Oracle Fusion Cloud Transportation and Global Trade Management

Integration Guide

Release 26A

Oracle Fusion Cloud Transportation and Global Trade Management
Integration Guide

Release 26A

G36022-01

Author: Oracle Transportation and Global Trade Management Product Team

Contents

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.

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.

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 Overview

Integration Overview

This manual is for members of the Transportation and Global Trade Management Cloud implementation teams, who are responsible for connecting the system to other external systems through integration interfaces. This manual explains how to send and receive integration messages and the format of each message.

Oracle Transportation and Global Trade Management Cloud exposes a large number of interfaces to support integration with external applications. There are two distinct use cases for application integration covered in this guide: Business Process Integration and User Interface Integration. Data Tier Integration is covered in the *Data Management Guide*.

Business Process Integration is defined in this guide as the sending and receiving of functional messages between applications. These messages are implemented as **Transmission** or **Mobile Device Communication** XML documents.

The Transmission XML document is the primary application integration message and is covered in detail in this guide. Mobile Device Communication integration is built on the infrastructure described in the main chapters of this guide but uses a different XML message format and is optimized for high volume, small messages e.g. for Fleet power or trailer devices. The details for this integration are covered in the *Mobile Device Communications chapter*.

As an example functional message, the Transportation Orders (`TransOrder`) Transmission interface receives transportation order messages from an external application into Transportation and Global Trade Management Cloud. This is referred to as an **inbound** interface. The **outbound** interfaces send messages from the Transportation and Global Trade Management Cloud application to other external applications. For example, shipments planned from Orders in Transportation Management Cloud can be sent in a `PlannedShipment` message to another system for additional processing.

The valid formats of all messages are described by Integration XML Schema Definition (XSD) documents.

This guide describes the following:

- The purpose of each XSD and the available messages defined by it.
- The application level message protocol describing the process of message exchange.
- The transport level protocols supported for inbound and outbound message communication.

User Interface Integration is defined in this guide as the retrieval and modification of application data to support customer-defined user interfaces. This is achieved by the use of the Transportation and Global Trade Management Cloud Representational State Transfer (ReST) API, which is covered in the *REST API Guide*.

3 Understanding the Integration Schemas

Definitions

This section will describe the overall format and design of the Integration XML Schema Definition (XSD) schemas for use when implementing interfaces.

The following definitions are used widely in the remainder of the document and deserve detailed explanation of their exact meaning.

- **Global element/type/attribute**: This is an XSD definition which is an immediate child of the <xsd:schema> document element.
- **Local element/type/attribute**: This is an XSD definition that is contained within a parent definition e.g. an element definition within a named complex type.
- **Local reference**: This refers to using the 'ref=' attribute to reference a global element definition i.e. as opposed to a Local element definition.
- **Primary Document**: These are the global element definitions for XML documents which can be sent as messages to and from Transportation and Global Trade Management Cloud. The main example is the `Transmission`.
- **Transaction interface**: These are the definitions which relate to the high level business object interface e.g. for Orders, Shipments, Invoices etc. The `Transmission` primary document will contain one or more Transaction interface elements.

Integration XML Schemas

The Transportation and Global Trade Management Cloud Integration XML schemas define the data elements that the system sends or receives for each type of interface. The XML schema definitions are considered the true definition for the interfaces, and this Integration Guide covers concepts which apply to all schemas. Information appearing in the schemas will be specific to each interface and will take priority over the generic information in this guide.

These are the sets of XML schema documents:

- Transmission related schemas
- Mobile Device related schemas
- Workflow Web Service related schemas

The Mobile Device Communication Message Schema defines the format in which you send or receive mobile device messages and are covered in the *Mobile Device Communications chapter*.

The Workflow Web Service related schemas are used by a set of Web Services which expose Agent workflow triggers. See the *Workflow Web Service chapter* for details.

W3C XML Schema Version

All schema files conform to the *W3C XML Schema* standard defined by the following namespace name:

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

Viewing the XML Schemas

The schema files can be obtained from the following User Interface menu location:

Business Process Automation > Integration > Integration Manager > Retrieve Schemas

When integrating to Transportation and Global Trade Management Cloud using XML, you must create documents that follow the structure and rules of the Transportation and Global Trade Management Cloud XML schemas. We recommend that you use an XML management tool to view the schema files. This will help in understanding the Transportation and Global Trade Management Cloud data elements and relationships.

> **Note:** The OTMGTM.xsd schema definition consolidates the separate schema files into one root document. This provides a more flexible format for the validation or processing of messages by third party tools.

The W3C XML Schema site (*W3C XML Schema*) provides links to several such tools. The examples in this document use the freely available Oracle XSD Visual Editor which is a built-in part of the JDeveloper IDE (see *Jdev overview on the Oracle website* for details.)

For a full description of the Visual Editor tool please see *Oracle® Fusion Middleware Developing Applications with Oracle JDeveloper*.

Transmission Schema

Schema Files

The following schema files contain all definitions for the Transmission schema:

- **Transmission.xsd**: Definitions for all Primary Documents.
- **TransmissionCommon.xsd**: Definitions for types shared across interfaces.
- **Transaction.xsd**: This is a convenient schema which imports all other Transaction definition schemas.
- **OTMGTM.xsd**: This file can be used to include/import all schema definitions contained in all other schema files.
- **GTM.xsd**: Transaction interfaces for Global Trade Management Cloud application functionality.
- **LocationContact.xsd**: Transaction interfaces for Location and Contact business objects and other related common types used in other interfaces e.g. Involved Party.
- **ShipUnit.xsd**: Transaction interfaces for Ship Units common to Orders, Shipments etc.
- **Item.xsd**: Transaction interfaces for Item and Package related business objects.
- **Document.xsd**: Transaction interfaces for Document Content for Content Management Systems.
- **Configuration.xsd**: Low level transaction interfaces e.g. User management.
- **Shipment.xsd**: Transaction interfaces for Shipment related business objects including Planned, Actual and Tendered shipments.

- **Order.xsd**: Transaction interfaces for Purchase Orders and Order Release business objects.
- **Planning.xsd**: Transaction interfaces related to planning of Orders onto Shipments.
- **Finance.xsd**: Transaction interfaces related Invoices, Payments etc.
- **Rate.xsd**: Transaction interfaces related to Rating structures used for cost calculations.
- **Job.xsd**: Transaction interfaces for Brokerage and Forwarding.
- **GenericTransaction.xsd**: Miscellaneous transaction interfaces that can apply to many business object types e.g. for object status updates.

XML Namespaces

The following table lists the namespace URLs and associated physical file name for each schema:

XML Namespaces

XSD Schema File	Namespace
Transmission.xsd and all other non-Global Trade Management Cloud schemas (listed in the Transmission Schema table).	`http://xmlns.oracle.com/apps/otm/transmission/v6.4`
GTM.xsd	`http://xmlns.oracle.com/apps/gtm/transmission/v6.4`

Inbound Transmission XML documents that do not declare a namespace association are supported but they are assumed to be based on the XML schema definitions prior to version 6.4.2. Therefore, these documents may incur a runtime performance penalty due to the transformation required to convert the XML to the current namespace. If, however, the XML conforms to the current namespace definitions but does not declare a namespace, then adding a Version attribute to the root Transmission element will avoid the transformation overhead. For example:

```
<Transmission Version=”CURRENT”>
```

...

> **Note:** The value of the Version attribute must be either CURRENT or PRE642 where the latter is the default value if no attribute is specified, i.e. the XML is assumed to require transformation.

Primary XML Documents

There are primary XML documents defined in the Transmission schema that are used inbound and outbound for both Transportation and Global Trade Management Cloud interfaces:

- **Transmission**: The Transmission is the primary document used for messages inbound to and outbound from the system. Each Transmission can contain multiple transactions to be processed. A unique Transmission Number is assigned to each inbound and outbound message.
- **TransmissionAck**: The TransmissionAck is the response message to the receipt of the Transmission. It contains the confirmation for the receipt of the Transmission with the unique assigned Transmission Number, or an error if the Transmission could not be persisted.

- **TransmissionReport**: The TransmissionReport summarizes the errors that were detected during the processing of the Transmission. The report is optionally sent after all the transactions in the Transmission have been completed (successfully processed or with generated errors). The requirement to receive a TransmissionReport is indicated in the inbound Transmission Header.

Transmission Processing Styles

There are different styles for processing the contents of a Transmission:

- **Transactional style**: where the Transactions within the Transmission are processed as messages used to create, modify or delete transactional content or to trigger processing for transactional content.
- **Query style**: where the Transactions within the Transmission are request/reply transactions to retrieve content or return the results of processing against transactional content.

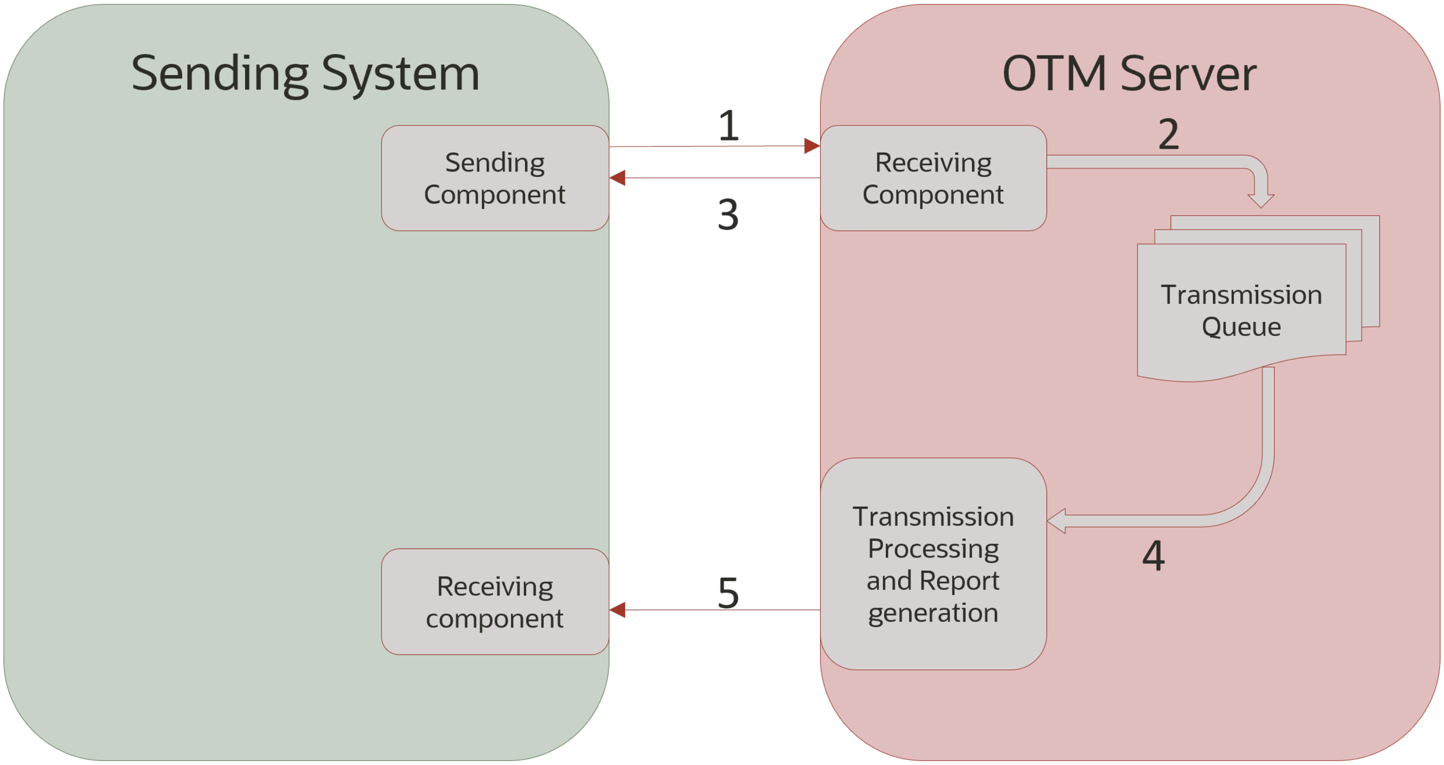
The style used is based on a combination of the value of the **Transmission Type** Transmission header property and the interfaces present in the Transmission body content. If no Transmission Type value is specified the style defaults to Transactional.

It is not supported to mix Transactional style content with Query style content in the same Transmission. It is also not valid to request Query style processing for interfaces which do not support that style.

Transactional Style Interfaces

Transactional Interfaces are 'submitted' to Transportation and Global Trade Management Cloud for subsequent processing similar to asynchronous processing. (Strictly speaking the process is a synchronous transaction that accepts the message for later processing followed by the asynchronous processing of message. The synchronous part returns the reference which can be used to track the results of processing.) This allows the Transportation and Global Trade Management Cloud Application server to manage workload and ensure changes to persistent object data are handled within a transactional context.

The following diagram shows how the documents are used when processing an inbound transactional Transmission XML into Transportation and Global Trade Management Cloud.

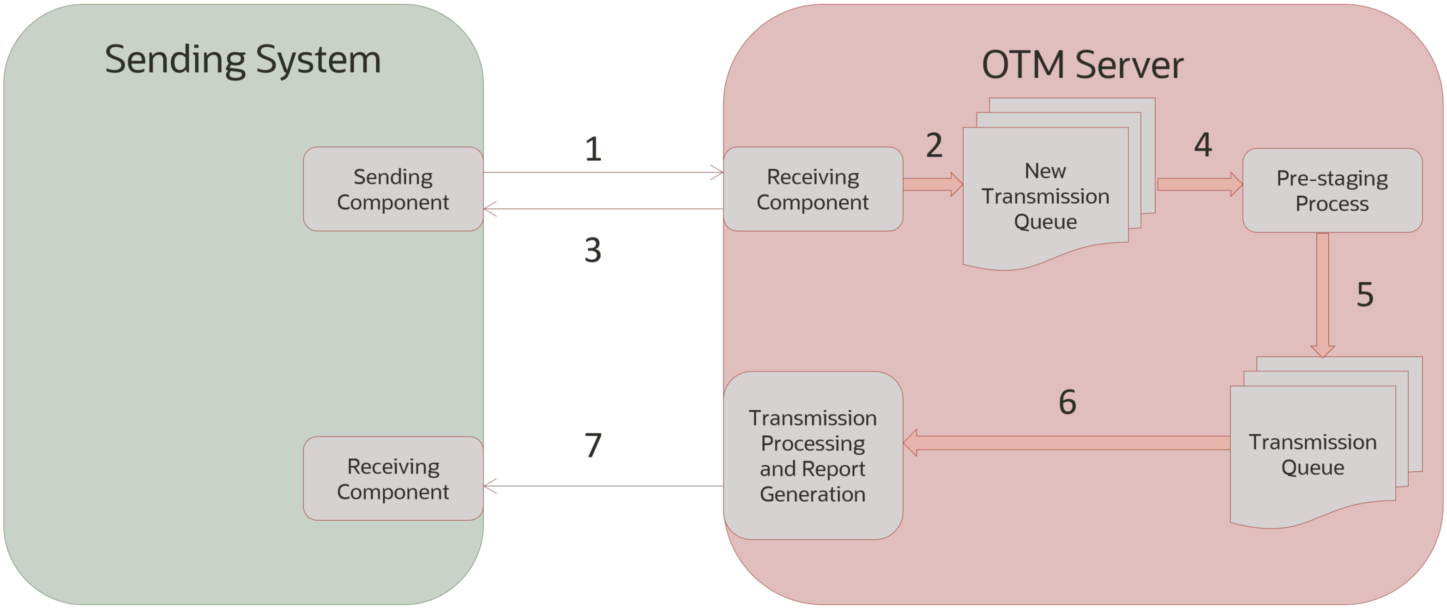


1. Transmission XML document is sent to Transportation and Global Trade Management Cloud. The nature of the sending component depends on the transport protocol used e.g. it will be an HTTP client if sending the message via HTTP POST.
2. Transmission message is persisted for later processing generating a unique Transmission number in the process.
3. A TransmissionAck document is returned which will contain the Transmission number.
4. The background Transmission processing workflow retrieves the Transmission at some time later and processes it. For example, it creates or updates the business object using the original XML message content.
5. Optionally, a TransmissionReport is sent to confirm the status (PROCESSED or ERROR) of the Transmission message. The report message contains the Transmission number for correlation. The sending of the report is configurable i.e. 'always', 'never' or 'on error only'. The nature of the receiving component depends on transport method e.g. it can be a Servlet URL or Web Service endpoint and the transport method can be different to original inbound transport method.

Large Transmission XML Processing Style

In step 2 above, the persistence of the Transmission involves the creation of a record in the transmission table and a record in the transaction table for each Transaction in the message. This is referred to as "staging". The TransmissionAck response is only sent once all these records are committed. If the inbound Transmission contains a large number of Transactions, then the time taken to respond with a TransmissionAck may exceed the external client's configured time out value.

There is an alternative process, referred to as "pre-staging", where the TransmissionAck is returned after the record is committed to the transmission table but before each transaction record is created. This process should not exceed any reasonable external client time out. The behaviour is property controlled and is described below.



1. The transmission XML document is sent to Transportation and Global Trade Management Cloud. The nature of the sending component depends on the transport protocol used, e.g. it will be an HTTP client if it is sending the message via HTTP POST.
2. The full transmission message is saved in the I_TRANSMISSION table with the status of "NEW" for later processing and generating a unique transmission number in the process.
3. A TransmissionAck document is returned which contains the transmission number.
4. The background transmission processing workflow retrieves the transmission for further processing by the pre-staging process. In this process, all of the transactions in the transmission XML are stored in the I_TRANSACTION table.
5. After completing the pre-staging process, the transmission now continues further processing identical to that performed after normal staging processing.
6. The background transmission processing workflow retrieves the transmission and processes it. For example, it creates or updates the business object using the original XML message content.
7. Optionally, a TransmissionReport can be sent to confirm the status (PROCESSED or ERROR) of the transmission message. The report message contains the transmission number for correlation. The conditions that trigger the sending of the report are configurable, i.e. "always," ""never," or "on error only". The nature of the receiving component depends on transport method, e.g. it can be a servlet URL or web service endpoint, and the transport method can be different from the original inbound transport method.

