

Oracle Fusion Cloud Transportation and Global Trade Management

**External Programming Interface
Guide**

Release 24B



Oracle Fusion Cloud Transportation and Global Trade Management
External Programming Interface Guide

Release 24B

F91688-01

Copyright © 2009, 2024, Oracle and/or its affiliates.

Author: Anil Somepalli

Contents

Get Help	i
<hr/>	
2 External Programming Interface Overview	3
Overview	3
3 External Distance Engine (EDE)	5
External Distance Calculation	5
Web Service Explanation	6
External Location Validation	7
4 External Service Engine (ESE)	9
Web Service Explanation	9
External Service Time Calculation	10
Web Service Explanation	11
5 External Rating Engine (ERE)	13
External Rate Calculation	13
Web Service Calculation	15
6 External Web Service Engine	19
Overview	19
Generate Stubs	19
Service End Point Implementation	20
Compile and Package Service Artifacts	20
Deploy	20
7 Configuration	21
Configure External Web Service Engine in Oracle Transportation and Global Trade Management Cloud	21
Document	21

Web Service	21
External System	21
External Engine	22


8 Appendix: WSDLs **23**

Distance Engine	23
Rating Engine	27
Service Time Engine	31

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons  to access help in the application.

Get Support

You can get support at [My Oracle Support](#). For accessible support, visit [Oracle Accessibility Learning and Support](#).

Join Our Community

Use [Cloud Customer Connect](#) to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle Accessibility Program](#). (if videos) Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to otm-doc_us@oracle.com.

Thanks for helping us improve our user assistance!

2 External Programming Interface Overview

Overview

The intent of this document is to provide documentation for the Web Service Interfaces that are provided for customers or third parties to extend existing Oracle Transportation and Global Trade Management Cloud functionality. Oracle Transportation and Global Trade Management Cloud connects as a client to these web services implemented by the customers or third parties conforming to the WSDL provided.

The Oracle Transportation and Global Trade Management Cloud products expose interfaces to support integration with external application in the areas of distance, service time and cost calculation. This document will describe the following:

- External Distance Engine (EDE) Interface
- External Service Engine (ESE) Interface
- External Rating Engine (ERE) Interface
- Generate, Deploy and Use Web Service in Oracle Transportation and Global Trade Management Cloud
- WSDL Definitions (Web Services Description Language – WSDL)

3 External Distance Engine (EDE)

External Distance Calculation

You have the ability to interface Oracle Transportation and Global Trade Management Cloud to any third-party distance engine by connecting Oracle Transportation and Global Trade Management Cloud as a web service client to the custom web service that adheres to the WSDL interface provided by Oracle Transportation and Global Trade Management Cloud.

You will need to implement a SOAP based web service by following the steps provides in the Web Service Implementation section and using the Distance Engine WSDL provided in the appendix. Register the newly implemented web-service with Oracle Transportation and Global Trade Management Cloud by following the steps provided in Configuration section to be able to access the web service to return distance to Oracle Transportation and Global Trade Management Cloud.

Setup

The first step is to ensure if there is a valid External Distance Engine (EDE) defined in Oracle Transportation and Global Trade Management Cloud. The External Distance Engine has to be defined with a Connection Type as “Webservice (W)”. External Distance Engine (EDE) is the main link between Oracle Transportation and Global Trade Management Cloud and a custom EDE.

External Distance Engine

Field	Value	Comments
External Distance ID	USER_DEFINED	Identifies the EDE.
Connection Type	Webservice(W)	Defines the type of transport mechanism used.
Implementing Java Class	glog.business.rate.ratedistance.external.webservice.client.ExternalDistanceWebServiceClient (this is automatically populated and field is read only).	Specifies the fully qualified name of the java implementation class, which will link to the external distance engine web service as a client.
External System ID	User Defined	External system is required for specifying the web service endpoint and the web service operations to invoke.

In the External Distance Engine, you can specify the information that will be passed from Oracle Transportation and Global Trade Management Cloud into the invocation of the external distance web service via parameters and are mapped via the edeParams variable in the interface. Using the lower half of the Power Data page, you can define a set of parameters like source address type, destination address type and other parameters.

External Distance Engine Parameters

Parameter	Value	Country Code	Comments
User Defined	User Defined	User Defined * for all countries	These parameters will be available to the Web Services and can be used for program control and/or data.

Web Service Explanation

A third-party external distance engine can be defined using EDE with the connection type of “WebService (W)” and using a java class `glog.business.rate.ratedistance.external.webservice.client.ExternalDistanceWebServiceClient`. Oracle Transportation and Global Trade Management Cloud as web service client connects to the custom third-party distance engine web service that adheres to the WSDL interface.

lookupDistance Operation

Inputs

`lookupDistance` operation takes `lookupDistanceRequest` as input that contain below element as child elements

- `sourceAddress` contains all the location information for the source address.
- `destinationAddress` contains all the location information for the destination address.
- `extEngineAuxInputList` contain SCAC and Route Codes off the locations, contains the equipment dimensions, weight and volume of the equipment being used for transportation and contains hazardous material information being shipped if hazardous routing is enabled. The SCAC and Route Codes are usually only necessary for rail lookups. This element is available as read-only.
- `edeParams` is a mapping to all the input parameters that were entered in Oracle Transportation and Global Trade Management Cloud for the specific EDE.

Outputs

`extEngineDistance` is the result of the `lookupDistance` operation. `extEngineDistance` element contains two child elements named `uom` and `amount`. An unsuccessful call to `lookupDistance` will return a fault to indicate the exception scenario with the reason populated in the `ExtEngineException` element.

Note: All the values for XML elements involved in web service should be valid as per Oracle Transportation and Global Trade Management Cloud WSDL.

External Location Validation

You have the ability to interface Oracle Transportation and Global Trade Management Cloud to any third-party engine that supports location validation by connecting as a web service client to the custom web service that adheres to the WSDL interface provided by Oracle Transportation and Global Trade Management Cloud.

You will need to implement a SOAP based web service by following the steps provided in the Web Service Implementation section and using the Distance Engine WSDL provided in the appendix. Register the newly implemented web-service with Oracle Transportation and Global Trade Management Cloud by following the steps provided in Configuration section to be able to access the web service to return distance to Oracle Transportation and Global Trade Management Cloud.

4 External Service Engine (ESE)

Web Service Explanation

A third-party external distance engine can be defined using EDE with the connection type of “WebService (W)” and using a java class `glog.business.rate.ratedistance.external.webservice.client.ExternalDistanceWebServiceClient`. Transportation Management as web service client connects to the custom third-party distance engine web service that adheres to the WSDL interface.

validateAddress Operation

Inputs

`validateAddress` operation takes `validateAddressRequest` as input that contain below element as child elements

- `sourceAddress` contains all the location information for the source address.
- `destinationAddress` contains all the location information for the destination address.
- `extEngineAuxInputList` contain SCAC and Route Codes off the locations, contains the equipment dimensions, weight and volume of the equipment being used for transportation and contains hazardous material information being shipped if hazardous routing is enabled. The SCAC and Route Codes are usually only necessary for rail lookups. This element is available as read-only. Mostly the information from this element is not used for address validation.
- `edeParams` is a mapping to all the input parameters that were entered in Oracle Transportation and Global Trade Management Cloud for the specific EDE.

