER_PRIMARY>
  <COUNTRY>BR</COUNTRY>
  <PROVINCE>SP</PROVINCE>
  <CITY>POTIRENDABA</CITY>
  <DISTRICT>POTIRENDABA</DISTRICT>
  <POSTCODE>15105000</POSTCODE>
  <GEOCODE></GEOCODE>
  <ALIAS></ALIAS><!-- Optional, max occurs 1 per address block,
varchar(100), (TEC-911)-->

```

```

<FACILITY_ID>abc123</FACILITY_ID><!-- Optional, max occurs 1 per address
block, varchar(100) TEC-910-->
    <IRS_REGISTERED_FACILITY>true</IRS_REGISTERED_FACILITY><!-- Optional,
max occurs 1 per address block, boolean TEC-910-->
        <BULK_STORAGE_FACILITY>true</BULK_STORAGE_FACILITY><!-- Optional, max
occurs 1 per address block, boolean TEC-910-->
            <LATITUDE>89.12345678</LATITUDE><!-- Optional, float at least 8 decimal
places TEC-898-->
            <LONGITUDE>130.1234567890</LONGITUDE><!-- Optional, float at least 10
decimal places (only 3 to the right) TEC-898-->
        </SELLER_PRIMARY>
<SHIP_FROM>
    <COUNTRY>BR</COUNTRY>
    <PROVINCE>SP</PROVINCE>
    <CITY>POTIRENDABA</CITY>
    <DISTRICT>POTIRENDABA</DISTRICT>
    <POSTCODE>15105000</POSTCODE>
        <GEOCODE>97035</GEOCODE>
    <ALIAS>ORIGIN</ALIAS>
        <FACILITY_ID>abc123</FACILITY_ID>
        <IRS_REGISTERED_FACILITY>true</IRS_REGISTERED_FACILITY>
        <BULK_STORAGE_FACILITY>true</BULK_STORAGE_FACILITY>
        <LATITUDE></LATITUDE>
        <LONGITUDE></LONGITUDE>
</SHIP_FROM>
<SHIP_TO>
    <COUNTRY>BR</COUNTRY>
    <PROVINCE>MG</PROVINCE>
    <CITY>CAPINOPOLIS</CITY>
    <DISTRICT>VILA BRASILIA</DISTRICT>
    <POSTCODE>38360000</POSTCODE>
        <FACILITY_ID>abc123</FACILITY_ID>
<IRS_REGISTERED_FACILITY>true</IRS_REGISTERED_FACILITY>
    <BULK_STORAGE_FACILITY>true</BULK_STORAGE_FACILITY>
</SHIP_TO>
<SUPPLY>
    <COUNTRY>BR</COUNTRY>
    <PROVINCE>MG</PROVINCE>
    <CITY>CAPINOPOLIS</CITY>
    <DISTRICT>VILA BRASILIA</DISTRICT>
    <POSTCODE>38360000</POSTCODE>
</SUPPLY>
<TRANSACTION_TYPE>GS</TRANSACTION_TYPE>
<USER_ELEMENT><!--document level element-->
    <DATE_NAME>ATTRIBUTE1</DATE_NAME><!-- optional only, max occurrence 5,
see current attribute name definition, TEC-899-->
    <DATE_VALUE>2018-01-01</DATE_VALUE><!-- optional only, max occurrence 5,
datatype field (see current invoice_date field for reference), TEC-899-->
</USER_ELEMENT>
<USER_ELEMENT><!--document level element-->
    <AMOUNT_NAME>ATTRIBUTE1</AMOUNT_NAME><!-- optional only, max occurrence
5, see current attribute name definition, TEC-899-->

```

```

<AMOUNT_VALUE>123456789.12456</AMOUNT_VALUE><!-- optional only, max
occurrence 5, see current gross amount field for definition, TEC-899-->
</USER_ELEMENT>
<MOVEMENT_TYPE></MOVEMENT_TYPE>

<MODE_OF_TRANSPORT></MODE_OF_TRANSPORT>
<LINE_ID="1">
  <COMMODITY_CODE>fta126</COMMODITY_CODE>
  <GROSS_AMOUNT>12557.47</GROSS_AMOUNT>
  <LINE_NUMBER>1</LINE_NUMBER>
    <MODE_OF_TRANSPORT>Pipeline</MODE_OF_TRANSPORT><!-- Current field would
be modified to use the enumerations that exist within the Transportation Types
table to key off of for Oil/Gas in the US (TEC-866), also currently used for VAT
intrastat reporting-->
    <PRODUCT_CODE>Gas87</PRODUCT_CODE>
    <IS_BLENDED>true</IS_BLENDED><!-- Optional, max occurs 1 per line entity,
boolean-->