Transmission Status (NEW): Indicates that the transmission is in a pre-staging state. This means that the transmission is stored in the I_TRANSMISSION table and queued for further processing. In its next step, it will process all the transactions in the transmission table and store the data in the I_TRANSACTION table.

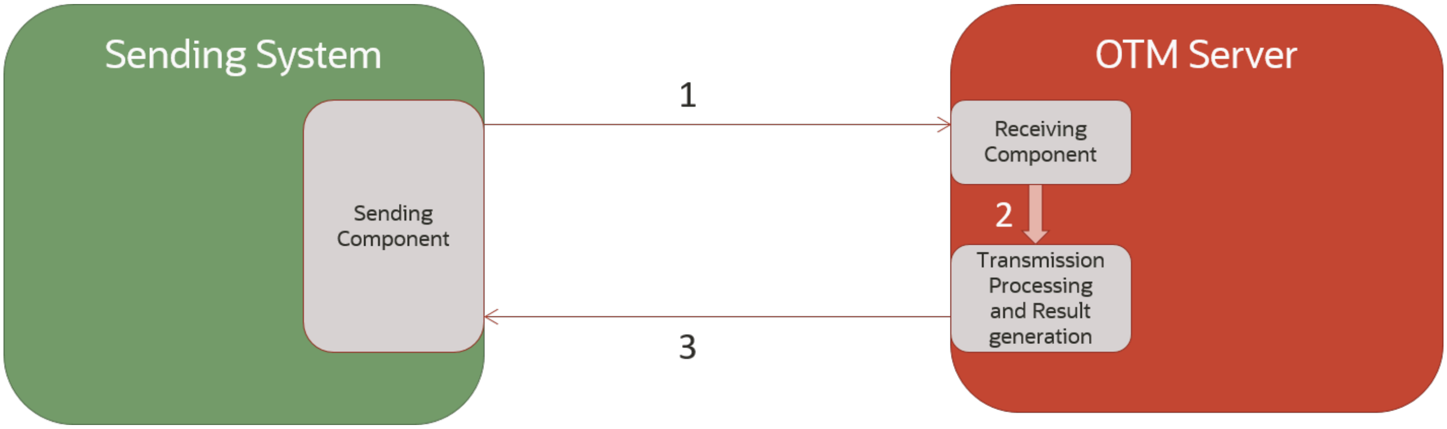
This functionality is enabled by setting the following property:

```
glog.integration.transmission.enableInboundPreStaging=true.
```

Query Style Interfaces

Query style interfaces are used when synchronous processing is required for example, for a rating request.

The following diagram shows the documents used when processing an inbound query Transmission.



1. Transmission XML is sent to Transportation and Global Trade Management Cloud. The nature of the sending component depends on the transport protocol used e.g. it will be an HTTP client if sending message via HTTP POST.
2. The internal processing is called. The Transmission Type property must be present and be either "QUERY" or "SERVICE" depending on the interface to be processed.

   > **Note:** If the Transmission Type isn't specified for "QUERY" interfaces such as RemoteQuery, then they will be processed asynchronously.

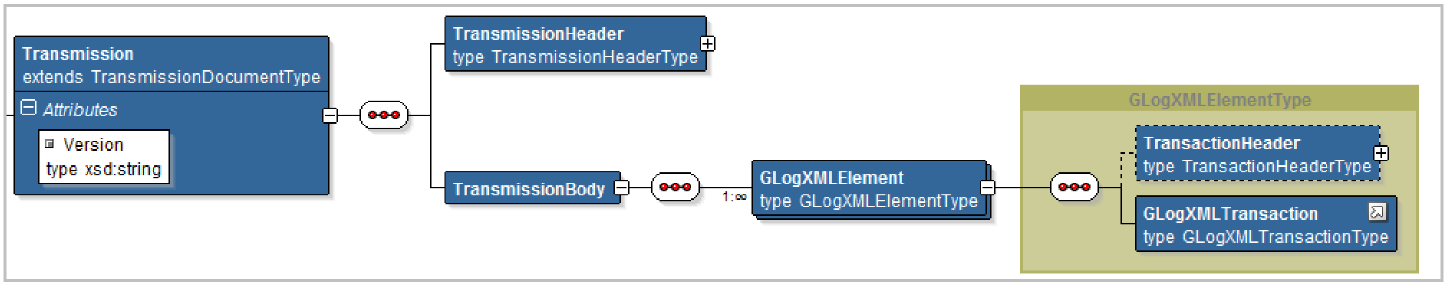
3. The processing results are returned as either a TransmissionAck or a new Transmission document. The document used can depend on:
   1. The protocol used. For example, Web Service calls must always return the document declared in the WSDL output parameter.
   2. Customer-defined property settings (if not using a Web Service). For example, when sending multiple RIQ requests via HTTP, the following property will result in a Transmission being returned:

      ```
      glog.integration.remoteQuery.WrapReplyInTransmission=1
      ```

Transmission Structure

The Transmission XML document has a root element with local name Transmission. The current namespace URL is `http://xmlns.oracle.com/apps/otm/transmission/v6.4`.

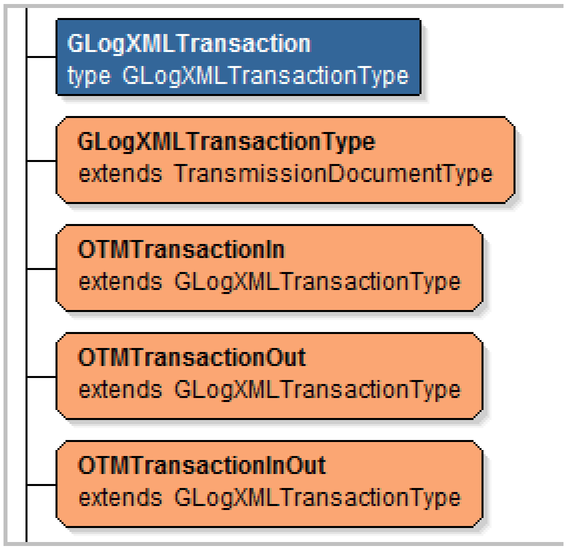
The Transmission contains a `TransmissionHeader` and a `TransmissionBody`. The `TransmissionBody` contains one or more interface transactions, each wrapped in a `GLogXMLElement` element. Each `GLogXMLElement` can contain an optional `TransactionHeader` element but must contain an **interface element** which must be a type of `GLogXMLTransaction` element. See figure below for schema diagram representation.



The Transmission schema uses **Substitution Groups** to define the possible interface elements that can be contained by a `GLogXMLElement`. This removes the need for the Transmission schema to list all possible interface elements. This approach allows external clients to only import the necessary schemas for implemented interfaces.

> **Note:** Some application integration tools may not fully support schema definitions which use substitution groups. The TransmissionCommon.xsd file comes with the commented out equivalent definitions and so can be edited to support such tools. Use the following process:
>
> 1. Obtain the set of XSD schema files for the relevant product version.
> 2. Open the TransmissionCommon.xsd file in a suitable schema editor, e.g. JDeveloper.
> 3. Follow the instructions in the documentation for the GLogXMLElementType definition to edit and save the file for use by the external tool.

The `GLogXMLTransaction` is essentially a marker element which specifies the base type – `GLogXMLTransactionType` - of the elements that can be used in its place i.e. substituted for it. Only elements whose type extends the `GLogXMLTransactionType` can be substituted for the `GLogXMLTransaction` element. At present there are three types which extend `GLogXMLTransactionType` to further qualify the interface and provide additional control for how the interface can be used. See the figure below for a schema diagram of the three types and the subsequent explanation.


GLogXMLTransaction
type GLogXMLTransactionType
GLogXMLTransactionType
extends TransmissionDocumentType
OTMTransactionIn
extends GLogXMLTransactionType
OTMTransactionOut
extends GLogXMLTransactionType
OTMTransactionInOut
extends GLogXMLTransactionType

- `OTMTransactionIn`: Interfaces which are only valid for inbound processing must extend this type

- `OTMTransactionOut`: Interfaces which are only valid for outbound processing must extend this type
- `OTMTransactionInOut`: Interfaces which are valid for both inbound and outbound processing must extend this type

For example, the `TransOrder` interface is valid inbound and outbound and so has the following schema definition (some information has been removed to aid clarity for this discussion):

```
<xsd:element name="TransOrder" type="TransOrderType" substitutionGroup="GLogXMLTransaction">
...
 <xsd:complexType name="TransOrderType">
...
 <xsd:complexContent>
 <xsd:extension base="OTMTransactionInOut">
 <xsd:sequence>
 <xsd:element name="TransOrderHeader" type="TransOrderHeaderType">
...
```

The `TransOrder` element definition declares that the element is a `TransOrderType` and that the element can be substituted for the `GLogXMLTransaction` element. The `TransOrderType` extends the `OTMTransactionInOut` type which ensures that it is a valid substitution (because it in turn extends the `GLogXMLTransactionType`). The `OTMTransactionInOut` type declares that is valid for both inbound and outbound Transmissions.

The following is an example Transmission XML document containing two transaction interfaces (`TransOrder` followed by `ShipmentStatus`).

```
<?xml version="1.0" encoding="UTF-8"?>
<Transmission xmlns="http://xmlns.oracle.com/apps/otm/transmission/v6.4">
 <TransmissionHeader>
 <SenderTransmissionNo>987654321</SenderTransmissionNo>
 <AckSpec>
 <ComMethodGid>
 <Gid>
 <Xid>HTTPPOST</Xid>
 </Gid>
 </ComMethodGid>
 <ContactGid>
 <Gid>
 <DomainName>GUEST</DomainName>
 <Xid>MY_CONTACT_ID</Xid>
 </Gid>
 </ContactGid>
 </AckSpec>
 </TransmissionHeader>
 <TransmissionBody>
 <GLogXMLElement>
 <TransOrder>
 ...etc...
 </TransOrder>
 </GLogXMLElement>
 <GLogXMLElement>
 <ShipmentStatus>
 ...etc...
 </ShipmentStatus>
 </GLogXMLElement>
 </TransmissionBody>
</Transmission>
```

AckSpec

The AckSpec element controls whether or not a TransmissionReport message is sent when the transactions within a Transmission have been completely processed i.e. when they have been persisted and any dependent workflow has been completed.

Note: AckSpec is only valid for Transactional Style interfaces.

The AckOption child element specifies when to receive a TransmissionReport. If unspecified, the behavior is property controlled where the default property setting (`glog.integration.TransmissionReport="on error"`) is equivalent to the `ERROR` option.

- ERROR: Send Transmission Report only when there are errors.
- YES: Send Transmission Report in all cases.
- NO: Do not send Transmission Report, even if there are errors.

The `ComMethodGid` element identifies the transport method to be used to send the report. The `ContactGid` element identifies a contact record with an associated External System containing the relevant details – user credentials, email address, servlet URL etc. – for the selected method.

Common Schema Elements and Types

The following sections describe, at a high level, a few of the most common data elements that are likely to be present in many interfaces. More detailed information is available in the *Inbound Transmission Processing chapter* and the *Outbound Transmission Processing chapter*.

Global IDs (GIDs)

GIDs are global identifiers that Transportation and Global Trade Management Cloud uses to define a primary key for various types of information (e.g., orders, shipments, locations, payment vouchers, etc.). A GID consists of the following two parts:

- **Domain name**: (optional). The domain name typically identifies a company or business unit and is used to separate data and secure it from other data in a shared, web-based environment. For example, if you are using Transportation and Global Trade Management Cloud in an environment where many companies may be using the same Transportation and Global Trade Management Cloud installation, the domain allows you to isolate data in Transportation and Global Trade Management Cloud for each company. Therefore, many users from different companies can work in the same Transportation and Global Trade Management Cloud installation (or web site) and use data that is private and specific to their company. If you do not include a domain name in a GID, it can be viewed across all domains in your system.
- **External ID (XID)**: (optional). The ID that defines the item on the external system. An external system is any system other than Transportation and Global Trade Management Cloud. If no value is specified, the GID is assumed to be completely null.

It is recommended to only use uppercase characters in GIDs and XIDs in your transmissions. The Transportation and Global Trade Management Cloud user interface normally only search for uppercase text strings and therefore it may be difficult to find records with mixed case XIDs. The following characters SHOULD NOT be present in XID strings. Failure to follow these rules may result in inconsistencies when referring to the business objects concerned:

- \
- ,
- x
- .
- <

- >
- =
- /
- &
- ?
- ;
- #
- $
- '
- "
- (
-)

> **Note:** You should also be careful not to create GIDs with trailing spaces, as these records will not be retrievable via the User Interface.

Transaction Code

Transaction Codes tell Transportation and Global Trade Management Cloud what to do with the transactions it receives from other systems. Transaction Code elements will have the type `TransactionCodeType` which is an enumerated type and should contain one of the following values:

- **I**: Insert. Use this transaction code to send new information to Transportation and Global Trade Management Cloud. Transportation and Global Trade Management Cloud creates a new record. If the record already exists, then the transaction will generate a "record already exists" error.
- **II**: Insert and Ignore. When used, if the record already exists, then it is not updated and the "record already exists" error message is not logged. If it does not exist, then it is inserted.
- **U**: Update. Transportation and Global Trade Management Cloud updates an existing record.
- **UU**: Transportation and Global Trade Management Cloud updates the existing record while suppressing "no data found" constraint violations.
- **IU**: Insert and Update. Transportation and Global Trade Management Cloud creates a new record unless it already exists, in which case Transportation and Global Trade Management Cloud updates the existing record with the new information.
- **UI**: Update and Insert. This works the same way as IU.
- **D**: Delete. Use this transaction code to delete an existing record.
- **DD**: Transportation and Global Trade Management Cloud deletes the existing record while suppressing "no data found" constraint violations.
- **NP**: No Persist. Use this transaction code to keep Transportation and Global Trade Management Cloud from persisting data to the database. For example, enter NP if you do not want to persist public locations. This is the default TransactionCode when the data is sent outbound from Transportation and Global Trade Management Cloud.
- **RC**: Replace Children. Use this transaction code to delete all child data corresponding to the top level parent, update the top level parent, and insert the new child data. If a child is added with the same primary key as one being replaced, then all the values in the existing record are replaced rather than deleting the record and

inserting a new one. You use the ReplaceChildren element to specify what child elements Transportation and Global Trade Management Cloud should replace. The remaining elements are processed using the IU transaction code.

- **RP**: Replace Primary/Parent. Use this transaction code to replace the primary/parent object without replacing the child objects. This will remove all fields in the primary/parent object that are not contained in the incoming xml, and will perform an insert/update on all the child data.
- **R**: Replace. Use this transaction code to replace the primary/parent and child objects. This is a combination of the RC and RP transaction codes.

Replace Children

Replace Children content is for inbound use only and is defined in elements whose type is ReplaceChildrenType. It is used in conjunction with a TransactionCode value of "RC". By specifying the ManagedChild content elements, it is possible to limit which child objects are replaced by new values within the present Transmission. See the *Extended Inbound Validation* section for more details.

Send Reason

Send Reason information is outbound only and is defined in elements whose type is `SendReasonType`. The information is used to show the internal application event which resulted in the generation of the outbound Transmission, for example, a SHIPMENT CREATED event.

Integration Saved Query

Integration Saved Queries are used in inbound transmissions and are defined by elements whose type is `IntSavedQueryType`. It allows an interface to support a method of identifying an object to be modified when the objects primary key is not known. An Integration Saved Query essentially represents an SQL query that must already have been created in the user interface. The element content will then contain the GID of this query and any parameters required by the query as arguments.

See *Using Integration Saved Queries* for more details.

Flex Fields

Flex fields are so called because they provide a flexible set of table columns for Transportation and Global Trade Management Cloud implementers to configure their data model to suit their functional needs. They can be entered via the User Interface and/or via Integration.

Flex fields cover a number of data types and have been added to most of the major business objects and their corresponding interfaces and are available for both inbound and outbound messages.

The valid schema types are described in the following table.

Valid Schema Types

Schema Type	DB Data type	(Maximum) Number	Description
FlexFieldStringType	VARCHAR2(150)	20	Holds string data
FlexFieldNumberType	NUMBER()	20	Holds numbers with any precision
FlexFieldCurrencyType	OTM Currency UOM	10	Holds financial amount
FlexFieldDateType	DATE	20	Holds date/time values

> **Note:** Not all flex field types have been added to all interfaces and not all interfaces can support the maximum number of fields. Please consult the relevant XML schema documentation to determine the support available for each business object and the associated database schema documentation to determine if underlying table supports the maximum number.

In addition to the above, the RATE_GEO and RATE_OFFERING outbound interfaces have equivalent Flex Fields available which are defined slightly differently due to the database centric nature of these (outbound only) interfaces. These interfaces use the following elements:

RATE_GEO and RATE_OFFERING Outbound interface Elements

Schema Element	DB Data type	Number	Description
ATTRIBUTE1…n	VARCHAR2(150)	20	Holds string data
ATTRIBUTE_NUMBER1…n	NUMBER()	10	Holds numbers with any precision
ATTRIBUTE_DATE1…n	DATE	10	Holds date/time values

Date & Time

The GLogDateTimeType schema type is a platform neutral representation of the date and time to the accuracy of seconds. It is a string in the format YYYYMMDDHHMMSS where:

- **YYYY**: The year e.g. 2012
- **MM**: The month e.g. 01 is January, 12 is December
- **DD**: The day number e.g. 01 to 31
- **HHMMSS**: The time in 24 hour clock e.g. 231530

An example value is "20121031083030" which represents October 31st, 2012 at 30 seconds after 8.30a.m.