Outputs

`locationAddressMatches` is the result of the `validateAddress` operation. `locationAddressMatches` element contains two child elements.

- `validFlag`: indicate where location is valid or not.
- `possibleMatches`: for engines that support multiple suggested locations, this element contain possible location’s data for given input.

An unsuccessful call to `locationAddressMatches` will return fault to indicate exception scenario with reason being populated in `ExtEngineException` element.

Note: All the values for XML elements involved in web service should be valid as per the Distance Engine WSDL defined in WSDLs appendix section.

External Service Time Calculation

You have the ability to interface Oracle Transportation and Global Trade Management Cloud to any third-party service time engine. You can connect Oracle Transportation and Global Trade Management Cloud as a web service client to the custom web service which adheres to the WSDL interface provided by Oracle Transportation and Global Trade Management Cloud.

You will need to implement a SOAP based web service by following the steps provides in the Web Service Implementation section and using the Distance Engine WSDL provided in the appendix. Register the newly implemented web-service with Oracle Transportation and Global Trade Management Cloud by following the steps provided in Configuration section to be able to access the web service to return distance to Oracle Transportation and Global Trade Management Cloud.

Setup

Ensure if there is a valid External Service Engine (ESE) defined in Oracle Transportation and Global Trade Management Cloud. Define the ESE with the “Webservice(W)” Connection Type. The ESE is the main link between Oracle Transportation and Global Trade Management Cloud and a custom ESE.

External Service Engine

Field	Value	Comments
External Service ID	USER_DEFINED	Identifies the ESE.
Connection Type	Webservice(W)	Defines the type of transport mechanism used.
Implementing Java Class	glog.business.rateservicescheduling.external.webservice.client.ExternalServiceTimeWebServiceClient (this is automatically populated and is a read- only field).	Specifies the fully qualified name of the java implementation class, which links to the external service engine web service as a client.
External System ID	User Defined	External system is required for specifying the web service endpoint and to invoke the web service operations.

In the ESE, you can specify information which will pass from Oracle Transportation and Global Trade Management Cloud into the invocation of the external service time web service via parameters. These parameters are mapped via the params variable in the interface. Use the Power Data page, to define a set of parameters such as source address type, destination address type, and other parameters.

External Service Engine Parameters

Parameter	Value	Country Code	Comments
User Defined	User Defined	User Defined * for all countries	These parameters are available to web services and can be used for program control and/or data.

Web Service Explanation

Use ESE to define a third party external service engine with the WebService(W) Connection Type and a java class `glog.business.rateservicescheduling.external.webservice.client.ExternalServiceTimeWebServiceClient`. Oracle Transportation and Global Trade Management Cloud as a web service client connects to the custom third-party service engine web service that adheres to the WSDL interface.

lookupServiceTime Operation

Inputs

`lookupServiceTime` operation takes `lookupServiceTimeRequest` as input that contains child elements below:

- `sourceAddress` contains all the location information for the source address.
- `destinationAddress` contains all the location information for the destination address.
- `extEngineAuxInputList` contains SCAC, Route Codes of the locations, equipment dimensions, weight and volume of the equipment used for transportation and hazardous material information shipped if hazardous routing is enabled. The SCAC and Route Codes are necessary for rail lookups. This element is available as read-only.
- `params` is a mapping to all the input parameters that were entered in Oracle Transportation and Global Trade Management Cloud for the specific ESE.

Outputs

`extEngineServiceTime` is the result of the `lookupServiceTime` operation. The `extEngineServiceTime` element contains two child elements named `uom` and `amount`. An unsuccessful call to `lookupServiceTime` returns the error message which indicates the exception scenario with the reason populated in the `ExtEngineException` element.

Note: All the values for the XML elements involved in web service must be valid as per Oracle Transportation and Global Trade Management Cloud WSDL.

5 External Rating Engine (ERE)

External Rate Calculation

You have the ability to interface Oracle Transportation and Global Trade Management Cloud to any third-party rating engine by connecting Oracle Transportation and Global Trade Management Cloud as a web service client to the custom web service that adheres to the WSDL interface provided by Oracle Transportation and Global Trade Management Cloud.

You will need to implement a SOAP based web service by following the steps provides in the Web Service Implementation section and using the Distance Engine WSDL provided in the appendix. Register the newly implemented web-service with Oracle Transportation and Global Trade Management Cloud by following the steps provided in Configuration section to be able to access the web service to return distance to Oracle Transportation and Global Trade Management Cloud.

Setup

The first step is to ensure there is a valid External Rating Engine (ERE) Fieldset defined in Oracle Transportation and Global Trade Management Cloud. The External Rating Engine Fieldset has to be defined with a Connection Type as “Webservice (W)”. External Rating Engine (ERE) is the main link between Oracle Transportation and Global Trade Management Cloud and a custom ERE.

External Rating Engine Fieldsets

Field	Value	Comments
Fieldset ID	USER_DEFINED	Identifies the fieldset.
Valid External Rating Engine ID	GENERIC – a predefined external rating engine ID.	Identifies the external rating engine that can utilize this particular fieldset.
Connection Type	Webservice(W)	Defines the type of transport mechanism used.
Implementing Java Class	glog.business.rate.rateengine.external.webservice.client.ExternalRatingWebServiceClient (this value is auto populated and is read only).	Specifies the fully qualified name of the java class, which will link to the external rating engine web service as a client.
External System ID	User Defined	External system is required for specifying the web service endpoint and the web service operations to invoke.

The External Rating Engine Fieldset also specifies the information that will be passed from Oracle Transportation and Global Trade Management Cloud into the invocation of the external engine by defining a list of one or more Rate Basis Items (RBIs). Using the lower half of the Fieldset Power Data page, you can select a set of RBIs and include them in the fieldset by clicking on one of the right-pointing arrows. These RBI references (listed below under the GID column) are stored in the external_rating_engine_fieldset_d table.