<!-- existing quantities structure:
<QUANTITIES>
  <QUANTITY>
    <AMOUNT>340</AMOUNT>
    <UOM>each</UOM>
  </QUANTITY>
</QUANTITIES> -->
  <!-- Modification to quantities structure -->
  <QUANTITIES>
    <QUANTITY>
      <DEFAULT>true</DEFAULT> <!--used to define the default quantity being
passed, also now used to determine what the 'billed quantity type' should be for
Oil/Gas transaction -->
      <TYPE>NET</TYPE> <!-- Optional, max occurs 1 per quantity, New element
can be GROSS or NET static values, if does not exist, then default to GROSS,
(TEC-869) If NET is default then will audit the value of Type "NET" to
BILLED_QUANTITY_TYPE-->
      <AMOUNT>1</AMOUNT>
      <UOM>each</UOM>
    </QUANTITY>
    <QUANTITY>
      <TYPE>GROSS</TYPE> OR if type is missing Gross is default
      <AMOUNT>2</AMOUNT>
      <UOM>each</UOM>
    </QUANTITY>
<QUANTITIES>
  <OPERATING_LICENSES><!--Optional, max occurs 1 per line entity, wrapper for
license types (TEC-1026)-->
    <ASSUME_BUYER_LICENSED>true</ASSUME_BUYER_LICENSED><!--optional, max
occurs 1 per operating license entity, boolean-->
    <!-- 0 to 10 repetitions of BUYER :-->
    <BUYER> <!--optional, max occurs 10 per operating licenses entity-->
      <LICENSE_TYPE>LADDEC</LICENSE_TYPE>
      <LICENSE_NUMBER>12345</LICENSE_NUMBER>

```

```

</BUYER>
<ASSUME_SELLER_LICENSED>true</ASSUME_SELLER_LICENSED><!--optional, max
occurs 1 per operating license entity, boolean-->
<!--0 to 10 repetitions of SELLER :-->
<SELLER> <!--optional, max occurs 10 per operating licenses entity-->
    <LICENSE_TYPE>LAPS</LICENSE_TYPE>
    <LICENSE_NUMBER>12E345</LICENSE_NUMBER>
</SELLER>
</OPERATING_LICENSES>
    <BILL_OF_LADING><!--Optional, max occurs 1 per line, wrapper for
bol info (TEC-868)-->
        <NUMBER>123456</NUMBER><!--Optional, max occurs 1 per bill of lading,
varchar 100-->
        <DATETIME>YYYY-MM-DD HH:MI:SS</DATETIME><!--Optional, max occurs 1 per
bill of lading, date and time, time is optional-->
    </BILL_OF_LADING>
    <IS_EXCHANGE>false</IS_EXCHANGE><!-- Optional, max occurs 1 per line
entity, boolean-->
<USER_ELEMENT><!--line level element-->
    <DATE_NAME>ATTRIBUTE1</DATE_NAME><!-- optional only, max occurrence 5,
see current attribute name definition, TEC-899-->
    <DATE_VALUE>2018-01-01</DATE_VALUE><!-- optional only, max occurrence 5,
datatype field (see current invoice_date field for reference), TEC-899-->
</USER_ELEMENT>
<USER_ELEMENT><!--line level element-->
    <AMOUNT_NAME>ATTRIBUTE1</AMOUNT_NAME><!-- optional only, max occurrence
5, see current attribute name definition, TEC-899-->
    <AMOUNT_VALUE>123456789.12456</AMOUNT_VALUE><!-- optional only, max
occurrence 5, see current gross amount field for definition, TEC-899-->
</USER_ELEMENT>
</LINE>
</INVOICE>
</INDATA>

```

The above list of new fields for Oil & Gas were identified during the Determination project and may not be all inclusive as new fields can be added in another release or update. We recommend you refer to Determination user guides and documentation to confirm if any changes have been added or deprecated.

Some of the new fields that were added to Determination 2018.3 will not be utilized in the tax calculations until the next release that is scheduled for later this year. Others are used in calculations now at time of their release August 2018. None of the new fields are included in standard field mapping and users will have to identify the matching field in SAP and create customer user field and address mappings in their Integration configuration in order to capture and pass the data to and from Determination.

Address Mapping of new fields

Five of the new fields reside within the address data for ship from, ship to, seller primary, buyer primary, etc. address sources. Currently the logic for adding new fields to the address sources is a ABAP function hidden inside of hard coded class programs for each of the standard address sources. This function has been improved with the Integration 6.4.4.0 as we have added an additional address source mapping table and exposed the address journey within the prior Field Mapping Table to make it easier for Oil & Gas industry users to map the new fields rather than requiring them to repetitively open and modify many classes of code. Users will be able to use the new mapping tables instead. They may also need to add new custom address sources based on their industry needs. All of this is documented in the *ONESOURCE Tables Configuration Guide* under the section on Address Mapping. Added instructions are also documented in the *Installation and Programmer's Guide* shortly after the information on loading the new release transports.

New Proxy and WSDL Configuration

The new fields are only being added to the Cloud Determination product and there is no current plan to add them to the on-premise version of our Determination. Because the WSDL contains this XSD field structure, there is now the necessity to have two different WSDL files and Proxy configurations: one for the on-premise needs and the new one for Oil & Gas functionality needs. This means that there could be two proxy versions being used at the same time if you are an Oil & Gas customer that is converting to Determination 2018.3 (or above) and you have both versions of Determination that you are using simultaneously. Changes were needed with our integration to recognize this need and prevent field mapping errors within our tables.

A new Proxy Group Code was created as a field that is used on several of our configuration tables. You will see this new field at the end of the Proxy Configuration Table, the Field Mapping Table, and the new Address Source Mapping Table. See documentation within the *ONESOURCE Tables Configuration Guide* for special instructions on the Proxy config and field/address mapping table sections.