Additionally, the GLogDateTimeType can hold a Time Zone ID element (TZId) and a Time Zone Offset. When date elements are inbound the Time Zone ID is used to calculate times relative to the system time. On outbound messages, the Time Zone Offset will be populated to show the offset in hours from the system Time Zone compared to the target system locale. See the *Time Zone Override* section for details.

Strict Schema Validation

Strict schema validation is applied to all incoming XML messages.

Any validation error will be reported immediately as part of the synchronous TransmissionAck response. The details of the error – i.e. which element is thought to be in error – is reported in the EXCEPTION log.

Note: If an XML format causing an error was known to be accepted before strict validation was enabled then the most common explanations are:

- elements are in an incorrect sequence.
- an invalid element name is used (e.g. incorrect cased letters).
- Mandatory elements are either missing or are specified but empty.
- Whitespace is present and is considered "empty".

The schema for the associated interface should be reviewed to verify and correct the XML message.

4 Supported Transport Protocols

Available Protocols

There are a number of transport protocol options available when sending messages to or from Transportation and Global Trade Management Cloud and Global Trade Management. The available options can vary for inbound versus outbound and whether the message is a Transmission XML or a Mobile Device Message XML.

The following table shows all valid combination (Y – available, N – not available):

Supported Transport Protocols

Transport Protocol	Inbound	Outbound	Supports Transmission	Supports Message
SOAP Web Services	Y	Y	Y	Y
HTTP POST	Y	Y	Y	Y

The protocol specific details e.g. resource URLs are covered in each of the following sections:

- *Security*
- *HTTP Integration*

> **Note:** In addition to the protocols listed above it is possible to upload XML files using the Integration Manager User Interface. This is a useful tool when testing new interfaces or diagnosing functional issues that are normally triggered via integration. See the *Integration Manager help* for usage instructions.

5 Security

OAuth 2

OAuth 2 Client Credentials flow is available as the authentication protocol for inbound REST API and XML integration using HTTP POST. The authorization service is managed by Oracle Identity Cloud Service (IDCS) (also used for single sign-on application user accounts).

The following steps must be followed to configure an external integration application to use OAuth 2 for inbound integration.

1. Configure external integration application in IDCS to obtain client ID and secret
2. Assign client ID generated in step 1 above as Nickname for Transportation and Global Trade Management Cloud integration user account, i.e. user account authorized to process inbound integration messages for all required application resources.

IDCS Configuration

> **Note:** The following instructions assume that the user has already obtained a valid account and URL for access to the IDCS Administration console.

The following steps are taken from the *online documentation for Oracle Identity Cloud Service*. Refer to the online documentation for up-to-date steps.

1. Log in to IDCS administration console using the credentials provided.
2. Under **Identity domain** click **Applications**.
3. Click **Add application**.
4. Select **Confidential Application** and click **Launch workflow**.
5. Provide a Name and click **Next**.
6. **Resource Server Configuration**
   a. Select "**Configure this application as a resource server now**" .
   b. Provide a value for **Access token expiration**.
   c. Provide a value for the Primary Audience, e.g. `https://{host-id}/` where `{host-id}` is the Transportation and Global Trade Management Cloud server. The actual value may depend on whether the target application is REST API - `https://{host-id}/logisticsRestApi/` - or XML integration - `https://{host-id}/logisticsXmlApi/`.
   d. Select **Add scopes** and click **Add**.
   e. Enter a unique name for the **Scope**. This can be any text, this scope name will be selected or used at later steps.
7. **Client Configuration**
   a. Select **Configure this application as a client now** and **Client Credentials** as the Allowed Grant Types.
8. Select **Next**.
9. Under **Web tier policy** select **Skip and do later**.
10. Click **Finish**.
11. Click **Edit OAuth configuration**.
12. Under **Client configuration** > **Token Issuance policy** select “**Add resources**".

13. Click **Add scope**, select the drop-down against the Confidential application we are created and select the available scope under this application and click **Add**.
14. Click **Save Changes**.
15. Click **Activate** and **Activate application**.
16. Make a secure note of the Client ID and Client Secret. These will be used by the external application to authenticate with IDCS when using the target application and when configuring the application user Nickname.

Transportation and Global Trade Management Cloud Configuration

1. Obtain the Client ID associated with target application previously configured in IDCS.
2. Edit the integration user account via the User Manager page.
3. Place the Client ID in the **Nickname** field.
4. **Save**.

Runtime Authentication

The OAuth 2 Client Credentials protocol works by completing the following stages, which MUST be performed using HTTPS:

1. Client submits Authorization request to obtain a token.
2. Client requests resource and pass token.

The first step requires the URL for the IDCS instance:

```
https://<IDCS-Service-Instance>.identity.oraclecloud.com/oauth2/v1/token
```

The HTTP request will use HTTP Authentication (Basic) header to pass the Client ID and Client Secret. The message body will use the "`x-www-form-url-encoded`" content type to send the following name/value parameters in the message body:

- `grant_type=client_credentials`
- `scope=[Copy the scope from the Confidential Application Client configuration > Token Issuance policy > Resources > Scope.]`

A successfully authenticated request will receive the HTTP 200 status code and a JSON message body containing the required "Access Token". This access token must then be passed as the HTTP "Authorization: Bearer" header.

6 Available Interfaces

Schema Files

This chapter lists each schema file and a description of the interfaces defined within it. The schema file will contain the detailed definition and documentation for each interface.

Shipment.xsd

Shipment.xsd Interfaces

Interface	Description	Inbound	Outbound
ActualShipment	Third parties send actual shipments to Transportation and Global Trade Management Cloud, which define the working or final form of a shipment.	Y	N
PlannedShipment	Transportation and Global Trade Management Cloud builds planned shipments and can send them to involved parties, to service providers as part of the tender process or to a warehouse management system (WMS).	N	Y
ShipmentStatus	Service providers and other third parties transmit shipment event information to Transportation and Global Trade Management Cloud with the ShipmentStatus interface. Shipment status information describes activity on shipments and shipment groups. For example, you can use shipment statuses to determine whether a shipment is running on time or whether it is late. In addition, you can send vessel and flight information, and equipment service provider and order information. Based on shipment status information it receives, Transportation and Global Trade Management Cloud can re-plan a shipment.	Y	Y
SShipUnit	Queries for and updates a shipment ship unit without direct knowledge of the owning shipment.	Y	N
ShipmentLink	Identifies related shipments at a consolidation or de-consolidation	Y	N

Interface	Description	Inbound	Outbound
	pool for both inbound and outbound interfaces.		
ShipStop	Used to modify specific stop related information for a shipment as an alternative to sending the whole shipment.	Y	N
Equipment	Defines the equipment that can be assigned to a shipment.	Y	Y
ShipmentGroup	Used to define a group of shipments.	Y	Y
TenderOffer	Once a shipment has been planned, Transportation and Global Trade Management Cloud sends a tender offer to a service provider. The tender offer provides a contract for the service provider to carry a particular shipment.	N	Y
TenderResponse	Service provider response to accept or decline a previous TenderOffer interface message.	Y	N
ShipmentGroupTenderOffer	Used to send a tender offer for a group of shipments to a service provider.	N	Y
BookingLineAmendment	Used to send booking line changes out of the system.	N	Y
Voyage	Used to send world-wide vessel schedule information to external systems. Voyage schedules define the rotation of a carrier's vessel as the vessel goes from a set of loading points (departure ports) to a set of unloading ports (arrival ports).	N	Y
DemurrageTransaction	Used to communicate the fees charged (demurrage) by a carrier to a customer for the use of assets beyond a contracted free time for loading and unloading.	N	Y
CharterVoyage	Represents an ocean transport movement by a carrier from a loading port to a discharge port.	Y	Y
Consol	Used to specify the shipment consolidator. A consol can be created for a charter voyage or air schedule (flight).	Y	Y
Driver	Sends and receives driver data to Transportation and Global Trade Management Cloud which includes basic personal information, CDL information, driver types, driver team data, status, remarks, etc.	Y	Y

Interface	Description	Inbound	Outbound
PowerUnit	Sends and receives power unit (such as a tractor) data to Transportation and Global Trade Management Cloud. This information includes power unit name, power unit type, special services, remarks, etc.	Y	Y
Device	Used to send device logon and logoff messages to Transportation and Global Trade Management Cloud.		
DriverCalendarEvent	Specifies a driver calendar event, which describes calendar event information on a driver.	Y	N

Order.xsd

Order.xsd Interfaces

Interface	Description	Inbound	Outbound
TransOrder	Transportation and Global Trade Management Cloud receives transportation orders from external systems. These orders can include basic information such as IDs, pick-up and delivery dates, service providers, and details such as ship units or line items. It represents orders which can be shipped as one or more 'released' amounts called an Order Release.	Y	Y
TransOrderLink	Used to establish a link between order base objects. Provides the ability to maintain orders in various states along with their relationships.	Y	N
Release	Transmits Order Release information to and from Transportation and Global Trade Management Cloud. An order release represents the shippable portion of a transportation order.	Y	Y
TransOrderStatus	Sets order base status events.	Y	N
ReleaseInstruction	Allows you to specify line items and ship units, and release specific quantities of them for a particular order base.	Y	N

Interface	Description	Inbound	Outbound
Quote	The Quote Interface allows customer service representatives to supply their customers with transportation quotes.	N	Y
OBShipUnit	Contains information on ship units in an order base.	N	Y
OBLine	Contains information on lines in an order base.	N	Y
OrderMovement	Represents part of the routing for an order. It is also a collection of ship units that are unplanned for certain part of a route.	Y	Y

GTM.xsd

GTM.xsd Interfaces

Interface	Description	Inbound	Outbound
GtmContact	Represents a Global Trade Management party.	Y	Y
GtmTransaction	Represents a Global Trade Management trade transaction for the screening of shipment and/or order release information.	Y	Y
GtmTransactionLine	Represents a Global Trade Management trade transaction line.	N	Y
GtmRegistration	Represents the registrations existing in external systems that are relevant for trade purposes. Global Trade Management Cloud supports different type of registrations: for parties (e.g. legal entities, company branches, customers, carriers, etc.) and for items.	Y	Y
GtmStructure	Represents Trade Item Structure (bill of material) to identify components that make up a single sellable product or item.	Y	N
GtmDeclaration	Represents a Global Trade Management Customs Declaration. Examples could be: import shipment, export shipment, inter-state shipment. The interface can be also used to pass the shipments information including charges	Y	Y

Interface	Description	Inbound	Outbound
	and relevant information to an external system such as an ERP Inventory Receiving or Landed Cost Management application.		
GtmDeclarationMessage	This Element maintains the information related to messages exchanged between different parties involved in the process of a Customs Declaration.		
GtmBond	Represents bond or guarantee related information.	Y	N
ServiceRequest	Request/Reply style interface to request the execution one of a number of services:- • Compliance Rule Screening • License Determination • Item Classification Screening • Restricted Party Screening • Sanctioned Territory Screening		
ServiceResponse	Request/Reply style interface reply to the ServiceRequest.	N	Y

Finance.xsd

Finance.xsd Interfaces

Interface	Description	Inbound	Outbound
Accrual	Used to transmit the allocated freight cost accrued or paid against orders. This can be used to communicate changes or differences in an order's allocated freight cost to other external systems. These differences, for example, could be used to establish liabilities in an accounting or general ledger system.	N	Y
Billing	Sends transmissions to an accounting system for customer billing. The billing information represents the amount owed by a customer to the planner of a shipment. The billing transmission includes the customer who is being charged and details of the bill such	N	Y

Interface	Description	Inbound	Outbound
	as the amount due, the date due, and any discount information.		
AllocationBase	Sends order release allocation cost information to an order owner. Allocation is a method for dividing the cost of a shipment among its order releases based on the line items and ship units on the shipment. Messages can be sent for planned and actual shipments.	N	Y
Invoice	Represents what is owed to the service provider or other third parties for transporting the shipment. The message content contains details about the payment and the specific transportation activity for which payment is owed.	Y	N
Voucher	Transmits payment information. A voucher represents the cost of a shipment owed to a third party such as a service provider and is designed to be sent after it has matched a third party charge to a shipment and approved the charge for payment.	N	Y
FinancialSystemFeed	Sends billing and shipment cost allocation information to an external financial system based on shipments (buy or sell). Can be used as an alternative to using Invoices or Bills as the source for financial information.	N	Y
Claim	Used to specify information for damaged shipments. A claim consists of any freight damage that occurs to a shipment prior to taking ownership/responsibility of the shipment.	Y	Y
WorkInvoice	Includes a record of driver activities for a shipment. This is the record sent to payroll and used to calculate driver pay.	Y	Y
ExchangeRate	Exchange Rate specifies the effective and expiration date for a set of currency conversions.	Y	Y

LocationContact.xsd

LocationContact.xsd Interfaces

Interface	Description	Inbound	Outbound
Location	A place where transportation related activities, such as loading and unloading freight, occur. In addition, a location is a corporation, and/or a service provider. Use the location element to transmit location information, for the Transportation Orders interface.	Y	Y
Contact	Defines a person or party that can be contacted through Transportation and Global Trade Management Cloud.	Y	Y
ContactGroup	Represents a list of contacts used for notification	Y	Y
Corporation			
ActivityTimeDef	This interface is used to specify fields for allowing multiple fixed and variable stop times for each location role based on transport handling unit and commodity.	Y	Y
PartySite	Defines the association between a Party (Contact) and a Site (Location)	Y	Y

Item.xsd

Item.xsd Interfaces

Interface	Description	Inbound	Outbound
ItemMaster	Transmits item master data to Transportation and Global Trade Management Cloud. Item master data represents the freight being shipped and includes : • Item numbers • Descriptions • Packaging details	Y	Y

Interface	Description	Inbound	Outbound
	• Applicable NMFC, STCC, SITC, or HTS codes • General ledger ID • Accessorial charges Item master data must exist in Transportation and Global Trade Management Cloud before you can create transportation orders.		
Item	Transmits item data to Transportation and Global Trade Management Cloud. Differs from Item master interface in that it only relates to Item and not Packaging.	Y	Y
HazmatItem	Sends and receive records for particular hazardous items.	Y	Y
HazmatGeneric	Transmits hazardous material based on the shipping name for an item.	Y	Y
Sku	Defines a stock keeping unit including what quantities to keep in stock, and the actual amount in the warehouse.	Y	Y
SkuTransaction	Represents a shipment of SKUs arriving or leaving the warehouse.	Y	N
SkuEvent	Specifies a SKU event, which describes activities on SKU's	Y	Y
PartnerItem	Represents the information for an item as referenced in partner external system.	Y	Y

Planning.xsd

Planning.xsd Interfaces

Interface	Description	Inbound	Outbound
Itinerary	Define the path between two locations and specifies the constraints for building the shipment.	Y	N
XLane	Defines a link from a source to a destination. The source and destination may specify either general or specific geography. For example, a source could be an exact location, or an entire	Y	N

Interface	Description	Inbound	Outbound
	state. May also contain mileage information.		
Mileage	Defines the distance between points for a particular lane. Lanes may contain specific or general geographic information, such as: • Zip code to zip code • City to city • State to state • Zip code to city • City to state • Address to address • City to address • State to address	Y	N
RouteTemplate	Route template represents the plan for a cooperative route. A cooperative route is a linking of lanes that have been identified to have sufficient recurring volume of shipments to form a good route for a fleet or dedicated vehicle.	Y	Y
BulkPlan	Describes the orders that Transportation and Global Trade Management Cloud planned and the shipments that Transportation and Global Trade Management Cloud created.	N	Y
BulkRating	Describes rating statistics on the orders that Transportation and Global Trade Management Cloud planned and the shipments that Transportation and Global Trade Management Cloud created.	N	Y
BulkTrailerBuild	Describes the shipment groups created during the bulk trailer build process.	N	Y
BulkContMove	Describes the shipments that were selected and linked during a given run of bulk continuous move.	N	Y
FleetBulkPlan	This is used to provide statistics about the fleet resource assignments that were created to the planned shipments during a given run of fleet bulk plan.	N	Y
ServiceTime	Transmits the time it takes to go between the two locations of a Lane. Service time information can also be included as part of the Mileage interface.	Y	N

Configuration.xsd

Configuration.xsd Interfaces

Interface	Description	Inbound	Outbound
User	Represents a Transportation and Global Trade Management Cloud user.	Y	N
CSVDataLoad	CSVDataLoad provides the capability to embed the contents of a CSV file for insertion into the database. Each CSVDataLoad element can contain only one CSV File. This element should only be used for small sets of data.	Y	N

Document.xsd

Document.xsd Interfaces

Interface	Description	Inbound	Outbound
Document	Provides a consistent way to send and receive business documents in and out of the system. Business documents are objects that contain the contents of a traditional document, like a bill of lading for example, in electronic format.	Y	Y

GenericTransaction.xsd

GenericTransaction.xsd Interfaces

Interface	Description	Inbound	Outbound
Topic	Raises a topic and gets Transportation and Global Trade Management Cloud to start processing an object.	Y	N
GenericStatusUpdate	Updates properties – status, remark, reference number and indicator – of various business	Y	N

Interface	Description	Inbound	Outbound
	objects including Location, Order Base, and Order Release etc. See schema documentation for a complete list.		
RemoteQuery	Queries Transportation and Global Trade Management Cloud for - Rating information, based on service provider, transportation mode, quantity, locations, and/or dates of a shipment. - Shipment information - Transmission Report – the processing result of a previous transmission.	Y	N
RemoteQueryReply	Provides Transportation and Global Trade Management Cloud's response to a RemoteQuery interface message	N	Y
TransactionAck	An external system sends a TransactionAck to acknowledge receipt and processing of a transaction.	Y	N
DataQuerySummary	Sends a summary of the data required by an external system.	N	Y

Job.xsd

Job.xsd Interfaces

Interface	Description	Inbound	Outbound
Job	Sends data that addresses how logistics services providers and freight forwarders manage the services they provide within Transportation and Global Trade Management Cloud. A Job offers a workspace that brings together the objects and activities required of them.	Y	Y

Rate.xsd

Rate.xsd Interfaces

Interface	Description	Inbound	Outbound
RATE_OFFERING	Represents a general contract with a service provider. It indicates what rate offering data was used to rate the shipment.	N	Y
RATE_GEO	Provides specific costing or rating data from one place to another. It indicates what rate record data was used to rate the shipment.	N	Y

ShipUnit.xsd

This schema does not define any external interfaces. It defines the common Ship Unit related structures needed by the other schemas; primarily Shipment.xsd and Order.xsd.