RBI References

RBI	GID
Shipment Number of Line Items	SHIPMENT.NUMLINES
Shipment Source Postal Code	SHIPMENT.SOURCE.POSTAL_CODE
Shipment Destination Postal Code	SHIPMENT.DEST.POSTAL_CODE
Rate Offering Service Provider	SHIPMENT.RATE.SERVPROV
Rate Offering Rate Service Calculator	SHIPMENT.RATE.RATE_SERVICE
Shipment Line Item Weight	SHIPMENT.LINES.WEIGHT

There are two ways to invoke an external rating engine using Oracle Transportation and Global Trade Management Cloud:

Rate Offering Method

This method attaches the invocation of an external rating engine to the rate offering. Current, purpose-built external rating engines are invoked using this method. An example would be the current implementation of the Oracle Transportation and Global Trade Management Cloud link to SMC's LTL RatewareXL web service rating engine. This method requires a rate offering to specify a rate offering type that includes the EXTERNALRATINGENGINE rate attribute in its costing sequence. The following fields must be specified on the Attributes page of the Rate Manager UI:

Rate Manager Configuration

Field	Value	Comments
External Rating Engine ID	GENERIC	Select the GENERIC external rating engine.
External Engine Fieldset ID	USER_DEFINED	Choose the fieldset defined for the external engine that invokes the accessorial calculations.

Rate Cost Method

It is possible to specify the invocation of an external rating engine as a rate cost during the processing of standard costs, or during the processing of accessorial costs. The rate cost UI contains a Cost Type dropdown list. Choosing the value of **External** allows the specification of the same external rating engine fields as found on the Rate Manager UI:

Rate Costs or Accessorial Cost Method

Rate Costs or Accessorial Cost Method

Field	Value	Comments
Cost Type	External	This is a field that explicitly defines the type of calculation described by the cost record.
External Rating Engine ID	GENERIC	Select the GENERIC external rating engine.

Field	Value	Comments
External Engine Fieldset ID	USER_DEFINED	Choose the fieldset defined for the external engine that invokes the calculations.

Optional – Service Time

There is also an option to return the service time from the external rating engine. For this type of rate service, the rate engine will return the specific pickup time and delivery time for a shipment or the number of days the service would take. The rate service engine uses the pickup and delivery time or the service days to test whether it will fit into the order’s pickup and delivery windows. If not, this rate cannot be used.

There are two ways to return service time information from a rating engine:

- The external rating engine can fill in the estimated pickup and delivery times. The RATE SERVICE associated with the rate must have a service type of TIMEDEFINITESERVICE.
- The external rating engine can fill the service Days field. The RATE SERVICE associated with the rate must have a service type of EXTERNALTRANSITDAYS.

If the rate service doesn’t have the above service types, then the service time data returned from the rating engine will be ignored.

For this type of rate service, location calendar, location capacity, and carrier capacity are not considered. Pickup or delivery activity time at a stop is always 0.

The following assumptions also hold true:

- Only applies to 2-stop shipments
- It does not make sense to have HOLD AS LATE AS POSSIBLE set to TRUE regardless of the setting of the parameter HOLD AS LATE AS POSSIBLE, this type of rate service is always calculated in HOLD AS LATE AS POSSIBLE = false.
- Order’s pickup and delivery windows should be a whole day. If the orders have a narrow pickup and delivery window, the specific pickup time and delivery time will fall out of the order’s windows, so it will fail to use this rate. For example, the order’s delivery window is 10:00-11:00am, but rate engine come back with a delivery time of 4:00PM, then rate service will fail this rate.
- It only applies to single leg shipments: i.e., shipment with rate service type cannot have upstream and downstream shipment.

Web Service Calculation

A third-party external rating engine can be defined using ERE Fieldset with the connection type of “WebService (W)” which uses a Java class `glog.business.rate.rateengine.external.webservice.client.ExternalRatingWebServiceClient`. Oracle Transportation and Global Trade Management Cloud as web service client connects to the custom third-party rating engine web service that adheres to the Rating Engine WSDL provided in WSDLs appendix section.

costShipment Operation

Inputs:

There is one input element called `inputDataList` which contains a list of `inputData` as its child element. `inputData` uses a name attribute to store RBI Name and a list of values for the RBI as its child elements. It will be similar to this:

```
<n:inputData name="SHIPMENT.LINES.COMMODITY">
  <n:values>
    <n:value>COMMODITY1</n:value>
    <n:value>COMMODITY2</n:value>
  </n:values>
</n:inputData>
<n:inputData name="SHIPMENT.LINES.WEIGHT">
  <n:values>
    <n:value>12.0 LB</n:value>
    <n:value>14.0 LB</n:value>
  </n:values>
</n:inputData>
```

If the shipment has two lines with two different commodity codes and you want to get the commodity code and weight, you will have to select two RBI. Line Item Commodity Code (SHIPMENT.LINES.COMMODITY), Line Item Weight (SHIPMENT.LINES.WEIGHT). The input element will have both the commodity and weight elements separately but in the same order i.e. COMMODITY1 weight is 12.0 LB and COMMODITY2 weight is 14.0 LB.

Outputs:

`rexRateResult` has the following child elements

- **err:** If this is set, this shows an error in the log but will not throw an exception.
- **serviceDays:** If the ERE returns the number of service days, this can be set here. Otherwise, a value of NULL equals no data.
- **utcPickupDateTime:** Estimated pickup time, UTC. The type of this variable is long. The value should be in milliseconds i.e. number of milliseconds of this time since January 1, 1970, 00:00:00 GMT. The section “utcPickupDateTime and utcDeliveryDateTime elements” provides more information. If both `utcPickupDateTime` and `utcPickupDateTimeStdFormat`, then `utcPickupDateTimeStdFormat` takes precedence.
- **utcPickupDateTimeStdFormat:** Estimated pickup time. The type of variable is `xs:dateTime`. When `xs:dateTime` is used to define a date or date time field for a SOAP Web Service, `wsimport` tool will use java class `XMLGregorianCalendar` object in the java client and the default formatting for `XMLGregorianCalendar` object is `yyyy-MM-dd'T'HH24:mm:ss.SSSZ`. Hence, provide the date time in this format. One more thing to note here is that the 'Z' in the format represents time zone offset (E.g. +2:00, +3:00, -4:00, -5:30, +5:30 etc...) and it is optional. If the timezone offset is provided, `pickupTimeZone` element should not be provided. If both `utcPickupDateTime` and `utcPickupDateTimeStdFormat`, then `utcPickupDateTimeStdFormat` takes precedence.
- **pickupTimeZone:** Max of 50 characters, java-style time zone identifier (ex. EDT, America/Chicago, etc.). Provide this element only if the time zone offset is not provided in `utcPickupDateTimeStdFormat` element.
- **utcDeliveryDateTime:** Estimated delivery time, UTC. The type of this variable is long. The value should be in milliseconds i.e. number of milliseconds of this time since January 1, 1970, 00:00:00 GMT. The section “utcDeliveryDateTime and utcDeliveryDateTime elements” provides more information. If both `utcDeliveryDateTime` and `utcDeliveryDateTimeStdFormat`, then `utcDeliveryDateTimeStdFormat` takes precedence.
- **utcDeliveryDateTimeStdFormat:** Estimated delivery time. The type of variable is `xs:dateTime`. When `xs:dateTime` is used to define a date or date time field for a SOAP Web Service, `wsimport` tool will use java

class `XMLGregorianCalendar` object in the java client and the default formatting for `XMLGregorianCalendar` object is `yyyy-MM-dd'T'HH24:mm:ss.SSSZ`. Hence, provide the date time in this format. One more thing to note here is that the 'Z' in the format represents time zone offset (E.g. +2:00, +3:00, -4:00, -5:30, +5:30 etc...) and it is optional. If the timezone offset is provided, `pickupTimeZone` element should not be provided. If both `utcDeliveryDateTime` and `utcDeliveryDateTimeStdFormat`, then `utcDeliveryDateTimeStdFormat` takes precedence.

- **deliveryTimezone**: Max of 50 characters. Java-style time zone identifier (ex. EDT, America/Chicago, etc.). Provide this element only if the time zone offset is not provided in `utcDeliveryDateTimeStdFormat` element.
- **chargeableWeight**: Chargeable weight. Copied to the shipment's chargeable weight field. Only used when invocation of ERE is specified on the rate offering.
- **String chargeableWeightUom**: Unit of measure for the chargeable weight (ex. LB, KG, etc.)
- **dimWeight**: Dimensional weight. Copied to the shipment's dimensional weight field. Only used when invocation of ERE is specified on the rate offering.
- **dimWeightUom**: Unit of measure for the dimensional weight.
- **costDetails**: List of `costDetail` Element where each `costDetail` is about each individual cost calculated by the external rating engine.
- **shipmentRefnums**: List of `shipmentRefnum` elements where each `shipmentRefnum` is a pair of `shipmentRefnumQualifierGid` and `ShipmentRefnumValue`. The `shipmentRefnumQualifierGid` needs to be configured to a pre-defined qualifier GID. Whenever a shipment is rated and one of the rates uses this ERE, these reference numbers will be stored on the shipment as reference numbers. For example, the application can return a quote ID and persist it as a reference number on the shipment.

utcPickupDateTime and utcDeliveryDateTime elements

The `utcPickupDateTime` and `utcDeliveryDateTime` is of "long" type. If we send the time as a date string, it will be setting it to '0' in Transportation and Global Trade Management Cloud. To make it work, we have to convert the date string `utcPickupDateTime` and `utcDeliveryDateTime` to milliseconds and replace the dates with the milliseconds in the response XML.

Please refer the sample code to convert date of 2019-08-06T16:30 to milliseconds:

```
import java.util.Calendar;
import java.util.Date;
public class Sample {
    public static void main(String []args) {
        convertDateToMillis(2019, Calendar.AUGUST, 06, 16, 30, 00);
    }
    public static long convertDateToMillis(int year, int month, int date, int hour, int minutes, int seconds) {
        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.YEAR, year);
        cal.set(Calendar.MONTH, month);
        cal.set(Calendar.DATE, date);

        cal.set(Calendar.HOUR_OF_DAY, hour);
        cal.set(Calendar.MINUTE, minutes);
        cal.set(Calendar.SECOND, seconds);

        Date fullDate = cal.getTime();

        System.out.println("The date is : "+fullDate);
        System.out.println("Millisecons : "+fullDate.getTime()); // date to milliseconds

        return fullDate.getTime();
    }
}
```

costDetail:

- **cost:** Numeric value of the final calculated cost.
- **currencyCode:** Currency code of the cost (USD, GBP, etc.)
- **costType:** One of Base, A - Accessorial, O - Other, D - Discount, S - Stop-off, C - Circuity, L - Delta.
- **accessorialCode:** If this cost is an accessorial, the accessorial code is specified here. This code must exist in Oracle Transportation and Global Trade Management Cloud.
- **costCode:** Secondary accessorial code for classification purposes only. This code must exist in Oracle Transportation and Global Trade Management Cloud.
- **specialServiceCode:** If this cost is a special service, the special service code is specified here. This code must exist in Oracle Transportation and Global Trade Management Cloud.
- **isWeightedCostOnly:** Some costs are calculated as weighting factors that do not affect the actual final cost of a shipment. Set this field to true if this is a weighted cost.
- **calculationDetails:** A list of simple text strings that the ERE implementer can use to give detailed information on how a cost was calculated.

shipmentRefnum:

- **shipmentRefnumQualifierGid:** A valid shipment reference number qualifier gid defined in OTM.
- **shipmentRefnumValue:** A reference number value defined for the qualifier.

Once the input is defined and an actual lookup is complete, the cost shipment method needs to return the `rexRateResult`.

Note: All the values for XML elements involved in web service should be valid as per Oracle Transportation and Global Trade Management Cloud WSDL.

6 External Web Service Engine

Overview

A third-party external engine that implement the Distance Engine WSDL or Rating Engine WSDL or Service Time Engine WSDL provided by Oracle Transportation and Global Trade Management Cloud can be defined as web services.

To generate a custom web services from WSDL following are the steps:

1. Generate Stubs
2. Service End Point Implementation
3. Compile and Package Service Artifacts
4. Deploy Web Service

For more information on how to create or generate web service please refer to [The Java Web Services Tutorial](#).

The defined procedure in the document is not the only way to create a web service. You can create JAXB Objects based on the WSDL definitions and use `@WebServiceProvider` annotation to create a web service. It is recommended to use JAX-WS to create the web service.

All the instruction provides are using `DistanceEngineService.wsdl` as an example.

Generate Stubs

Using `wsimport` create the stubs using the WSDL:

```
<java.home>/bin/wsimport -keep -d <directory where the source files are generated> -p  
'test.service.distanceengine' DistanceEngineService.wsdl
```

The following files will be generated in `/test/service/distanceengine/` directory.

- AddressLines.java
- DistanceEngineService.java
- EdeParams.java
- ExtEngineAddress.java
- ExtEngineAuxInputList.java
- ExtEngineDistance.java
- ExtEngineException.java
- ExternalDistanceEngine.java
- KeyValue.java
- LocationAddressMatches.java
- LookupDistanceRequest.java
- ObjectFactory.java

- package-info.java
- PossibleMatches.java
- ValidateAddressRequest.java

For more details on using `wsimport` please refer to [wsimport - Java™ API for XML Web Services \(JAX-WS\) 2.0](#).