7 Inbound Transmission Processing

General Overview

The internal processing of inbound transmissions is highly configurable and can support a wide range of requirements from the very simple single object Create, Retrieve, Update and Delete (CRUD) interfaces through to multi-message groups and flexible sequencing involving complex workflow.

The following steps outline the default process for a simple Transactional style transmission. The various configuration options are described in the subsequent sections.

Processing Overview

On receipt of a Transmission XML, the integration application module performs the following steps. Note that authentication and authorization will already have been performed in a protocol specific way before the message is passed to the module:

1. Parse and validate Transmission XML according to schemas, generate a new unique Transmission Number and persist records in the I_TRANSMISSION and I_TRANSACTION tables for this transmission number.
2. Depending on transport protocol used, and if requested, create a TransmissionAck XML message and return to the sender. Schema validation errors will be returned at this time.
3. An event is staged in the inbound integration data queue which will be the default or override queue specified in the incoming message. When the event is polled by a data queue poller the message is then placed on the internal Integration Workflow in-memory queue. See the *Integration Data Queues chapter* for more details on data queue configuration.
4. The Integration Workflow retrieves the message from the internal queue and:
    a. Determines any grouping or sequencing constraints
    b. Submits group(s) to Mediator for processing.
5. When the Mediator selects a group for processing, it will process each Transaction in the group as follows:
    a. If extended validation is configured the integration module validates the following:
        i. Foreign keys, for example a Location GID must exist in the Location table.
        ii. Data Types, for example a number only contains numeric characters.
        iii. TransactionCode. If set to I, the integration module checks that the primary key does not exist. If set to U or D, the integration module checks that the primary key does exist.
        iv. Required Elements are not NULL.
    b. Saves any validation errors.
    c. If there are no validation errors or if extended validation was not configured, the corresponding business object is populated with the data from the XML and passed to the persistence module.
    d. If there are agents listening for pre-persist events, the persistence module triggers those now. An example of this is the public Order Base - Insert agent for TransOrder transmissions.
    e. The business objects are now persisted to the corresponding database tables. If extended validation had not been configured all integrity constraints are now checked and any violations would cause the transaction to fail.
    f. If there are agents listening for post-persist events, the persistence module triggers those.

6. When the Mediator has processed all transactions for all groups in a Transmission the Transmission processing is complete. The completion process optionally sends a TransmissionReport with validation and processing errors. See *AckSpec* for details.

> **Note:** The Transmission Report is only physically transmitted when all workflow that is considered to be part of the Transmission Process has completed (known as "child" processes). If there is a significant amount of agent workflow configured this could appear as a delay in delivering the report. If this is an issue it may be worth considering the use of customer-defined events with the "Create New Process" option selected.

Transmission Status

Each Transmission sent into Transportation and Global Trade Management Cloud has a status field that indicates the state of the Transmission at each stage in the processing. The state of the Transmission will change as it progresses through the various steps outlined in *Processing Overview*. You can view the status for a Transmission in the Transmission Manager UI that can be accessed via the following menu: **Business Process Automation > Integration > Transmission Manager**. The status of the transmission could be one of the following:

Transmission Statuses

Status	Description
STAGED	Indicates that the Transmission has been parsed and is stored in the transmission and transaction tables. An internal event will also be queued which, when triggered, will process the transmission content. No Transmission should stay in a STAGED status permanently. Such a scenario would normally suggest an application configuration issue.
STAGING	Indicates that the Transmission has been persisted and is waiting on a condition to be released for processing. In this status, additional transactions can be added to it prior to processing. Refer to *Inbound Transmission Staging and Processing* for additional details.
FRESH	Indicates that the Transmission is waiting to complete processing. The individual transactions may still be processing. Normally a STAGED Transmission will go to FRESH status when its Transactions have been sent to the Mediator for scheduling.
ERROR	Indicates that the Transmission had completed processing and there were errors in the processing.
PROCESSED	Indicates that the Transmission successfully completed processing.
REDO	Indicates that the Transmission is waiting for the REDO logic to initiate re-processing of it. See the *REDO Transmission Processing help* for details.

Extended Inbound Validation

After Transportation and Global Trade Management Cloud has processed a transmission, Transportation and Global Trade Management Cloud sends back a Transmission Report to the external system with a list of validation and/or processing errors.

System administrators can set the default level of extended validation that Transportation and Global Trade Management Cloud performs. Changing the validation level can improve performance by removing unnecessary queries and logic for validating the data. The default level of validation is specified via the following glog.integration.validation property. See the *glog.integration Properties help* for the list of possible values for the property.

If your Transportation and Global Trade Management Cloud installation validates all transmissions fully, you can skip validation for certain transmissions on a case-by-case basis. Just include this processing instruction in all transmissions where Transportation and Global Trade Management Cloud should skip validation:

```
<?gc3-xml-process validate_required_fields="N"?>
```

If instead your Transportation and Global Trade Management Cloud installation never validates errors or only validates when receiving persist errors from the Oracle Database, you can get Transportation and Global Trade Management Cloud to validate certain transmissions on a case-by-case basis. Include this processing instruction in all transmissions Transportation and Global Trade Management Cloud should validate:

```
<?gc3-xml-process validate_required_fields="Y"?>
```

Case Sensitive Data

Transportation and Global Trade Management Cloud provides functionality to automatically change text to upper case when processing the inbound XML. You can enable the functionality by setting the following property:

```
glog.integration.enableCaseChange=true
```

If there are some elements that you do not want to change, you can set the `glog.integration.casechange.element` property for those elements. Valid values are element names. For example, to prohibit the ArgName and ArgValue values from changing, you would define the following:

```
glog.integration.casechange.element=ArgName

glog.integration.casechange.element=ArgValue
```

"Replace" Transaction Codes

The following example Shipment XML structure illustrates the basic rules for transaction codes starting with "R".

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

If the data is not found in database, the shipment is inserted into database. In this case, the RC is equivalent to IU. The use cases described in the following sections are edited from this XML.

- Remove ShipmentStop 2 and two ShipUnitContents of ShipUnit A from the above XML.
- Expected result: ShipmentStop 2, two shipUnitContents as well as their corresponding children are deleted from database.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 <!-- ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1
 ShipUnit: ShipUnitGid=ShipUnit B
 <!--ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4 -->
 SEquipment: SEquipment_A
```

- Remove ShipmentStop 2 from the original XML and add ManagedChild=ShipmentStop.
- Expected result: ShipmentStop 2 and all its child tables are deleted from database.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 <!-- ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

If the Is_permanent of the above stop in database equals true or the ManagedChild is set to a value other than ShipmentStop or ShipmentStopDetail, the stop as well as it child tables are not able to be removed from database.

- Remove ShipmentStopDetail elements of ShipmentStop 2 and add ManagedChild = ShipmentStop from the original XML.
- Expected result: ShipmentStopDs of ShipmentStop 2 are deleted from database.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 <!-- ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

You can set ManagedChild=ShipmentStopDetail in order to get the same result.

- Remove ShipmentStopDetail elements of ShipmentStop 2 and ShipUnitContent elements of ShipUnit A with ManagedChild= ShipmentStop from the original XML.
- Expected result: ShipmentStopDs for ShipmentStop 2 are deleted. SShipUnitLines are unchanged. In theory, SShipUnitLines should be deleted too. However, this is an exception case since ShipUnit, SEquipment and Text are not really child or grandchild nodes of shipment. They can independently exist in database. In order to delete ShipUnitContent, you have to specify the ShipUnitContent in ManagedChild as described in next section

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 <!-- ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 <!-- ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1 -->
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

- Remove ShipUnitContent elements of ShipUnit A with ManagedChild=ShipUnit or ManagedChild=ShipUnitContent.
- Expected result: The SShipUnitLines 0 and 1 are deleted from database. SShipUnit in the XML is replaced.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild = ShipUnitContent
 OR
 ManagedChild = ShipUnit
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipUnit: ShipUnitGid=ShipUnit A
 <!-- ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1 -->
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

- Add two or more ManagedChild elements (ManagedChild=ShipmentStop, ManagedChild=ShipUnit) and remove ShipmentStopDetail elements in ShipmentStop 2 and ShipUnitContent elements of ShipUnit A.
- Expected result: ShipmentStopDs of ShipmentStop 2 and SShipUnitLines of ShipUnit A are unchanged.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ManagedChild =ShipUnit
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 <! -- ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 <!-- ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1 -->
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

- Remove ShipUnit A and the ShipUnitContent 3 and 4 for the remaining ShipUnit B with ManagedChild=ShipUnit.
- Expected result: SShipUnit corresponding to ShipUnit A is unchanged. SShipUnit corresponding to ShipUnit B is replaced. The SShipUnitLines corresponding to ShipUnitContents 3 and 4 are deleted from database.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild = ShipUnit
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 <!-- ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1 -->
 ShipUnit: ShipUnitGid=ShipUnit B
 <!-- ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4 -->
 SEquipment: SEquipment_A
```

- The same result can be achieved through specifying the transaction code R and ManagedChild = ShipUnitContent in ShipUnit B.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 <!-- ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1-->
 ShipUnit: ShipUnitGid=ShipUnit B
 TransactionCode= R
 ManagedChild = ShipUnitContent
 <!-- ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4 -->
 SEquipment: SEquipment_A
```

- Change ManagedChild = SEquipment with TransactionCode= RC in ShipmentHeader.
- Expected result: SEquipment_A data in database will be replaced with SEquipment data.

```
Shipment
```

```
 ShipmentHeader
 TransactionCode RC
 ManagedChild = SEquipment
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 ShipUnitContent: LineNumber=1
 ShipUnit: ShipUnitGid=ShipUnit B
 ShipUnitContent: LineNumber=3
 ShipUnitContent: LineNumber=4
 SEquipment: SEquipment_A
```

- Remove ShipmentStop 2 with ManagedChild=ShipmentStop from the original XML and change the transaction code to R.
- Expected result: ShipmentStop 2 and all its child tables are deleted. The shipment table is replaced (This is different from transaction code RC).

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 SEquipment: SEquipment_A
```

- Change the transaction code to RP in the XML above.
- Expected result: Only shipment table is replaced. The ManagedChild element is ignored. ShipmentStop 2 and all its child tables are unchanged.

```
Shipment
 ShipmentHeader
 TransactionCode RC
 ManagedChild =ShipmentStop
 ShipmentStop=1
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B
 <!-- ShipmentStop=2
 ShipmentStopDetail: ShipUnitGid=ShipUnit A
 ShipmentStopDetail: ShipUnitGid=ShipUnit B -->
 ShipUnit: ShipUnitGid=ShipUnit A
 ShipUnitContent: LineNumber=0
 SEquipment: SEquipment_A
```

The value for element ManagedChild is defined in CHILD_ELEMENT_ALIAS column of INT_MANAGED_CHILDREN_MAP table. This value can be viewed or edited from Power Data page found by navigating to **Business Process Automation > Power Data > Integration > Managed Children Map**.

Time Zone Override

Generally, time information sent in the inbound XML to Transportation and Global Trade Management Cloud is associated with a time zone for a particular location and time zone offset. For example, the early pickup dates and late delivery dates on a transportation order line (within the TransOrder interface) are associated with the ship from and ship to locations for the order line, and each of those locations is associated with a specific time zone. The time information sent in the early pickup date and late delivery dates elements are assumed to be in the time zone of the location. So, if your time is at 8 AM and the location is King of Prussia, PA, you would specify the time in the XML as 080000 (HHMISS, where HH=Hours, MI=Minutes and SS=Seconds) and Transportation and Global Trade Management Cloud would understand this to mean 8AM in the America/New York time zone.

For systems that only maintain their times in a single time zone, or in other circumstances when you are unable to specify times as expected in the time zone of the corresponding locations, an override can be specified to indicate the time zone for all times for the transaction. As an example, if your sending application maintains all times in a specific time zone such as San Francisco where the early pickup date is maintained as 5 AM, it would be possible to send 050000 as the time to Transportation and Global Trade Management Cloud and indicate an override that all the times in the transportation order are specified in the America/Los_Angeles time zone.

A valid time zone override can be specified in the TimeZoneGid element within the Transaction Header. When the TimeZoneGid is specified, it will be assumed that all the times within the transaction (GLogXMLElement) are in that time zone.

Business Number Generator (BNG)

When creating new objects i.e. with the 'I' TransactionCode, it is possible to send a transmission without entering a value in the primary key GID element (for example, Ship Unit ID, Order Release ID, Order Base ID, etc.). Transportation and Global Trade Management Cloud generates values for these fields based on the default business number rules for the object type.

Using Null Values

Null values in XML messages are usually represented by the absence of the corresponding element. However, when the desire is to change an elements current value to null a different approach is required.

In Transportation and Global Trade Management Cloud XML messages, to delete (null out) values from certain fields in a record (without deleting the whole record), you can specify the (~) character in the element. For example, if a value was entered for the External System ID field in the TransOrderHeader, and that value needed to be removed in a subsequent TransOrder update, the following would be specified for the XML:

```
<ExternalSystemId>~</ExternalSystemId>
```

Using Integration Saved Queries

You can select or identify objects to update or delete without using a GID by using a configurable matching integration saved query. Integration Saved Queries are defined in Power Data via the following menu: **Business Process Automation > Power Data > Integration > Integration Saved Queries**. The queries are written as SQL statements that contain references to the information in the incoming XML transmission. For example, a query for a shipment GID given a shipment reference number would be as follows (e.g. if query GID is "GUEST.TEST_SAVED_QUERY_001"):

```
select s.shipment_GID from shipment_refnum s where s.shipment_refnum_qual_GID = '{%QUAL%}' and
 s.shipment_refnum_value = '{%VALUE%}'
```

The inbound XML would then contain the following IntSavedQuery element to use the above query:

```
<IntSavedQuery>
 <IntSavedQueryGid>
 <Gid>
 <DomainName>GUEST</DomainName>
 <Xid>TEST_SAVED_QUERY_001</Xid>
 </Gid>
 </IntSavedQueryGid>
 <IntSavedQueryArg>
 <ArgName>QUAL</ArgName>
 <ArgValue>ZZ</ArgValue>
 </IntSavedQueryArg>
 <IntSavedQueryArg>
 <ArgName>VALUE</ArgName>
 <ArgValue>MY_SHIP_REFNUM_001</ArgValue>
 </IntSavedQueryArg>
 <IsMultiMatch>Y</IsMultiMatch>
 <NoDataFoundAction></NoDataFoundAction>
</IntSavedQuery>
```

During processing, Transportation and Global Trade Management Cloud would then replace the '{%QUAL%}' with 'ZZ' and the '{%VALUE%}' with ' MY_SHIP_REFNUM_001' to search for the shipment GID. And the transaction would be processed with using the shipment GID returned from the query. Note that the `IsMultiMatch` element can be used to indicate if multiple GIDs can be returned from the query and used for processing. Refer to the schema notes for additional fields that are available for the query.

The queries can also be written using XPath expressions to search for values from specific elements in the XML. An example of a query that relies on XPath expressions is as follows:

```
select ob.ORDER_BASE_GID from OB_REFNUM ob where ob.ORDER_REFNUM_QUAL_GID = 'PO' and ob.OB_REFNUM_VALUE
 = '{TransOrder/TransOrderHeader/OrderRefnum[OrderRefnumQualifierGID/GID/Xid='PO' and
 (not(OrderRefnumQualifierGID/GID/DomainName) or OrderRefnumQualifierGID/GID/DomainName = '' or
 OrderRefnumQualifierGID/GID/DomainName = 'PUBLIC')]/OrderRefnumValue}'
```

Note that the preferred method is to use the ArgName and ArgValue as they perform much better than the XPath expressions.

If `NoDataFoundAction` is not null and the Integration Saved Query returns no value, then the transaction code for the XML is switched to the transaction code specified by `NoDataFoundAction`, which must be a valid transaction code.

Note: The `NoDataFoundAction` tag does not work on child elements. It is designed primarily for the root elements.

Default Integration Saved Queries for Updates

There are a few interfaces that support default integration saved queries without having to specify the IntSavedQuery element in the inbound XML. This functionality is supported for Location, TransOrder, TransOrderLine, and Shipment.

The default integration saved queries are only used when the transaction code element has a value of U or D, and the primary GID for the interface is missing. If a TransOrderGid is missing, then the INT_TRANS_ORDER_GID_1 and INT_TRANS_ORDER_GID_2 saved queries are used. If a TransOrderLineGid is missing, then the INT_TRANS_ORDER_LINE_GID_1 and INT_TRANS_ORDER_LINE_GID_2 saved queries are used. If a Shipment GID is missing, then the INT_SHIPMENT_GID_1 and INT_SHIPMENT_GID_2 saved queries are used. If a Location GID is missing, then the INT_LOCATION_GID_1 and INT_LOCATION_GID_2 saved queries are used.

The defined queries must return a single GID of the element being referenced (for example, Order_Base_GID for TransOrder, Shipment_GID for Shipment, etc.). When a query returns multiple records, it will generate an error. Transportation and Global Trade Management Cloud supports up to two queries for each interface/record. If the first query generates an error or does not resolve to a single record, the second query will be applied. And if nothing is returned, then an error is generated.