Service End Point Implementation

In the above classes `ExternalDistanceEngine.java` is an interface which represents the service endpoint. Create an implementation class implementing this interface. Add `@WebService` annotation to this implementation class. A sample implementation class is provided below

```
@WebService(targetNamespace = "http://xmlns.oracle.com/apps/otm/distanceengine", endpointInterface =
"test.service.distanceengine.ExternalDistanceEngine",
serviceName = "DistanceEngineService", portName = "ExternalDistanceEnginePort" )
public class DistanceEngineServerImpl implements ExternalDistanceEngine {

    @Override
    public ExtEngineDistance lookupDistance(LookupDistanceRequest lookupDistanceRequest) throws
ExtEngineException {

    }

    @Override
    public LocationAddressMatches validateAddress(ValidateAddressRequest validateAddressRequest) throws
ExtEngineException {

    }
}
```

Compile and Package Service Artifacts

Compile the entire generated stubs and the Service Endpoint Implementation classes and create a JAR file with the above generated classes and place the JAR file in a WAR file.

Deploy

Once the WAR file is available, deploy this to a weblogic server (non Oracle Transportation and Global Trade Management Cloud). You can access the web service using the below URL of the form

```
http://<host>:<port>/<webcontext>/<servicename>
```

To access the WSDL

```
http://<host>:<port>/<webcontext>/<servicename>?wsdl
E.g. http://localhost:8001/edeservice/DistanceEngineService?wsdl
```


7 Configuration

Configure External Web Service Engine in Oracle Transportation and Global Trade Management Cloud

Follow the below topics to access the External Web Service from Oracle Transportation and Global Trade Management Cloud.

Document

Create a document under **Business Process Automation > Document Manager** to represent the web service created and deployed in the above chapter.

1. Create a new document and provide the web service URL.
2. No need to associate the document with any Document Type or Content Management System.

Web Service

Create a web service definition under **Business Process Automation > Communication Management > Web Services** from the document created above.

1. Create a new Web Service.
2. Provide the document created above.
3. Click on the service detail.
4. Check the Use Web service security to add the user credentials in the Username token in the **SOAP** Header.
5. In the Web service endpoints, if the web service is secured i.e. uses https, then we should make sure to ensure the Oracle Transportation and Global Trade Management Cloud System Administrator has configured the application server to be able to accept the external server certificate.
6. Also enter the authentication username and password.

External System

Create external system under **Business Process Automation > Communication Management > External Systems**, using the web service definition created above and select the appropriate web service operation.

1. For distance, create an External System referencing the web service using `lookupDistance` operation.
2. For address validation, create an External System referencing the web service using `validateAddress` operation.
3. For service time, create an External System referencing the web service using `lookupServiceTime` operation.

4. For cost, create an External System referencing the web service using `costShipment` operation.

External Engine

Define the External Distance or External Service Time or Rating Engine:

1. Choose **Connection Type** as web service (W).
2. JavaClass:
 - o For distance engines, the class `glog.business.rate.ratedistance.external.webservice.client.ExternalDistanceWebServiceClient` will be populated. This field is not editable.
 - o For service time engines, the class `glog.business.rateservicescheduling.external.webservice.client.ExternalServiceTimeWebServiceClient` will be populated. This field is not editable.
 - o For rating engines, the class `glog.business.rate.rateengine.external.webservice.client.ExternalRatingWebServiceClient` will be populated. This field is not editable.
3. External System ID: This is mandatory for web service. Choose the external system which is going to act as a web service.

8 Appendix: WSDLs

Distance Engine

```
<?xml version='1.0' encoding='UTF-8'?>

<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://xmlns.oracle.com/apps/otm/distanceengine"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
targetNamespace="http://xmlns.oracle.com/apps/otm/distanceengine"
name="DistanceEngineService">
  <types>
    <xs:schema xmlns:tns="http://xmlns.oracle.com/apps/otm/distanceengine" elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/otm/distanceengine">

      <xs:complexType name="keyValue">
        <xs:annotation>
          <xs:documentation>
            This acts like place holder for requirement similar to Map with a key and value like SCAC code, Weight_UOM,
            Length_UOM etc...
          </xs:documentation>
        </xs:annotation>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="value" type="xs:string"/>
      </xs:complexType>
      <xs:complexType name="keyValues">
        <xs:annotation>
          <xs:documentation>
            This acts like place holder for requirement similar to Map with a key and list of values like HAZMAT_TYPES.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element ref="tns:values"/>
        </xs:sequence>
        <xs:attribute name="name" use="required" type="xs:string"/>
      </xs:complexType>
      <xs:element name="values">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="tns:value" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="value" type="xs:string"/> <xs:element name="addressLine" type="xs:string"/>
      <xs:element name="accuracy" type="xs:string" />
      <xs:element name="city" type="xs:string" />
      <xs:element name="countryCode" type="xs:string" />
      <xs:element name="countyCode" type="xs:string" />
      <xs:element name="latitude" type="xs:double" nillable='true' />
      <xs:element name="locationName" type="xs:string" />
      <xs:element name="longitude" type="xs:double" nillable='true' />
      <xs:element name="postalCode" type="xs:string" />
      <xs:element name="province" type="xs:string" />
      <xs:element name="provinceCode" type="xs:string" />
      <xs:element name="railSplc" type="xs:string" />
      <xs:element name="railStationCode" type="xs:string" />
      <xs:element name="seqNumber" type="xs:int" nillable='true' />
    
```

```
<xs:element name="shortPostalCode" type="xs:string" />
<xs:element name="possibleMatch" type="tns:extEngineAddress"/>
<xs:element name="validFlag" type="xs:boolean" default='false' />
<xs:element name="sourceAddress" type="tns:extEngineAddress"/>
<xs:element name="destinationAddress" type="tns:extEngineAddress"/>
<xs:element name="address" type="tns:extEngineAddress" />
<xs:element name="uom" type="xs:string"/>
<xs:element name="amount" type="xs:double" nillable='true' />

<xs:element name="ExtEngineException" type="xs:string" >
<xs:annotation>
<xs:documentation>
Message contain information to be shown to user when fault occurs.
</xs:documentation>
</xs:annotation>
</xs:element>

<xs:element name="edeParam" type="tns:keyValue">
<xs:annotation>
<xs:documentation>
<!-- TODO: Had not identified the use of this constants -->
The valid values to name attribute are
SOURCE_ADDRESS_TYPE
DEST_ADDRESS_TYPE
SAME_SOURCE_DEST_DIST
</xs:documentation>
</xs:annotation>
</xs:element>

<xs:element name="extEngineAuxInput" type="tns:keyValue" >
<xs:annotation>
<xs:documentation>
The valid values to name attribute are
SCAC,
RouteCode,
Height,
Height_UOM,
Length,
Length_UOM,
Width,
Width_UOM,
Weight,
Weight_UOM
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="extEngineAuxValueList" type="tns:keyValues">
<xs:annotation>
<xs:documentation>
The valid values to name attribute are
HAZMAT_TYPES
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="addressLines">
<xs:complexType>
<xs:sequence>
<xs:element ref='tns:addressLine' maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="extEngineAddress">
<xs:sequence>
<xs:element ref="tns:accuracy" />
<xs:element ref="tns:addressLines" />
```

```
<xs:element ref="tns:city" />
<xs:element ref="tns:countryCode" />
<xs:element ref="tns:countryCode" />
<xs:element ref="tns:latitude" />
<xs:element ref="tns:locationName" />
<xs:element ref="tns:longitude" />
<xs:element ref="tns:postalCode" />
<xs:element ref="tns:province" />
<xs:element ref="tns:provinceCode" />
<xs:element ref="tns:railSplc" />
<xs:element ref="tns:railStationCode" />
<xs:element ref="tns:seqNumber" />
<xs:element ref="tns:shortPostalCode" />
</xs:sequence>
</xs:complexType>

<xs:element name="possibleMatches">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:possibleMatch" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="locationAddressMatches">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:possibleMatches"/>
<xs:element ref="tns:validFlag" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="extEngineAuxInputList">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:extEngineAuxInput" maxOccurs="unbounded"/>
<xs:element ref="tns:extEngineAuxValueList" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="edeParams" >
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:edeParam" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="lookupDistanceRequest">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:sourceAddress" />
<xs:element ref="tns:destinationAddress" />
<xs:element ref="tns:extEngineAuxInputList"/>
<xs:element ref="tns:edeParams"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="extEngineDistance">
<xs:annotation>
<xs:documentation>the values for UOM element should be valid unit of measure in OTM.</xs:documentation>
</xs:annotation>
<xs:complexType>
```

```
<xs:sequence>
<xs:element ref="tns:uom" />
<xs:element ref="tns:amount" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="validateAddressRequest">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:address" />
<xs:element ref="tns:extEngineAuxInputList" />
<xs:element ref="tns:edeParams"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
</types>

<message name="lookupDistanceRequest">
<part name="lookupDistanceRequest" element="tns:lookupDistanceRequest"/>
</message>

<message name="lookupDistanceResponse">
<part name="extEngineDistance" element="tns:extEngineDistance"/>
</message>

<message name="ExtEngineException">
<part name="fault" element="tns:ExtEngineException"/>
</message>

<message name="validateAddressRequest">
<part name="validateAddressRequest" element="tns:validateAddressRequest"/>
</message>

<message name="validateAddressResponse">
<part name="locationAddressMatches" element="tns:locationAddressMatches"/>
</message>

<portType name="ExternalDistanceEngine">
<operation name="lookupDistance" >
<input message="tns:lookupDistanceRequest" />
<output message="tns:lookupDistanceResponse" />
<fault message="tns:ExtEngineException" name="ExtEngineException"/>
</operation>
<operation name="validateAddress">
<input message="tns:validateAddressRequest" />
<output message="tns:validateAddressResponse" />
<fault message="tns:ExtEngineException" name="ExtEngineException"/>
</operation>
</portType>

<binding name="ExternalDistanceEnginePortBinding" type="tns:ExternalDistanceEngine">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="lookupDistance">
<soap:operation soapAction="lookupDistance"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ExtEngineException">
<soap:fault name="ExtEngineException" use="literal"/>
</fault>
</operation>
```

```
<operation name="validateAddress">
<soap:operation soapAction="validateAddress"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ExtEngineException">
<soap:fault name="ExtEngineException" use="literal"/>
</fault>
</operation>
</binding>
<service name="DistanceEngineService">
<port name="ExternalDistanceEnginePort" binding="tns:ExternalDistanceEnginePortBinding">
<soap:address location="http://localhost/GC3Services/DistanceEngineService"/>
</port>
</service>
</definitions>
```

Rating Engine

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://xmlns.oracle.com/apps/otm/ExternalRating"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
targetNamespace="http://xmlns.oracle.com/apps/otm/ExternalRating"
name="ExternalRatingService">

<types>
<xs:schema xmlns:tns="http://xmlns.oracle.com/apps/otm/ExternalRating"
elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/otm/ExternalRating">

<xs:complexType name="keyValues">
<xs:annotation>
<xs:documentation> Name, Values elements </xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element ref="tns:values"/>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>

<xs:simpleType name="costType">
<xs:annotation>
<xs:documentation>CostType specifies which kind of cost was calculated.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:enumeration value="Base"/>
<xs:enumeration value="Accessorial"/>
<xs:enumeration value="Discount"/>
<xs:enumeration value="SpecialService"/>
</xs:restriction>
</xs:simpleType>

<xs:element name="value" type="xs:string" />
<xs:element name="inputData" type="tns:keyValues"/>
<xs:element name="err" type="xs:string" />
```

```
<xs:element name="serviceDays" type="xs:int" nillable='true' />
<xs:element name="utcPickupDateTime" type="xs:long" />
<xs:element name="utcPickupDateTimeStdFormat" type="xs:dateTime" />
<xs:element name="pickupTimezone" type="xs:string" />
<xs:element name="utcDeliveryDateTime" type="xs:long" />
<xs:element name="utcDeliveryDateTimeStdFormat" type="xs:dateTime" />
<xs:element name="deliveryTimezone" type="xs:string" />
<xs:element name="chargeableWeight" type="xs:double" nillable='true' />
<xs:element name="chargeableWeightUom" type="xs:string" />
<xs:element name="dimWeight" type="xs:double" nillable='true' />
<xs:element name="dimWeightUom" type="xs:string" />
<xs:element name="cost" type="xs:double" nillable='true' />
<xs:element name="currencyCode" type="xs:string" />
<xs:element name="costType" type="tns:costType" />
<xs:element name="accessorialCode" type="xs:string" />
<xs:element name="costCode" type="xs:string" />
<xs:element name="specialServiceCode" type="xs:string" />
<xs:element name="calculationDetail" type="xs:string" />
<xs:element name="isWeightedCostOnly" type="xs:boolean" />
<xs:element name="costCategoryGid" type="xs:string" />
<xs:element name="shipUnitGid" type="xs:string" />
<xs:element name="shipUnitLineSeq" type="xs:long" />

<xs:element name="values">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='tns:value' maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="calculationDetails">
  <xs:complexType >
    <xs:sequence>
      <xs:element ref="tns:calculationDetail" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="costDetail" >
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>
        This class contains detailed information about an individual calculated cost. An invocation of an External Rating Engine can return any number of CostDetail records.
        cost:
          Numeric value of the calculated cost.
        currencyCode:
          String representing the currency of the calculated cost. Must be a valid currencyCode as understood by OTM. (ex. USD, BGP, etc.)
        costType:
          Type of calculated cost. One of Base, Accessorial, Other, Discount, Stop-off, Circuitry, Delta.
        accessorialCode:
          If costType is Accessorial, this specifies the accessorial code associated with it. Code must be a valid OTM accessorial code.
        costCode:
          Alternative code for classification purposes only. Applies to all cost types. Code must be a valid OTM accessorial code.
        specialServiceCode:
          If costType is SpecialService, this specifies the special service code associated with it. Code must be a valid OTM special service code.
        calculationDetails:
          Collection of String containing details about how the calculation was performed. There are no restrictions on the generated strings, but OTM will only store a maximum of 4000 characters per line.
        isWeightedCostOnly:
      </xs:documentation>
    </xs:annotation>
  </xs:complexType>
</xs:element>
```


Flag to indicate that the cost should only affect the final weighted cost of a shipment. If true, the cost will not be added to the base or total cost.

costCategoryGid:

String specifying a Cost Category Code GID. This is a code that allows, within OTM, the ability to specify a set of category codes for the purpose of filtering out costs during the rating process. This is primarily used for driver assignment during fleet operations. It is included here for informational purposes only. This value can be viewed as part of the results of an RIQ query.

sShipUnitGid:

Linking of the costs for ship unit or ship unit line is not supported for external rating web service.

sShipUnitLineSeq:

Linking of the costs for ship unit or ship unit line is not supported for external rating web service.