Character Encodings

The default character encoding for inbound XML transmissions is UTF-8. However, Transportation and Global Trade Management Cloud can receive transmissions in other encodings. The database character set must be UTF-8 which means that all the characters, irrespective of initial encoding, can be stored in the database. To accept a different character encoding:

1. Specify it in the XML file. For example, `<?xml version="1.0" encoding="ISO-8859-1"?>`
2. Save your XML file using that same encoding.
3. If sending via HTTP post, you will also need to specify the encoding in the HTTP header. You must use the "Content-Type" attribute on the HTTP header to indicate that it's a stream of character data with a given encoding.

Most modern text editors have the capability to save files in various encodings.

Inbound Transmission Staging and Processing

A Transmission is ready to be processed when it is in the STAGED state, which means it has been parsed and all Transactions and Process groups have been determined.

However, for certain scenarios it is feasible that transactions that should be processed together do not all arrive in one Transmission. For example, depending on the message routing for various applications, a Tracking Event transaction could arrive from a WMS system a few seconds before the transaction from a Planning system which actually creates the shipment to which the tracking event should apply.

Transportation and Global Trade Management Cloud supports this scenario by using **Staging Transmissions**.

Staging Transmissions are identified by a `TransmissionType` of STAGING. This is an indication that processing can be delayed until some later time or event. The logic which should be applied to the Staging Transmission must be defined in the `StagingInfo` content.

Staging Info

The `stagingInfo` content in the Transmission header defines the following:

- **A Staging Query**: The `stagingQuery` element is an Integration Saved Query which should return one or more Transmission numbers. The first transmission number returned from the query is used as the transmission number to which the transactions in the current transmission are appended thus constructing a more complete Transmission.
- **A Staging Process**: The `stagingProcess` element is an Integration Saved Query which determines if the Transmission is now ready for processing. If the query returns a result, the Transmission will proceed with processing, and if no result is returned then the transmission will remain in a STAGING status until another event triggers it to be processed.

> **Note:** Initiating the processing of the Staging Transmission can also be enabled through the TRANSMISSION agent type, and the Staging Transmission Processing process management actions.

Staging Info Examples

To put a transmission immediately into a STAGING state, use the following TransmissionHeader:

```
<TransmissionHeader>
 <TransmissionType>STAGING</TransmissionType>
</TransmissionHeader>
```

If a Transmission contains the above header the transactions within the transmission will remain in STAGING state until a subsequent transmission or process changes their state.

If the current transmission should also be put into STAGING state but the transactions contained within it should be added to an already existing transmission also in STAGING state, the following TransmissionHeader should be used:-

```
<TransmissionHeader>
 <TransmissionType>STAGING</TransmissionType>
 <StagingInfo>
 <StagingQuery>
 <IntSavedQuery>
 …integration saved query elements…
 </IntSavedQuery>
 </StagingQuery>
 </StagingInfo>
</TransmissionHeader>
```

The Integration Saved Query should return the transmission number of the previously staged transmission which should have the current transactions added to it. If no result is returned the transactions remain associated to this transmission and it will remain in STAGING state.

8 Outbound Transmission Processing

General Overview

All outbound transmission processing requires the configuration of an External System to which the messages will be sent. At a minimum, the External System must define the protocol to be used for the transport and any required external credentials.

The character encoding for all outbound transmissions is UTF-8.

There are various ways to trigger the sending of Transmissions to the external system from Transportation and Global Trade Management Cloud:

- Some messages are sent automatically as the result of workflow notification.
- Send and schedule integration transmissions in the Process Manager.
- Re-send and schedule integration transmissions in the Process Manager.
- Send transportation records to external systems from various managers using the Finder or Manager UI actions. The content of the information that is transmitted is determined by the context of the UI in Transportation and Global Trade Management Cloud from which you are sending it.

Processing Overview

Once an event triggers the requirement to send a Transmission for one or more objects, the following processing takes place. The exact steps differ depending on the media type of the target message i.e. whether the External System is expecting "application/xml" or "application/json".

1. Construct internal objects based on SQL Queries. Out XML Profiles are applied during the generation of the objects for XML messages.
2. Convert the objects into a string representation.
3. If anything other than the Current Target Namespace is selected then this conversion takes place now.
4. Apply XSL Transformations if needed (for XML).
5. Save the resulting output to the I_TRANSMISSION table. If the outbound integration is designated as a Mobile Device message (e.g. generated via the Compose and Send UI action or dispatch actions), then an entry is also stored in the I_MESSAGE table.
6. Send the resulting output via the indicated notify type (e.g. HTTP POST). The External System record in Transportation and Global Trade Management Cloud indicates whether or not an acknowledgment is required from the external system.
7. Updates the status of the Transmission.

External User Credentials

When sending messages outbound from Transportation and Global Trade Management Cloud it may be required to specify the username and password required by the receiving external server. This can be achieved by using the username related fields on the external system.

Outbound XML Profiles

Out XML Profiles are used to exclude portions of outbound XML with a high degree of control. They reduce the size of the XML and minimize the number of queries that are used to generate the XML, thereby reducing the memory and time used and improving overall performance. There are several options for specify the elements to exclude. See the *Out XML Profiles help* for details.

> **Note:** There is currently no corresponding control for JSON messages.

Transform Outbound XML with XSL

Do the following to have Transportation and Global Trade Management Cloud transform your outgoing transmission from the Transmission schema to some other XML schema.

1. Define an XSL file that transforms between the schemas.
2. Upload the XSL file to create a Stylesheet Content record.
3. Create Stylesheet Profile to reference Stylesheet Content record created above.
4. Modify external system to reference Stylesheet Profile created above.

> **Note:** Referencing Stylesheet files in the External System is deprecated and will be removed in a future version. Please convert to using Stylesheet Profile/Stylesheet Content as soon as possible.

> **Note:** There is currently no capability to produce custom JSON messages.

Support for Responses to Outbound Messages

The outbound integration processing supports receiving and processing of synchronous responses to outbound messages as a functional acknowledgement of receiving the Transmission and/or Transaction. Agent objects have been added for types "TRANSMISSION OUT" and "TRANSACTION OUT" to provide custom workflow handling of the response message.

For outbound XML messages these are the TransmissionAck, TransactionAck, and TransmissionReport.

For outbound JSON messages this is the "transmissionStatus" resource. (See the REST API online documentation.)

Integration Unit of Measure Preferences

You can specify the Unit of Measure (UOM) preference and precision to be displayed in outbound XML by defining preference for each UOM. These preferences can be defined with the Integration Preference UI that can be accessed via the following menu: **Business Process Automation > Power Data > Integration > Integration Preferences**. Integration preferences can be associated to an external system or an out XML profile. See the *Integration Preference help* for details.

Integration preferences can also be associated with the Rate Inquiry and Generic Query interfaces. To use the integration preference in those synchronous interfaces, you would specify the IntPreferenceGid in the RIQQuery, or the GenericQuery XML. When specified in the RIQQuery, the RIQQueryResponse would apply the preferences before responding with the result XML. When specified in the GenericQuery XML, the preference would be applied to the resulting generated interface XML.

Controlling Target Namespace

A field on the External System Manager UI will specifies the target namespace URL to be used for outbound messages. There are the following possible values:

- **Current**: causes the latest version to be output
- **Versions prior to 6.4.2**: performs backward compatibility conversion from new to old namespace URL.
- **None**: causes the latest version to be output but with the namespace declarations removed.
- **None (Compatibility Mode)**: performs backward compatibility conversion from new to old namespace URL format and then also removes the namespace declarations.

All new External Systems will default to the Current target namespace.

9 REST Integration

REST Integration

Refer to the *REST online documentation* for details on inbound integration and outbound integration via the REST com method.

10 Web Service Integration

Web Service Security (WS-Security)

The Web Service Security Specification is an OASIS standard for defining security related information as part of a SOAP message. See *OASIS Open*.

Both Inbound and Outbound web services security requires the correct WSS policy to be used.

Inbound

Transportation and Global Trade Management Cloud supports a number SOAP Web Services for sending messages inbound to the application. All available services are described in the following table.

SOAP Web Services for Sending Inbound to the Application

Service Name	Purpose
TransmissionService	Process Transmission documents according to the Transmission Schemas.
CommandService	Processes requests for DBXML export and import.
Workflow Web Services	Process agent actions against one or more existing transactional objects. See the *Workflow Web Service section* for details.
MessageService	Process Mobile Device messages. See the *Mobile Device Communications section* for details.

Security

All Inbound Web Services default to using a Web Service Security (WSS) policy which requires a Username Token to be passed and to be securely transported over HTTPS.

Accessing Web Services Description Language (WSDL)

Each application server provides a different service endpoint URL for accessing the WSDL file. A WSDL can be accessed via the following menu path: **Business Process Automation > Integration > Integration Manager > Retrieve WSDLs** on main page.

The page also provides access to any related XSD schema files for each service.

Transmission Service

The Transmission Service is the standard inbound Web Service for processing of Transmission XML documents.

TransmissionService is a SOAP Document Literal WRAPPED style of web service which means the SOAP Body of the SOAP message contains a Transmission XML wrapped in an element which matches the name of the operation to be performed. This style is recommended by the OASIS Web Services Interoperability Basic Profile. See *OASIS Open* for details.

The service has two operations:

- `publish`: This operation accepts and stages a Transmission for processing and returns a TransmissionAck as response.
- `execute`: This operation synchronously processes a Transmission and returns a reply Transmission with the processing results. For example, this operation should be used for `RemoteQuery` and `ServiceRequest` interfaces. In future, other interfaces may be supported inbound.

Refer to the TransmissionService WSDL for each operations input and output supported schema types.

Outbound

It is also possible to configure a Transportation and Global Trade Management Cloud External System to use an external web service endpoint as the destination for an outbound message.

The external system is configured to use a Web Service record created from the WSDL for the external service. See the *Web Service help* for details but, at a high level, the main steps are as follows:

1. Create a Document in Transportation and Global Trade Management Cloud from the external service WSDL URL.
2. Create a Web Service record in Transportation and Global Trade Management Cloud using the Document created in step 1.
3. Configure External System to reference Web Service for the chosen Web Service operation and service endpoint.

> **Note:** A single service can offer multiple operations.

In step 1, the document can be created from an uploaded text file containing the definitions (normally with the .wsdl file extension) or by specifying the URL for the WSDL resource. See the *Document Manager help* for details.

> **Note:** If the external service specifies a WS-SecurityPolicy (see below) then the document must be created using the URL.

Security

The following WS-Security policies are supported for outbound web services:-

- Username Token Profile: uses username and password configured on Web Service Endpoint.

- HTTPS Transport
- Message Encryption

Service Call Pattern

Transportation and Global Trade Management Cloud outbound Web Services support both Document Literal WRAPPED and Document Literal BARE styles. Document Literal WRAPPED is the preferred style as it offers the highest level of interoperability.

11 HTTP Integration

Inbound HTTP Integration

There are various options for sending integration messages into Transportation and Global Trade Management Cloud using HTTP:

- **Standard Transmission XML integration**: Requires XML which conforms to the XML schemas described within the *Transmission Schema* section.
- **Transformed Non-standard or external XML format**: A custom XSLT stylesheet can be applied to inbound XML to convert to the standard Transmission format. See the *Transform Inbound XML with XSL* section for additional details.
- **Direct Load of standard Transmission**: Used for optimised pre-load of business objects without triggering internal workflow or automation agents. Refer to the *DirLoad* section for additional details.

The specific URL path to call for each of the supported cases depends on the desired method of authentication. There are currently the following options:

- HTTP Authentication (Basic)
- OAuth 2

HTTP Authentication (Basic)

All integration messages using HTTP as the transport should use HTTPS URLs i.e. secured by Transport Layer Security (TLS).

The internal Transportation and Global Trade Management user credentials must be passed in the HTTP Authorization header (*RFC 7617*) i.e. not the Cloud SSO credentials. The following URL endpoints must be targeted (relative the Cloud server/port):

Endpoints and Relative URLs

Endpoint	Relative URL
Standard Transmission XML	GC3/glog.integration.servlet.WMServlet
Transformed	GC3/glog.integration.servlet.TransformerServlet
Direct Load	GC3/glog.integration.servlet.DirLoadServlet
DB XML	GC3/glog.integration.servlet.DBXMLServlet

If HTTP Authentication (Basic) is not required, it is recommended to disable the support for it. See the *Remove Inbound HTTP Basic Authentication Support for External Integration End Points* section in the Oracle Fusion Cloud Transportation and Global Trade Management Security Guide.

OAuth 2

The Transportation and Global Trade Management application also supports OAuth 2 Client Credentials authentication. The Oracle Identity Cloud Service (IDCS) processes the authentication request and provides the access token that is used to access the application integration servlets.

See the *OAuth 2 section of the Security chapter* for configuration details.

The following URL endpoints must be targeted (relative the Cloud server/port):

Endpoints and Relative URLs

Endpoint	Relative URL
Standard Transmission XML	logisticsXmlApi/Transmission
Transformed	logisticsXmlApi/Transformer
Direct Load	logisticsXmlApi/DirLoad
DB XML	logisticsXmlApi/DBXML
Device Message	logisticsXmlApi/DeviceMessage

> **Note:** The supported content types for HTTP messages (corresponding to the `Content-Type` HTTP header parameter) are `text/xml` and `application/xml`.

Transform Inbound XML with XSL

You can configure Transportation and Global Trade Management Cloud to transform your inbound transmissions from another XML schema to a valid Transmission XML. To do this:

1. Create a Stylesheet Content record in the Transportation and Global Trade Management Cloud Database.
2. Include this processing instruction in the beginning of every transmission that Transportation and Global Trade Management Cloud needs to transform: `<?gc3-int-translate stylesheet_name="stylesheet_name"?>`, `stylesheet name` is the GID of the required Stylesheet Content record prefixed with "db:". You can include multiple processing instructions in one transmission, in which case, Transportation and Global Trade Management Cloud will transform in the order the processing instructions appear in the transmission.
3. Post XML documents to `http://hostname/GC3/glog.integration.servlet.TransformerServlet` instead of `http://hostname/GC3/glog.integration.servlet.WMServlet`.

> **Note:** The use of uploaded (file-system based) stylesheets instead of Stylesheet Content is still supported for backward compatibility but is deprecated and their use should be replaced with Stylesheet Content as soon as possible. Support for file-system based stylesheets will be removed in a future version.

Outbound HTTP Integration

HTTP is used as the transport protocol for both the original HTTP POST and new REST communication methods that can be selected by the workflow process which triggers the outbound message. REST provides access to more HTTP methods e.g. PATCH and DELETE, whereas HTTP POST as its name suggests, is limited to just HTTP POST.

Therefore, many of the same configuration options apply to both. Use cases where there are differences will be highlighted explicitly.

Security

There are multiple supported approaches for HTTP authentication:

- None (Default).
- HTTP Authentication (Basic)
- OAuth 2.0 – Client Credentials (only available for REST com method)

The External System has an Authentication Type option in the user interface where the selection can be made.

No Selection

When no selection is made no credentials will be passed to the external URL.

External Systems migrated from a previous version will not show any selected option. However, if a username and password had previously been specified, this will be equivalent to HTTP Authentication (Basic). In these cases, it is recommended to explicitly select this option to declare this requirement to avoid any future confusion. For example, if there is a genuine reason not to send credentials then any values in the present Username and Password fields would also have to be removed.

HTTP Authentication (Basic)

The username and password credentials from the External System, if present, are used as the HTTP Basic Authorization authentication properties on the outbound HTTP connection. HTTPS URLs are also supported. These credentials are sent using the standard Authorization HTTP header as defined in *RFC 7617*.

OAuth 2.0 – Client Credentials

OAuth 2.0 is the de facto standard for authentication and authorization for REST APIs and is now available as an option for outbound Transmission XML when sent using the "REST" communication method.

The roles involved in the protocol are:

- **Resource Owner**: The entity who owns and controls who can access the protected resource. The resources can be limited by "scope".
- **Authorization Server**: The entity trusted by the resource owner to apply the authentication and authorization process.
- **Client**: The user or application requesting access to the protected resource.

The protocol works by running the following stages:

1. Client submits Authorization request to obtain a token.

2. Client requests resource and passes token.

Each step above requires its own URL to process the request. Additionally, the Authorization Server can use one of a number of protocols to run the process of receiving an access token:

- **HTTP Authentication (Basic)**: This must be supported and is expected to be the most common use case. This option also uses the Username, Password and Application Scope fields to pass as the Client Credentials for the authorization.
- **HERE OAuth 1.0 Signature**: This is specific to the HERE REST API (not covered here).

12 Setting Up Interfaces

General Information

The following general information helps you set up your interfaces. If an interface has specific setup requirements, they are found with the pages defining each interface.

To set up interfaces, you must define where to send transmissions and what to do with the transmissions Transportation and Global Trade Management Cloud receives. Information throughout Transportation and Global Trade Management Cloud acts interdependently; one piece of information depends on another to perform an action. For some interfaces to work, data from other sources must already be present in Transportation and Global Trade Management Cloud. For example, before you can create a shipment, you must create itineraries.

Define External Systems

To send transmissions to other systems, you must define the systems in Transportation and Global Trade Management Cloud using the External System Manager.

User Management

You must add service providers as users and enter user associations for them. To perform user management functions, log in to the SERVPROV with a username that contains administrator (ADMIN) rights.

- Define service providers as users in Transportation and Global Trade Management Cloud.
- Define associations for the service providers.

Workflow Parameters

In Power Data, define workflow parameters that determine how Transportation and Global Trade Management Cloud responds to inbound and outbound transmissions. You define Workflow Power Data topics to define the way the tendering shipments works.

- **Workflow Parameters**: Use the Workflow Parameters to define how Transportation and Global Trade Management Cloud tenders shipments. You also define shipment notification messages. For example, you define information, warning, and fatal messages that Transportation and Global Trade Management Cloud sends out as the results of status information sent by service providers about particular shipments.
- **Workflow Trigger Parameters**: Use the Workflow Trigger Parameters to define how often Transportation and Global Trade Management Cloud performs tender activity. This topic helps you control system performance. For example, if Transportation and Global Trade Management Cloud is performing tender actions too frequently, your system performance may be slowed.

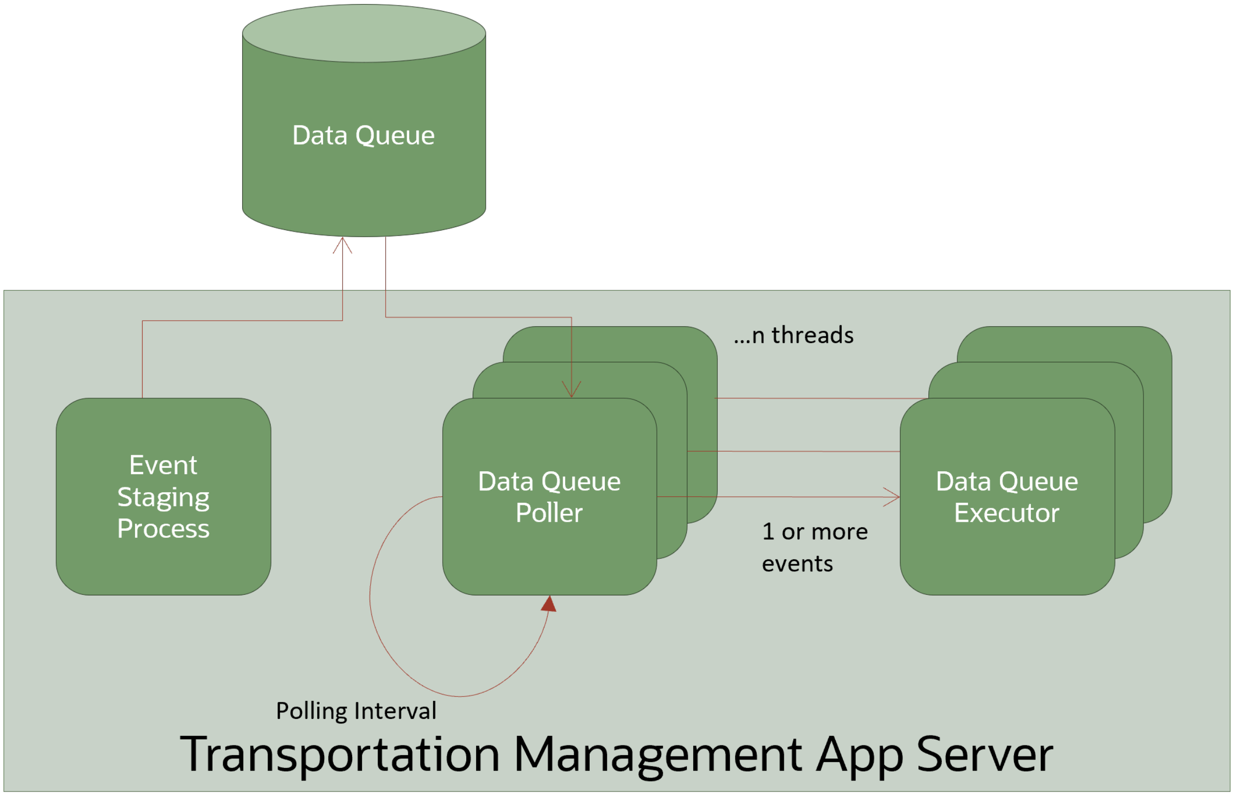
Agent Manager

The Automation Agent Manager lets you construct workflow agents that are key components to automate Transportation and Global Trade Management Cloud. A workflow agent listens for an Transportation and Global Trade Management Cloud event, verifies a user-defined condition, and runs one or more actions that you choose from an action library.

13 Integration Data Queues

Overview

By default, the Transportation and Global Trade Management Cloud application server workload is managed by a set of in-memory event queues; with a configurable number of threads available to process each queue. The Transportation and Global Trade Management Cloud Data Queue infrastructure was introduced to permit a more configurable fine-grained control of various aspects of this internal workload. The Data Queues are database resident queues where a **configurable** number of "poller" threads retrieve a **configurable** number of events at a **configurable** interval and pass them to an "executor" to be processed.



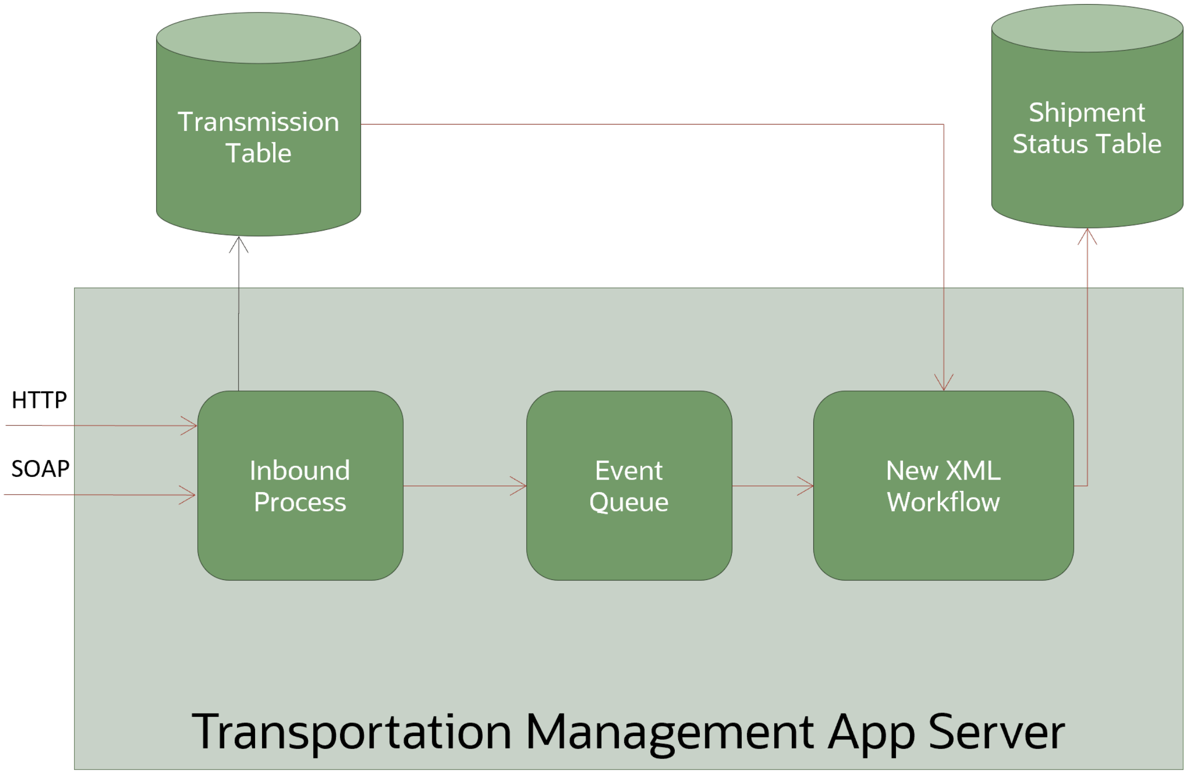
> **Note:** Although the entity names used are similar, Transportation and Global Trade Management Cloud Data Queues do not use Oracle Advanced Queue (OAQ) schema objects for processing data queue events. Oracle Advanced Queueing is not supported.

The internal processes for Inbound and Outbound Integration can now be configured to use this infrastructure.

> **Note:** See the *Data Queue Manager help* for coverage on the User Interface used to configure each data queue.

The following figure shows the application server processing of an inbound Shipment Status XML without the use of data queues. The XML message will arrive via one of the supported protocols: HTTP or SOAP Web Service.

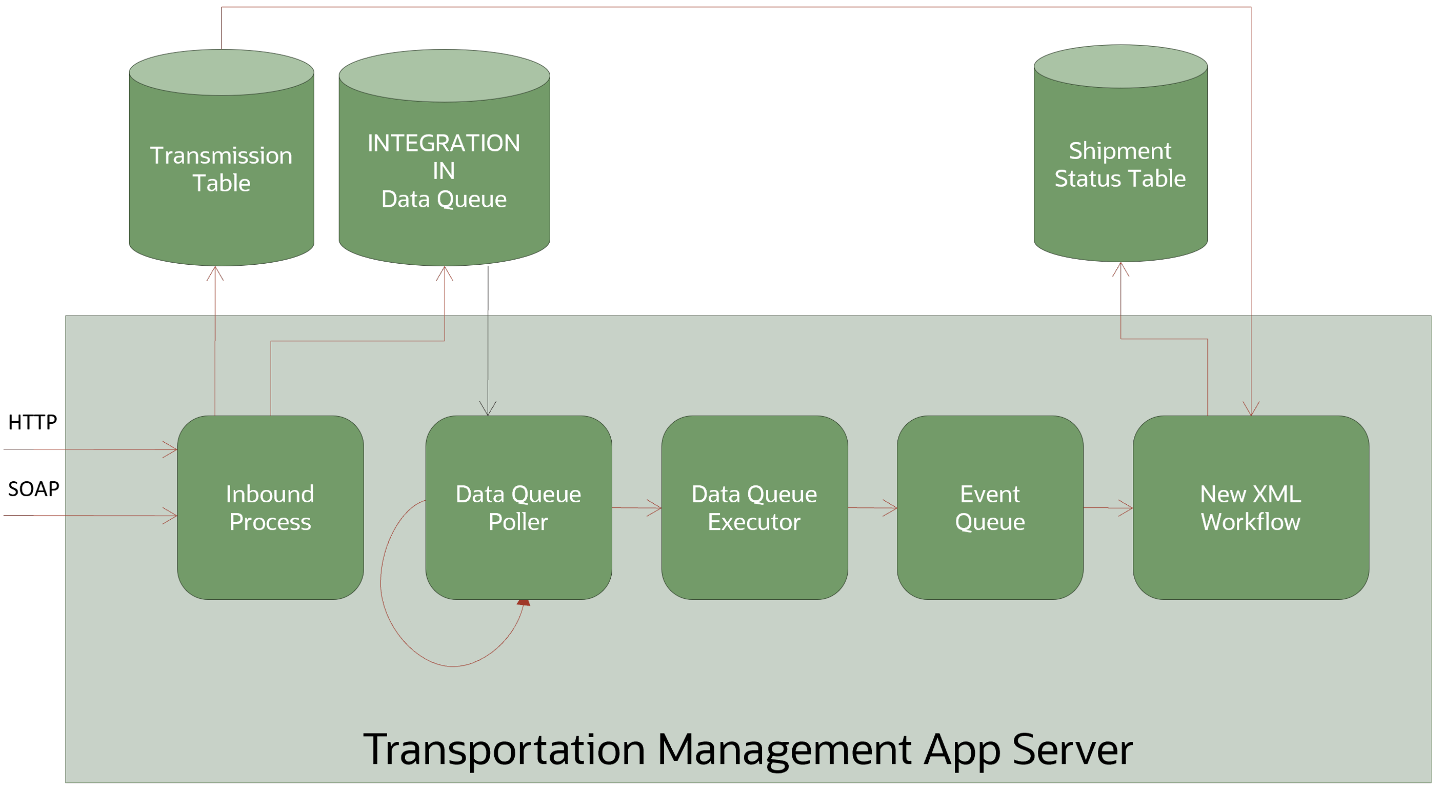
1. The XML is stored in the Transmission table with a new unique Transmission number.
2. A New XML topic containing the Transmission number is placed on the in-memory event queue.
3. A listener thread which has subscribed to New XML topics removes the event and passes it to the New XML Workflow to process.
4. Workflow processes the content of the XML, which in this case results in a new status being assigned to a shipment.



Imagine a Transportation and Global Trade Management Cloud implementation scenario that has an inbound interface with a carrier status application where shipment statuses are sent in one batch of 10000 to Transportation and Global Trade Management Cloud. Without Data Queues, the application server in-memory event queues would contain all 10000 Shipment Status transactions in one backlog, therefore competing with all other internal application server processes for resources and process time.

The following figure shows the application server processing where data queues were configured for inbound XML messages. The Inbound Data Queue process can instead be configured to stage the initial 10000 transactions and process in batches of 1000 every 5 minutes.

1. The XML is stored in the Transmission table and the associated Transmission number is stored in a Data Queue Event.
2. At some point in time within the configured interval, a Poller thread will retrieve up to 1000 data queue events and pass individually to the Executor to be processed.
3. The Executor publishes a New XML topic which, from then on, will be processed as with normal in-memory event queues.



The application server processing for Outbound XML messages also uses in-memory event queues to manage workload; one for the building of the XML and another for the physical transport of the message. The use of Data Queues for outbound message works essentially identical to the inbound scenario described above i.e. instead of publishing new build or transport topics on in-memory event queues, data queue events are stored in outbound data queues. The same arrangement of Poller and Executor then publishes the topic to the next step in the process.

Activating Integration Data Queues

There are several default Integration Data Queues available in the PUBLIC domain:

- **INTEGRATION IN**: Used to manage all Inbound Transmissions sent in via UI (Upload Transmission), HTTP (WMServlet), and SERVICE (TransmissionService).
- **INTEGRATION OUT XML BUILD**: Used to manage outbound process of generating Transmission content from database objects and staging in Transmission tables ready for transport.
- **INTEGRATION OUT TRANSPORT – HTTP**: Used to manage transmissions ready to be transported via HTTP to external system.
- **INTEGRATION OUT TRANSPORT – SERVICE**: Used to manage transmissions ready to be transported via web service call to external system.

The inbound data queue events reside in the Q_INTEGRATION_IN Data Queue table. The outbound data queue events reside in the Q_INTEGRATION_OUT Data Queue table.

By default, the PUBLIC data queues are inactive, except for the INTEGRATION IN data queue which is active. To activate the other queues, login as DBA.ADMIN and edit the Data Queue records via the Data Queue Manager (under **Business Process Automation > Integration** menu).

Alternatively, the PUBLIC records can be copied to DOMAIN specific records and activated there. In order for the application server to recognize the new queue names as the default data queues, the following properties must also be set:

```
glog.integration.dataqueue.inbound
glog.integration.dataqueue.xmlBuild
glog.integration.dataqueue.transport.http
glog.integration.dataqueue.transport.service
```

For example,

```
glog.integration.dataqueue.inbound=MYDOMAIN.MYINBOUNDQ
```

> **Note:** An application server can only process one default data queue of each type. For example, it is not possible to configure two inbound default data queues: MYDOMAIN.INBOUNDQ1 & MYDOMAIN.INBOUNDQ2 or MYDOMAIN2.INBOUNDQ1.

Although only a single default inbound data queue can be specified, it is possible to configure as many additional inbound data queues as required. Additional data queues are useful for scenarios where different inbound transactions require different prioritization or throughput management. For example, it may be desirable to process inbound Tracking Event transactions on a data queue with more threads or that while Order Release creation and update must be regarded as top priority, other lower priority transactions must still be processed.

It is now possible to specify a data queue GID and/or event priority in the Transmission header for each individual Transmission to override the default inbound data queue. Consult the Transmission schema documentation for details on how to specify these overrides.

Using the earlier example where Order Releases are top priority, but others must still be processed, it should be clear from the following discussion that multiple data queues would best be used in this scenario.

The priority value for an event corresponds to the `q_preemption_priority` column in the data queue. The poller configured for the data queue can then use this column in the 'order by' clause for retrieving batches of events to process. The INTEGRATION IN poller is an example of such a poller. For the single data queue scenario where orders were higher priority than all other transactions, if there were 100 events in the queue – 60 order releases and 40 tracking events – and the data queue batch size was 50, then only the first 50 order releases would be selected for processing. If, before the poller attempts to poll a new batch another 50 order release transmissions arrive, then these would be selected for the next batch thus meaning that the processing of tracking events message is delayed longer than might be preferable. However, with different data queues each queue can process events in parallel.

The inbound Order Release Transmissions should specify the data queue used to handle orders and the other Transmissions should specify a different data queue and possibly also a priority value to further control the order of processing in this data queue. Obviously, there are many other possible configurations of batch sizes, poller frequency and threads counts that can handle similar scenarios. The addition of support for multiple data queues expands these options considerably.

Once one or more Integration Data Queues are active, the processing can be controlled on a domain-by-domain basis by setting one of the following PARAMETER SET parameters to 'TRUE' on the DOMAIN PARAMETER SET.

- DATA QUEUES - USE INBOUND DATA QUEUE
- DATA QUEUES – USE QUEUE FOR XML BUILD
- DATA QUEUES – USE QUEUE FOR TRANSPORT – HTTP

- DATA QUEUES – USE QUEUE FOR TRANSPORT – SERVICE

If either the queue is not active or the domain parameter is FALSE, the default in-memory queues will be used.

> **Note:** It is possible to override this default to process via in-memory queues. Set property glog.integration.dataqueue.inbound.useMemoryQueue=false. This will cause the inbound Transmissions to remain in 'STAGED' status.

Understanding the Integration Data Queue Definitions

The standard installation of Transportation and Global Trade Management version 6.2 has PUBLIC Data Queue Definitions configured for each of the Integration Data Queues described in the preceding section.

This section describes these definitions so that any configuration can be done without any unforeseen impact on the integrity or performance of the Transportation and Global Trade Management Cloud servers.

The Data Queue Definition is a unique (based on a Data Queue Definition GID) configuration of the following properties.