```

</xs:documentation>
</xs:annotation>
<xs:sequence>
  <xs:element ref="tns:cost"/>
  <xs:element ref="tns:currencyCode" />
  <xs:element ref="tns:costType" />
  <xs:element ref="tns:accessorialCode" />
  <xs:element ref="tns:costCode" />
  <xs:element ref="tns:specialServiceCode" />
  <xs:element ref="tns:calculationDetails" />
  <xs:element ref="tns:isWeightedCostOnly" />
  <xs:element ref="tns:costCategoryGid" />
  <xs:element ref="tns:shipUnitGid" />
  <xs:element ref="tns:shipUnitLineSeq"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="costDetails">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:costDetail" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="inputDataList">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>
        InputData Map of RBI ID to List of Strings representing the shipment data required
        by the external rating engine. EX:
        {SHIPMENT.NUMLINES,3},
        {SHIPMENT.SOURCE.POSTAL_CODE,19406},
        {SHIPMENT.DEST.POSTAL_CODE,34639},
        {SHIPMENT.RATE.SERVPROV,UPS },
        {SHIPMENT.RATE.RATE_SERVICE,UPS GROUND},
        {SHIPMENT.RATE.SERVPROV_ACCOUNT_NUMBER, 12345634},
        {SHIPMENT.LINES.WEIGHT,{15 LB,10 LB}},
        The RBIs and data present in the map is determined by the External Rating Engine Fieldset specified in OTM
        and associated with the calling Rate Offering or Rate Cost/Accessorial Cost.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="tns:inputData" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="rexRateRequest">
  <xs:complexType >
    <xs:sequence>
      <xs:element ref="tns:inputDataList"/>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>

<xs:element name="rexRateResult">
<xs:complexType>
<xs:annotation>
<xs:documentation>
Describes the result of an invocation of an External Rating Engine
err:
Existence of error message string indicates failure The error message will be logged, but will not cause an
exception.
serviceDays:
If the external rating engine calculates the number of service days, it can be returned here. A value of
null indicates no data
utcPickupDateTime:
If the external rating engine calculates estimated pickup and delivery times, they can be returned
here.Values of null indicate no data. Timezones can be specified if necessary, otherwise, UTC will be
assumed.
utcPickupDateTimeStdFormat:
Estimated pickup time. The type of variable is xs:dateTime. When xs:dateTime is used to define a date or
date time field for a SOAP Web Service, wsimport tool will use java class XMLGregorianCalendar object in
the java client and the default formatting for XMLGregorianCalendar object is yyyy-MM-dd'T'HH24:mm:ss.SSSZ.
Hence, provide the date time in this format. One more thing to note here is that the 'Z' in the format
represents time zone offset (E.g. +2:00, +3:00, -4:00, -5:30, +5:30 etc...) and it is optional. Do not
provide the pickupTimeZone element if this offset is provided.
pickupTimezone:
Max of 50 characters. Java-style time zone identifier (ex. EDT, America/Chicago, etc.). Provide this
element only if the time zone offset is not provided in utcPickupDateTimeStdFormat element.
utcDeliveryDateTime:
If the external rating engine calculates estimated pickup and delivery times, they can be returned here.
Values of null indicate no data. Timezones can be specified if necessary, otherwise, UTC will be assumed.
utcDeliveryDateTimeStdFormat:
Estimated delivery time. The type of variable is xs:dateTime. When xs:dateTime is used to define a date or
date time field for a SOAP Web Service, wsimport tool will use java class XMLGregorianCalendar object in
the java client and the default formatting for XMLGregorianCalendar object is yyyy-MM-dd'T'HH24:mm:ss.SSSZ.
Hence, provide the date time in this format. One more thing to note here is that the 'Z' in the format
represents time zone offset (E.g. +2:00, +3:00, -4:00, -5:30, +5:30 etc...) and it is optional. Do not
provide the pickupTimeZone element if this offset is provided.
deliveryTimezone:
Max of 50 characters. Java-style time zone identifier (ex. EDT, America/Chicago, etc.). Provide this
element only if the time zone offset is not provided in utcDeliveryDateTimeStdFormat element.
chargeableWeight:
Chargeable weight. Copied to the Shipments chargeable weight field. Only used when invocation of ERE is
specified on the Rate Offering
chargeableWeightUom:
Unit of Measure for the chargeable weight (ex. LB, KG, etc.)
dimWeight:
Dimensional weight. Copied to the Shipments dimensional weight field. Only used when invocation of ERE is
specified on the Rate Offering
dimWeightUom:
Unit of Measure for the dimensional weight (ex. LB, KG, etc.)
</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element ref="tns:err" />
<xs:element ref="tns:costDetails"/>
<xs:element ref="tns:serviceDays" />
<xs:element ref="tns:utcPickupDateTime" />
<xs:element ref="tns:utcPickupDateTimeStdFormat" />
<xs:element ref="tns:pickupTimezone" />
<xs:element ref="tns:utcDeliveryDateTime" />
<xs:element ref="tns:utcDeliveryDateTimeStdFormat" />
<xs:element ref="tns:deliveryTimezone" />
<xs:element ref="tns:chargeableWeight"/>
<xs:element ref="tns:chargeableWeightUom" />
<xs:element ref="tns:dimWeight" />
<xs:element ref="tns:dimWeightUom" />

```

```
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>
</types>

<message name="costShipmentRequest">
<part name="rexRateRequest" element="tns:rexRateRequest"/>
</message>

<message name="costShipmentResponse">
<part name="rexRateResult" element="tns:rexRateResult"/>
</message>

<portType name="ExternalRating">
<operation name="costShipment">
<input wsam:Action="http://xmlns.oracle.com/apps/otm/rexExternalService/costShipmentRequest"
message="tns:costShipmentRequest"/>
<output wsam:Action="http://xmlns.oracle.com/apps/otm/rexExternalService/costShipmentResponse"
message="tns:costShipmentResponse"/>
</operation>
</portType>

<binding name="ExternalRatingPortBinding" type="tns:ExternalRating">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="costShipment">
<soap:operation soapAction="costShipment"/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>

<service name="ExternalRatingService">
<port name="ExternalRatingPort" binding="tns:ExternalRatingPortBinding">
<soap:address location="http://localhost/GC3Services/REXPublicExternalService"/>
</port>
</service>
</definitions>
```

Service Time Engine

```
<?xml version='1.0' encoding='UTF-8'?>

<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://xmlns.oracle.com/apps/otm/servicetimeengine"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
targetNamespace="http://xmlns.oracle.com/apps/otm/servicetimeengine"
name="ServiceTimeEngineService">
<types>
<xs:schema xmlns:tns="http://xmlns.oracle.com/apps/otm/servicetimeengine" elementFormDefault="qualified"
targetNamespace="http://xmlns.oracle.com/apps/otm/servicetimeengine">

<xs:complexType name="keyValue">
<xs:annotation>
```

```

<xs:documentation>
This acts like place holder for requirement similar to Map with a key and value like SCAC code, Weight_UOM,
Length_UOM etc...
</xs:documentation>
</xs:annotation>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="value" type="xs:string"/>
</xs:complexType>
<xs:complexType name="keyValues">
<xs:annotation>
<xs:documentation>
This acts like place holder for requirement similar to Map with a key and list of values like HAZMAT_TYPPES.
</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element ref="tns:values"/>
</xs:sequence>
<xs:attribute name="name" use="required" type="xs:string"/>
</xs:complexType>
<xs:element name="values">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:value" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="value" type="xs:string"/>
<xs:element name="addressLine" type="xs:string"/>
<xs:element name="accuracy" type="xs:string" />
<xs:element name="city" type="xs:string" />
<xs:element name="countryCode" type="xs:string" />
<xs:element name="countyCode" type="xs:string" />
<xs:element name="latitude" type="xs:double" nillable='true' />
<xs:element name="locationName" type="xs:string" />
<xs:element name="longitude" type="xs:double" nillable='true'/>
<xs:element name="postalCode" type="xs:string" />
<xs:element name="province" type="xs:string" />
<xs:element name="provinceCode" type="xs:string" />
<xs:element name="railSplc" type="xs:string" />
<xs:element name="railStationCode" type="xs:string" />
<xs:element name="seqNumber" type="xs:int" nillable='true'/>
<xs:element name="shortPostalCode" type="xs:string" />
<xs:element name="possibleMatch" type="tns:extEngineAddress"/>
<xs:element name="validFlag" type="xs:boolean" default='false' />
<xs:element name="sourceAddress" type="tns:extEngineAddress"/>
<xs:element name="destinationAddress" type="tns:extEngineAddress"/>
<xs:element name="address" type="tns:extEngineAddress" />
<xs:element name="uom" type="xs:string"/>
<xs:element name="amount" type="xs:double" nillable='true'/>

<xs:element name="ExtEngineException" type="xs:string" >
<xs:annotation>
<xs:documentation>
Message contain information to be shown to user when fault occurs.
</xs:documentation>
</xs:annotation>
</xs:element>

<xs:element name="param" type="tns:keyValue">
<xs:annotation>
<xs:documentation>
<!-- TODO: Had not identified the use of this constants -->
The valid values to name attribute are
SOURCE_ADDRESS_TYPE
DEST_ADDRESS_TYPE
SAME_SOURCE_DEST_DIST

```

```

</xs:documentation>
</xs:annotation>
</xs:element>

<xs:element name="extEngineAuxInput" type="tns:keyValue" >
<xs:annotation>
<xs:documentation>
The valid values to name attribute are
SCAC,
RouteCode,
Height,
Height_UOM,
Length,
Length_UOM,
Width,
Width_UOM,
Weight,
Weight_UOM
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="extEngineAuxValueList" type="tns:keyValues">
<xs:annotation>
<xs:documentation>
The valid values to name attribute are
HAZMAT_TYPES
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="addressLines">
<xs:complexType>
<xs:sequence>
<xs:element ref='tns:addressLine' maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="extEngineAddress">
<xs:sequence>
<xs:element ref="tns:accuracy" />
<xs:element ref="tns:addressLines" />
<xs:element ref="tns:city" />
<xs:element ref="tns:countryCode" />
<xs:element ref="tns:countyCode" />
<xs:element ref="tns:latitude" />
<xs:element ref="tns:locationName" />
<xs:element ref="tns:longitude" />
<xs:element ref="tns:postalCode" />
<xs:element ref="tns:province" />
<xs:element ref="tns:provinceCode" />
<xs:element ref="tns:railSplc" />
<xs:element ref="tns:railStationCode" />
<xs:element ref="tns:seqNumber" />
<xs:element ref="tns:shortPostalCode" />
</xs:sequence>
</xs:complexType>

<xs:element name="extEngineAuxInputList">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:extEngineAuxInput" maxOccurs="unbounded"/>
<xs:element ref="tns:extEngineAuxValueList" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

```

```

<xs:element name="params" >
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:param" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="lookupServiceTimeRequest">
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:sourceAddress" />
<xs:element ref="tns:destinationAddress" />
<xs:element ref="tns:extEngineAuxInputList" />
<xs:element ref="tns:params" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="extEngineServiceTime">
<xs:annotation>
<xs:documentation>the values for UOM element should be valid unit of measure in OTM.</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element ref="tns:uom" />
<xs:element ref="tns:amount" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
</types>

<message name="lookupServiceTimeRequest">
<part name="lookupServiceTimeRequest" element="tns:lookupServiceTimeRequest" />
</message>

<message name="lookupServiceTimeResponse">
<part name="extEngineServiceTime" element="tns:extEngineServiceTime" />
</message>

<message name="ExtEngineException">
<part name="fault" element="tns:ExtEngineException" />
</message>

<portType name="ExternalServiceTimeEngine">
<operation name="lookupServiceTime" >
<input message="tns:lookupServiceTimeRequest" />
<output message="tns:lookupServiceTimeResponse" />
<fault message="tns:ExtEngineException" name="ExtEngineException" />
</operation>
</portType>

<binding name="ExternalServiceTimeEnginePortBinding" type="tns:ExternalServiceTimeEngine">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="lookupServiceTime">
<soap:operation soapAction="lookupServiceTime" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
<fault name="ExtEngineException">
<soap:fault name="ExtEngineException" use="literal" />
</fault>

```

```
</operation>

</binding>

<service name="ServiceTimeEngineService">
  <port name="ExternalServiceTimeEnginePort" binding="tns:ExternalServiceTimeEnginePortBinding">
    <soap:address location="http://localhost/GC3Services/ServiceTimeEngineService"/>
  </port>
</service>

</definitions>
```