Data Queue Definition Properties

Parameter	Description
Active	Only queues which are marked as 'Active' will have Poller threads started on the application server.
Thread Count	The number of Poller threads that will be started on each application server. All Poller threads will synchronize their access to the events on a data queue. One thread at a time will retrieve a batch of events, mark them all as 'ACTIVE' and commit the transaction before releasing the queue for another thread to access. **This would normally be tuned to match the expected volume of events.**
Batch Size	The maximum number of events that will be retrieved by each Poller thread. **This would normally be tuned to match the expected volume of events.**
Polling Frequency	The duration each Poller thread will sleep between attempts to retrieve a batch of events. **Note:** if the time taken to process a batch exceeds the polling frequency the Poller thread will essentially start processing the next batch more or less immediately after completing the current batch. **This would normally be tuned to match the expected volume of events.**
Data Queue Table	Corresponds to the physical table which will contain the events **This should not require customization.**
Data Queue Poller	The Poller configuration specifies the behavior and attributes of the SQL statement used to retrieve the batch of events. In addition to the Class that will run the query, the following can be configured:- • Filter – these are additional predicates added to the basic statement (e.g. status = 'QUEUED' etc.). For example, if a domain specific queue is in use it may be necessary to add a predicate which selects the q_queue_name column which matches the full GID.

Parameter	Description
	• Order – this is specified as the ORDER BY criteria. This is how priority ordering is achieved. Poller Definitions are configured via Business Process Automation Power Data. **Custom Poller definitions may be required if additional criteria is used to further partition events placed on the same queue e.g. by joining other tables.**
Data Queue Executor	The Executor configuration species the Class that will process the actual event and whether events in the ACTIVE or FAILED state should be reprocessed on app server startup. Executor Definitions are configured via Business Process Automation Power Data. **Custom Executor definitions may be required if reprocessing of events needs to be handled differently.**
Finder Set	This specifies the Class which formats the queue specific information in the Data Queue Manager Events UI. **This should not require customization.**
Related Queue	This is used to manage the parent/child relationship between events. **This is not used by Integration Data Queues.**

Configuring Inbound Data Queue Definitions

There is one inbound data queue definitions created by default in the PUBLIC domain: INTEGRATION IN. Events are held in the Q_INTEGRATION _IN Data Queue Table. By default, the queue is inactive.

If these base configurations are deemed to be sufficient for the expected volumes, they can be activated by logging in as DBA.ADMIN and setting each queue definition to 'Active'. Additionally, as mentioned previously, the INTEGRATION IN Parameter set value also needs to be set to 'TRUE'.

Configured Poller Definition

The Poller Definitions for both queues are configured to use the PREEMPTIVE POLLER plug-in, which uses a 'Top N query' format to retrieve an ordered batch of events. The format of the SQL statement that would be used to retrieve a batch of events would essentially be as follows:

```
SELECT rownum, {column list}
 FROM (SELECT {column list}
 FROM Q_INTEGRATION_IN
 WHERE q_state='QUEUED' and {filter}
 ORDER BY {order})
 WHERE rownum <= {batch size}
```

The arguments in braces are replaced at runtime as follows:

Poller Definition Parameters

Parameter	Description
column list	The complete column list is controlled via the Event Class which for Q_INTEGRATION_IN events would be:- q_event_seq, rownum, q_start, q_process_start, domain_name, q_preemption_priority, topic_class_name, topic_parameters, topic_description, i_transmission_no, q_queue_name.
filter	The PUBLIC Poller definitions for INTEGRATION IN has a filter clause as follows:- q_start < sys_extract_utc(current_timestamp) and q_queue_name = 'INTEGRATION IN' Therefore, if a customer-defined Data Queue Definition is to be used there will need to be a corresponding customer-defined Poller Definition in order to override the q_queue_name parameter.
order	For Preemptive Data Queues the base ORDER BY clause will be as follows:- q_preemption_priority, q_event_seq That is, ordered by priority first and then event sequence number.
batch size	This is the batch size taken from the Data Queue Definition.

Configuring Outbound Data Queue Definitions

The Outbound data queue definitions are subjected to the same configuration capabilities and constraints as the inbound data queues.

Monitoring Integration Data Queues

In addition to its primary task of administering the data queues themselves, the Data Queue Manager is also used to monitor the activity of events on each queue via the Events action button. This provides access to a Finder/Results screen from where all events for a queue can be listed and individual event details can be viewed. The majority of information is generic to all Data Queues e.g. Process Time, and Log Process ID. The following sections describe the Integration Data Queue specific data.

Inbound Events

All inbound events have an associated transmission number. In normal processing, when a transmission is sent to Transportation and Global Trade Management Cloud, the XML is persisted in the transmission tables with an initial status of STAGED and a New XML event is published almost immediately. Once the New XML event processing has started the status is changed to FRESH. When the Inbound Data Queue is active, the XML is persisted in the transmission tables with a status of STAGED and the New XML event is persisted in the data queue table. Consequently, transmissions can remain in this STAGED status for much longer.

Inbound events can also be preempted. In other words, events already present in the data queue can be selected for processing ahead of other events that may have been inserted before it. By selecting an event for preemption, the Q_PREEMPTION_PRIORITY is set to the lowest value in the data queue table. This will ensure the next POLLER will retrieve the selected event.

Outbound Events

Outbound events may go through two event queues: XML Build and Transport. The XML Build process happens before a transmission record has been created and so these events do not have an available transmission number. They do, however, have a Notification Context that shows the object type of the communication, such as Location, and one or more object GIDs.

The display of the context is limited to approximately 4000 characters and so if the context is greater than this it will not be displayed.

As with inbound events, outbound events can be preempted.

However, unlike inbound events, outbound event priority can be set when the data queue event is staged in the data queue table. The priority value is managed via the External System where it can be set alongside an OUT XML PROFILE for a particular interface Element e.g. PlannedShipment.

14 Descartes Integration

Overview

The Global Trade Management component of Transportation and Global Trade Management Cloud supports an interface to Descartes for retrieving reference data used in the services. This section details the setup for accessing the data.

Descartes Registration

Registration with Descartes is required for access to the Descartes data for Global Trade Management. The registration process will give a site ID and password to the client that can be entered in the setup of Global Trade Management. This user and password combination will be used to download the actual data.

Setup in Global Trade Management

Global Trade Management Properties Setup

Network Connectivity to Descartes Server

If the network Global Trade Management is installed requires a proxy server for external access to the Descartes server, specify the following properties in the glog.properties:

```
glog.integration.http.proxyHost=proxy-address
glog.integration.http.proxyPort=proxy-port
```

See the *glog.integration Properties help* for details.

Properties for Data Loading Directories

The data loader relies on pre-defined directories for the source and output directories. The directories are defined via properties:

```
gtm.dataload.basedir=$temp.dir$/dataload
gtm.dataload.inputdir=$gtm.dataload.basedir$/input
gtm.dataload.outputdir=$gtm.dataload.basedir$/output
gtm.dataload.workingdir=$gtm.dataload.basedir$/output
```

These properties do not need to be changed for implementation. The logic will create the directories needed if they are not already created.

Setup of Descartes User Credentials

After retrieval of the user credentials from Descartes, configure the credentials in Global Trade Management as follows:

- Setup an External System record in Global Trade Management with the following details:
  - **User Name**: User ID provided during registration.
  - **Password**: Password provided during registration.
  - **Password (Confirm)**: Same value for password.
  - **URL (in For HTTP/HTTPS section)**: URL for accessing the Descartes status document. The URL may resemble the following: https://gtm.content.descartes.com/status/statusdocument.xml.
- Update the Content Source to refer to the newly created External System record as follows:
  - Access menu: **Trade Master Data > Power Data > Data Loading > Content Source**
  - Edit the Content Source with ID which has been predefined for Descartes
  - Add an entry in the Content Source Config section at the bottom as follows:
    - **ID**: Specify an ID (e.g. CI_GTM_DATA).
    - **Comm Method**: Specify "HTTPPOST".
    - **External System**: Select the external system created above.
    - **Active**: Select to enable the configuration.

Processing Global Trade Management Content

The actions to initiate the Global Trade Management content loading are defined with process control. This is accessible from the following menu:

Trade Master Data > Process Management

The following processes are available:

- **Download Data Content**: Used to retrieve the content from a Global Trade Management content source. For Descartes, this process will download the latest files from the Descartes server into the local Global Trade Management server.
- **Purge Data Content**: This is used to cleanup up the files on the Global Trade Management server used for the data loading.

15 Global Trade Management Screening Service

Screening Services

This section captures the screening services developed in Global Trade Management.

All interactions with the Global Trade Management Services through integration is made using the Service Request element (refer to the GTM.xsd schema for additional details). Upon receiving the ServiceRequest XML in Global Trade Management, the request is processed synchronously and a ServiceResponse xml is generated as a response to the caller. Note that the Transmission/TransmissionHeader/TransmissionType must be set to "SERVICE" for the synchronous processing to occur.

The ServiceRequest element contains several options for the screening services. A few of the available services are as follows:

Example ServiceRequest Element Options for Screening Services

ElementName (Service)	Description
RestrictedParty	Used to indicate if the specified party is on the restricted parties list.
SanctionedTerritory	Used for transactional screening of territories. The screening uses the rules engine and runs all rules that have a control category of SANCTION. Users specify any number of territories and qualifiers in the LocationInfo element.
Classification	Used for transactional searching for global classification based on product description.
ComplianceRule	Used for transactional screening of personal information, product and usage information, location information, and user defined conditions.
LicenseDetermination	Used to find matching licenses for a license control, based on party, location and user defined code data.

See the GTM.xsd schema file for additional fields in the ServiceRequest and ServiceResponse.

16 Mobile Device Communications

Message Schema

The Mobile Message infrastructure is used for mobile communication messages sent inbound and outbound to Transportation and Global Trade Management Cloud. These messages are intended for communication to a mobile communication device or sensor. When sent inbound to Transportation and Global Trade Management Cloud, these messages should be small. The message body can contain text-formatted content, which may be parsed and converted to the Transmission XML when transmission processing is required.

Schema Files

The following schema files contain all definitions for the Message schema:

Schema Files

Schema Name	Description
Message	Definitions for all Message XML Primary Documents.
Message Content Source	Definitions for Message Group implementations for the Content Source infrastructure.

XML Namespaces

XML Namspaces

XSD Schema File	Namespace
Message.xsd	http://xmlns.oracle.com/apps/otm/msg/v1
MessageContentSource.xsd	http://xmlnx.oracle.com/apps/otm/cs/msg/v1

Primary XML Documents

The following are the primary XML documents in the Message XML schema that are used inbound and outbound to/ from Transportation and Global Trade Management Cloud:

- **Message**: The Message is the primary document used for inbound to and outbound for mobile communication messages. Each message can contain either text based or XML-based content in the message body.

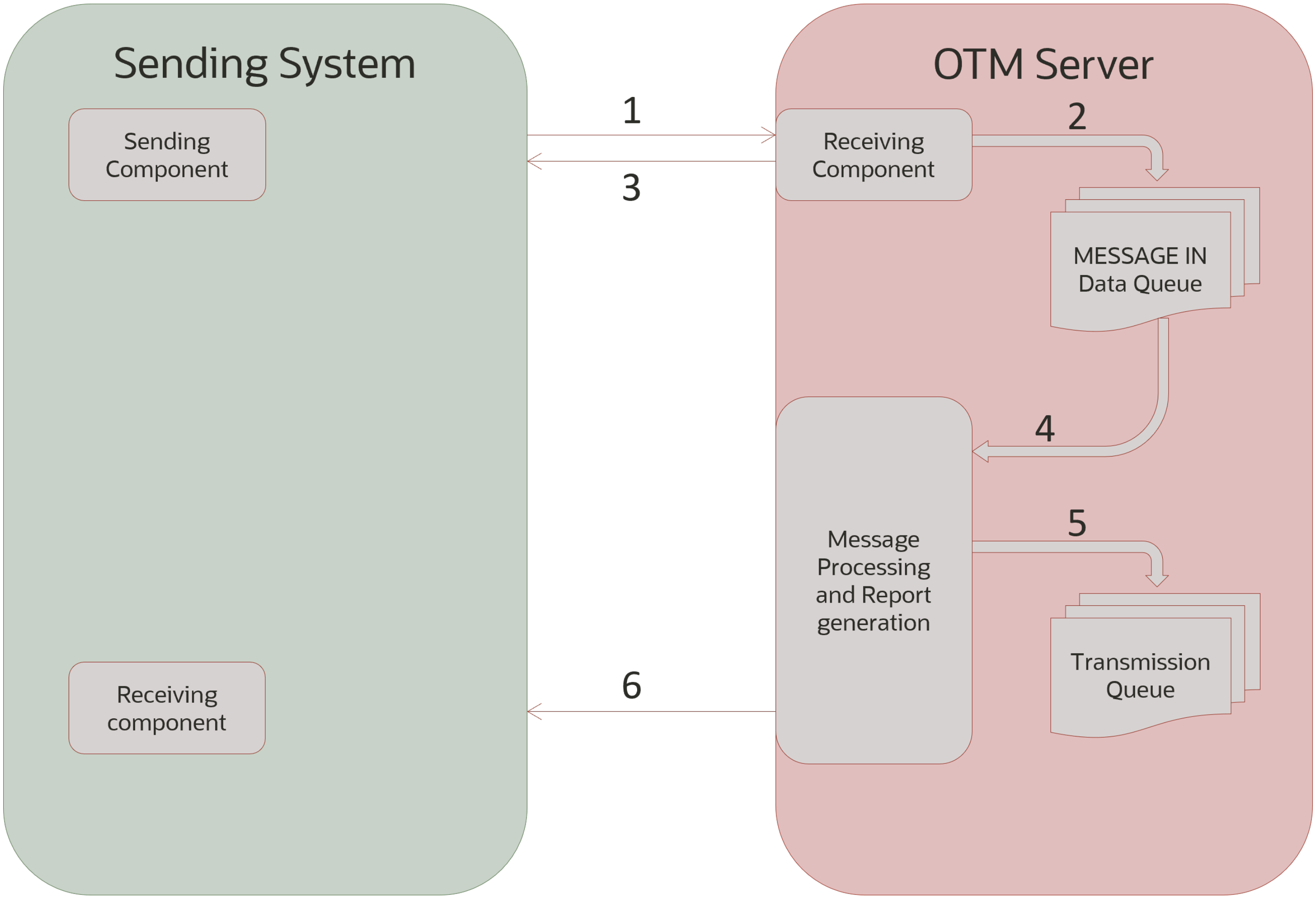
- **MessageAck**: The MessageAck is the response message to the receipt of the message. It contains the confirmation for the receipt of the message with an assigned ReferenceMessageId element, or an error if the message was not correctly received.
- **MessageReport**: The MessageReport summarizes the errors that were detected during the processing of the message if the Message Body contains 'actionable' content. For example the content could describe a Tracking Event which is translated into a Transmission XML and processed as an Integration message. The report is optionally sent after all the transactions in the Transmission have been completed (successfully processed or generated errors). The request for the MessageReport is indicated in the message in the AckSpec element in the MessageHeader. The MessageReportBody in the MessageReport will include the TransmissionReport that will summarize the errors that were detected during the processing of the Transmission.

Each of these documents is detailed in the Message XML schema file.

In the case where the message is correctly parsed and converted to a Transmission XML for inbound transmission processing, the Transmission Report XML may also be sent as a summary of processing if the AckSpec is specified in the generated Transmission resulting from the parsing.

Application Level Protocol

The following diagram shows when the documents are sent for an inbound Message XML scenario into Transportation and Global Trade Management Cloud.



1. Message XML document is sent to Transportation and Global Trade Management Cloud. The nature of the sending component depends on the transport protocol used e.g. it will be an HTTP client if sending message via HTTP POST.
2. Mobile message is persisted generating a unique Message Reference ID in the process. A data queue event is queued to the MESSAGE IN data queue for later processing of the message content.
3. A MessageAck document is returned which will contain the Message Reference ID. The nature of the response depends on the transport protocol used e.g. if using HTTP it will be a synchronous response to the initial HTTP POST.
4. The MESSAGE IN data queue event is polled and run.
5. If the Message Body contains actionable content a Transmission XML will be staged for processing. The background Transmission processing daemon retrieves the Transmission at some time later and processes it e.g. creates or updates the business object using the original XML message content.

6. A MessageReport is optionally sent to confirm the status (PROCESSED or ERROR) of the actionable content. Again the report message contains the Message Reference ID for correlation.

Processing Overview

If the servlet receives a Message XML, the integration module does the following:

1. Extracts authentication information from the MessageHeader. This information can be in the HTTP header instead.
2. Validates username and password.
3. Stores an entry in the I_message table, and the content in the I_transmission table.
4. Sends a MessageAck back to the sender as the synchronous response.
5. Checks for presence of MessageTypeGid and StylesheetProfileGid. If neither are present, checks for the MessageProfileGid, and uses the StylesheetProfileGids specified in the Message Profile to determine the correct Stylesheet Profile.
6. Using the Stylesheet Profile, the message is parsed to extract designated fields and updates the message table with extracted fields as needed.
7. Using the Message Type specified, the module performs the following:
   a. Notifies Message Center recipients specified on the Message Type
   b. Associates or disassociates devices to drivers, pieces of equipment, and power units as indicated on the Message Type
   c. Generates a Transmission XML if indicated on the Message Type. The Transmission XML would then be processed as indicate in the Transmission Processing section above.
8. Sends a MessageReport with validation and processing errors; Depending on your property settings, Transportation and Global Trade Management Cloud might only send a MessageReport if there are errors.

Message Status

Each message sent into Transportation and Global Trade Management Cloud has a status field that indicates the state of the message. In addition, the content of the message is stored in the transmission table which also has its associated status. You can view the status for a message in the Message Hub Manager UI that can be accessed via the following menu: **Business Process Automation > Integration > Message Hub Manager**. The status of the message, and its related transmission, could be one of the following:

Message Statuses

Message Status	Related Transmission Status	Description
RECEIVED	MESSAGE	Indicates that the message is initially received in the message table, with the content stored in the transmission table.
PROCESSING	MESSAGE or (dependent on Transmission processing)	Indicates the message is being processed. If the Message Type for the received message indicates a Transmission is to be generated and processed, then the status of the Transmission will be initially changed to FRESH and will be handled by normal transmission processing.

Message Status	Related Transmission Status	Description
PROCESSED	MESSAGE or (dependent on Transmission processing)	Indicates that the message successfully completed processing.
ERROR	MESSAGE or (dependent on Transmission processing)	Indicates that the message had completed processing and there were errors in the processing.

Web Service Integration

Mobile Device Communication messages can be sent inbound to Transportation and Global Trade Management Cloud by calling the MessageService Web Service. The service is a SOAP Document Literal BARE style service.

The service has the following operations:

- `receiveMessage`: Accepts Message XML for staging and processing
- `receiveMessageAck`: Accepts acknowledgment MessageAck XML in response to an outbound Message XML

By default the WS-Security policy requires a Username Token and to be transported over HTTPS.

HTTP Integration

Mobile Device Communication messages can be sent inbound to Transportation and Global Trade Management Cloud by sending via HTTP POST to the glog.integration.servlet.WMServlet. The servlet accepts both a Message and MessageAck XML document and returns a corresponding MessageAck in both cases.

17 Transactional Data Pre-Loading

DirLoad

You can get Transportation and Global Trade Management Cloud to load your inbound transmissions into the database faster without involving the application server. This is good when you just want to pre load data into Transportation and Global Trade Management Cloud and process the data later, like during setup of Transportation and Global Trade Management Cloud.

DirLoadServlet only supports these interfaces:

- TransOrder
- Shipment
- ShipmentLink
- TenderResponse
- Location
- Item
- ItemMaster
- HazmatGeneric
- HazmatItem
- ShipmentStatus
- Invoice
- Release
- ShipmentGroup
- SShipUnit
- Sku
- SkuTransaction
- Contact
- TransOrderStatus

> **Note:** The DirLoadServlet does not raise lifetime events (like shipment - created for ActualShipment) so automation agents cannot be triggered. Therefore, care must be taken to ensure that any objects created via DirLoadServlet will be complete and in the expected state.

To do this:

1. Make sure your transmissions only use the transaction code I. It is possible to use other transaction codes but with the limitation that you need to make sure that no user accesses that data through the application server while you update/delete your data. If a user accesses the data, you need to restart your application server after uploading your data to refresh its caches. To use other transaction codes with the DirLoadServlet you need to enable the relevant `glog.integration.dirload.support.<code>` property, where `<code>` is one of:

- update for "U"
- insertUpdate for "IU"
- delete for "D"

Setting the property value to "on" will enable that transaction code.

2. Post XML transmissions to `http://hostname/GC3/glog.integration.servlet.DirLoadServlet`
3. DirLoadServlet saves your data to the database.

DirLoadServlet sets default statuses for business objects you insert.

18 Workflow Web Service

Agent Workflow Triggers

There are several web services which provide access to trigger agent workflow for selected business objects. The capabilities include:

- Execution of individual agent action against one or more objects of a supported type.
- Execution of multiple agent actions against one or more objects of a supported type.

Currently the supported object types are SHIPMENT (Buy Shipment), SELL SIDE SHIPMENT, ORDER RELEASE, DRIVER and ORDER MOVEMENT.

There is a separate service for each business object type:

- ShipmentService
- SellSideShipmentService
- OrderReleaseService
- OrderMovementService
- DriverService
- AgentService

The type specific services handle individual agent actions for that type and have one operation: `processAction`. The AgentService handles processing of multiple actions essentially identical to a Transportation and Global Trade Management Cloud Agent but with some constraints (covered later) and also has one operation: `processAgent`.

All services are implemented with the synchronous REQUEST/RESPONSE messaging model. However, with respect to Transportation and Global Trade Management Cloud agent action processing, the response indicates that the action has been scheduled successfully. This is due to the fact that the Transportation and Global Trade Management Cloud application could have a significant amount of workflow triggered by such an action. Therefore waiting for completion may require an excessive transaction timeout value.

The input and output messages for each service are specified in the service XSDs. Namely:

- AgentService.xsd
- ShipmentService.xsd
- SellSideShipmentService.xsd
- OrderReleaseService.xsd
- OrderMovementService.xsd

Agent specific message content is specified in the following XSDs:

- **Agent.xsd**: contains agent header details
- **ShipmentAction.xsd**: contains all actions related to buy shipments
- **SellSideShipmentAction.xsd**: contains all actions related to sell shipments
- **OrderReleaseAction.xsd**: contains all actions related to order releases
- **OrderMovementAction.xsd**: contains all actions related to order movements

> **Note:** Due to a change in V6.2 in how these services are deployed in the application server at runtime, the WSDL for each service will not directly reference the XSD files mentioned above. However, the formats defined in these XSD files will be functionally equivalent to the definitions referenced in the WSDL.

AgentService Constraints

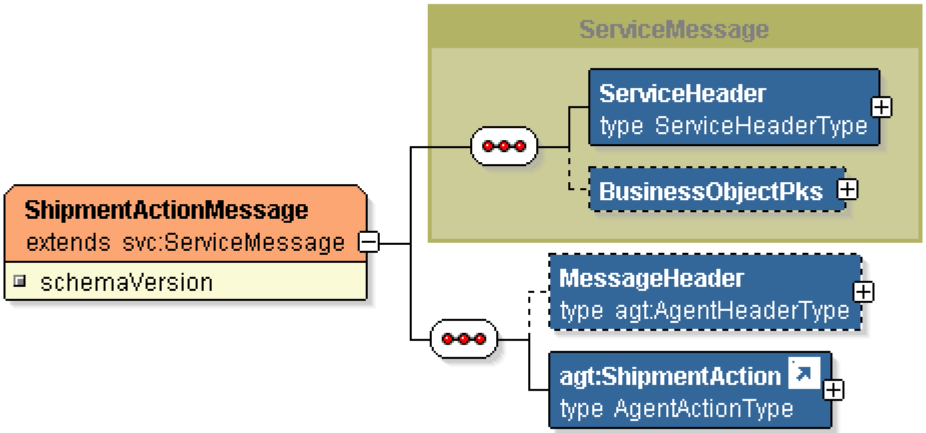
Although in theory a complete agent could be defined using the schemas, some actions will not be supported in the initial version. These actions are known as the BLOCK actions: IF, ELSE, FOR EACH, etc.

Version Control

The service and message definition schemas are under version control starting from version 1.0, i.e. major version number is 1, and minor version number is 0. The current target namespace for each schema will contain the major version number, for example the target namespace for version 1.0 of Agent.xsd is:

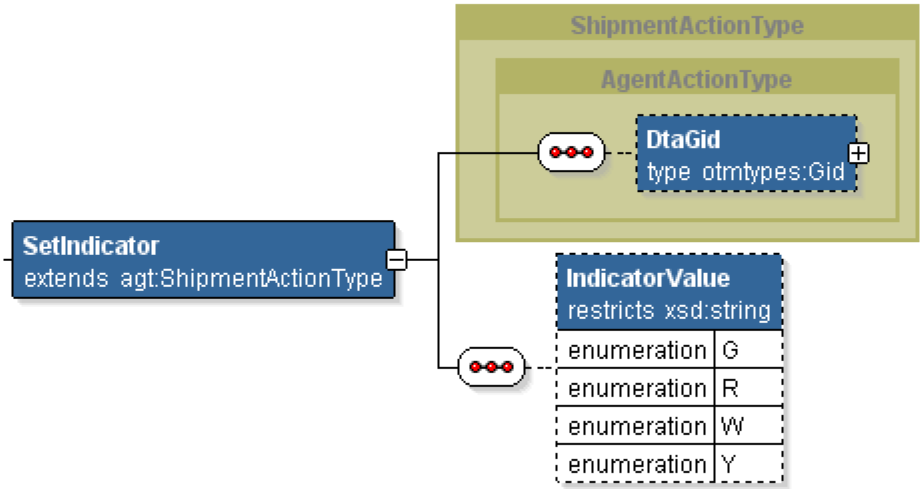
```
http://xmlns.oracle.com/apps/otm/agent/v1
```

Example Service Message



All service messages extend the ServiceMessage complexType (defined in the Service.xsd schema).

The Agent Action messages then include Agent Header information (MessageHeader) followed by an Agent Action, in this example ShipmentAction. All valid Shipment Actions are defined in the ShipmentAction.xsd schema. The following example shows the Set Indicator Action:



The SetIndicator element is allowed to be substituted for the ShipmentAction element because it is defined as an XSD **substitutionGroup** and extends the same `AgentActionType` complexType.

Changes between Version 1.0 and Version 2.0

Optional Username and Password Elements

The initial version of the Service XSD schema had the username and password element as "required". Due to the added support for WS-Security this is now an unnecessary restriction.

Removal of "Fields" Element

The agent action element part of the message e.g. "SetIndicator", in version 1.0 required the action fields to be wrapped in an outer Fields element. Due to improvements in the underlying web service standards supported (e.g. JAXB) this is now not required resulting in a smaller message size.

19 Appendix: Integration Messages

Integration Messages

This appendix lists integration messages, describes why the message occurs, and describes what you need to do as a result of receiving the message.

You might find these error messages in a TransmissionReport element.

Integration Message

Heading	Data
Message:	THE DATE ELEMENT {0} WITH VALUE {1} IS NOT OF FORMAT YYYYMMDDHHMMSS
Occurs When:	An invalid date format error occurs when the date is not provided in the format YYYYMMDDHHMMSS.
Corrective Action:	Enter the date using the required format.

Integration Message

Heading	Data
Message:	DATA CONVERSION FAILED FOR THE ELEMENT {0} WITH VALUE {1}
Occurs When:	A data conversion error occurs when character data cannot be converted to an internal data type.
Corrective Action:	Eliminate extraneous characters from the data element.

Integration Message

Heading	Data
Message:	THE ELEMENT(S) {0} WITH VALUE(S) {1} IS A DUPLICATE PRIMARY KEY
Occurs When:	A duplicate Key error occurs when the primary key for a given element already exists in the GLog database.
Corrective Action:	Change the Transaction Code element from I to IU.

Integration Message

Heading	Data
Message:	THE DATE ELEMENT {0} WITH VALUE {1} IS NOT OF FORMAT YYYYMMDDHHMMSS

Heading	Data
Occurs When:	A PKNotFoundError most often occurs when an attempt is made to update or delete data that does not exist in the GLog database.
Corrective Action:	If your Transaction Code is U, then use UI instead. If your Transaction Code is D, then there is no corrective action.

Integration Message

Heading	Data
Message:	THE FOREIGN KEY ELEMENT {0} WITH VALUE {1} COULD NOT BE FOUND IN TABLE {2} COLUMN {3}
Occurs When:	A FK Not Found Error occurs when a referenced primary key does not exist in the GLog database.
Corrective Action:	Correct the XML data value such that it refers to a primary key that does exist in the GLog database.

Integration Message

Heading	Data
Message:	THE REQUIRED ELEMENT {0} IS MISSING
Occurs When:	A missing required element error occurs when a required element has been omitted from a GLogXMLElement.
Corrective Action:	Provide the missing required element in your XML data.

Integration Message

Heading	Data
Message:	THE I_TRANSACTION_NO WITH VALUE {0} DOESN'T EXIST IN THE DATABASE
Occurs When:	An ITransactionNoNotFoundError occurs when a GLogXMLElement, such as a Tender Response, refers to a transaction number that does not exist in the G-Log I_TRANSACTION table.
Corrective Action:	Refer to an existing transaction number.

Integration Message

Heading	Data
Message:	THE TRANSACTION CODE {0} is not valid. Valid codes are I,U,D,IU,UI,NP
Occurs When:	An invalid transaction code error occurs when the Transaction Code element is specified with an invalid value.
Corrective Action:	Specify a valid Transaction Code in your XML data.

Integration Message

Heading	Data
Message:	THE TRANSACTION CODE {0} IS NOT SUPPORTED IN THIS INTERFACE. VALID CODES ARE {1}
Occurs When:	A transactionCodeNotSupported occurs when the TransactionCode element is specified with an unsupported value.
Corrective Action:	Correct the XML data to specify a valid TransactionCode.

Integration Message

Heading	Data
Message:	THE ELEMENT {0} AND THE ELEMENT {1} CANNOT BOTH BE SPECIFIED
Occurs When:	A conflicting element error occurs when two elements have been provided in a G-Log XML Element, when only one out of the two may be used.
Corrective Action:	Eliminate one of the two conflicting elements in your XML data.

Integration Message

Heading	Data
Message:	THE NUMBER ELEMENT {0} WITH VALUE {1} IS NOT IN A NUMBER FORMAT
Occurs When:	An invalid number format error occurs when non-numeric characters are specified in a numeric element.
Corrective Action:	Eliminate the non-numeric characters in your XML data.

Integration Message

Heading	Data
Message:	WE REQUIRE A {0} ELEMENT WITH VALUE {1} IN THE {2} ELEMENT
Occurs When:	A missing element error occurs when an element with a particular value is required.
Corrective Action:	Provide the element with the required value as indicated in the error message.

Integration Message

Heading	Data
Message:	THE BOOLEAN ELEMENT {0} WITH VALUE {1} MUST BE EITHER Y or N
Occurs When:	An invalid Boolean error occurs when a Boolean element is provided with a value other than Y or N.
Corrective Action:	Provide either Y or N in the indicated Boolean element in your XML data.Table 20-14

Integration Message

Heading	Data
Message:	THE ACTION CODE {0} is not valid. Valid codes are A, D
Occurs When:	An invalid action code error occurs when an action code is specified with a value other than A or D.
Corrective Action:	Provide either A or D in your XML data.

Integration Message

Heading	Data
Message:	THE ELEMENT {0} WITH VALUE {1} is not valid. Valid codes are {2}
Occurs When:	An invalidCodeError occurs when a code is specified that is not valid for that element
Corrective Action:	Correct the XML data to provide a valid value.

Integration Message

Heading	Data
Message:	THE ACTIVITY {0} is not valid. Valid codes are D, P or O
Occurs When:	An invalidActivityError occurs when an activity is specified with a value other than P, D or O.
Corrective Action:	Correct the XML data to provide anyone of D, P or O.

Integration Message

Heading	Data
Message:	CAUGHT THE FOLLOWING EXCEPTION WHILE PROCESSING TRANSACTION: {0}
Occurs When:	A transactionProcessorException occurs when the integration layer validation method has not checked for a given error condition, and the condition is caught by the underlying database. Most often this error occurs when an attempt is made to update the Transportation and Global Trade Management Cloud database so that one or more database referential integrity constraints are violated. For example, this error occurs if you attempt to delete a transportation order after one or more releases have been created.
Corrective Action:	No particular correction can be defined. Analyze the error based on the requirements of the data in the underlying database.

Integration Message

Heading	Data
Message:	ELEMENT {0} VALUE {1} HAS LENGTH {2} WHICH EXCEEDS THE MAX LENGTH {3} OF TABLE {4} COLUMN {5}
Occurs When:	The length of an element value exceeds the length of the corresponding database column
Corrective Action:	Correct the XML data to provide a value which does not exceed the maximum length.

Integration Message

Heading	Data
Message:	ELEMENT {0} VALUE {1} EXCEEDS THE MAX LENGTH {2}
Occurs When:	The length of an element value exceeds the length of the corresponding database column
Corrective Action:	Correct the XML data to provide a value which does not exceed the maximum length.

Integration Message

Heading	Data
Message:	ERROR VALIDATING MAX LENGTH OF ELEMENT {0} VALUE {1} - CHECK TABLE NAME {2} COLUMN NAME {3}
Occurs When:	The specified table/column information could not be found.
Corrective Action:	Call Support.

Integration Message

Heading	Data
Message:	TRANSACTION NUMBER {0} APPLICATION {1} PRIMARY KEY {2} TRANSACTION CODE {3} SUCCESSFULLY PROCESSED
Occurs When:	A transaction success informational message occurs when a transaction has been successfully processed.
Corrective Action:	No action needed.

Integration Message

Heading	Data
Message:	UNABLE TO PROCESS DUE TO MULTIPLE SHIPMENTS MATCHED ON {0}

Heading	Data
Occurs When:	A matchMultipleShipment error message occurs when ShipmentRefnums/EquipmentNumber match different Shipment_GID. This type of error happens when receiving a TenderResponse or a ShipmentStatus.
Corrective Action:	Correct the XML data to provide ShipmentRefnum values which correspond to the same Shipment.

Integration Message

Heading	Data
Message:	WE REQUIRE A {0} ELEMENT WITH A VALUE MATCHING A {1} IN THE {2} TABLE
Occurs When:	A missingRequiredDataError occurs when an element with a particular value is missing from the database table.
Corrective Action:	Correct the database data to provide the value as indicated in the error message.

Integration Message

Heading	Data
Message:	ORDER {0} IS NOT MODIFIABLE AND SO COULD NOT BE MODIFIED OR DELETED.
Occurs When:	An orderNotModifiable informational message occurs when the status on an order is not "WKFLW_ORDER_OB_MODIFIABLE". This can occur if an order is in a state that restricts it from being modified, or an agent is setup to restrict modification.
Corrective Action:	If you want to be able to modify the order, you may have to change the state of the order or modify the agent that handles order modifications.

Integration Message

Heading	Data
Message:	UNABLE TO raiseNewXMLTopicsForRedoTransmissions: {0}
Occurs When:	
Corrective Action:	None.

Integration Message

Heading	Data
Message:	THE SAVED QUERY {0} RETURNS NO DATA
Occurs When:	The saved query in the SShipUnit element did not return any values.
Corrective Action:	Verify that the integration saved queries are correct and that the desired ship units exist.

20 Appendix: Uploading XSL Stylesheets

Uploading XSL Stylesheets

Using XSL Stylesheet files for inbound or outbound XML transformation is deprecated as of version 6.4.2 and will be removed in a future version. Any current use should be converted to use Stylesheet Content or Stylesheet Profile that refers to Stylesheet Content, as appropriate. For inbound transformation via HTTP see the *Transform Inbound XML with XSL chapter*. For outbound transformation via all transport protocols see the *Transform Outbound XML with XSL chapter*.

The following process captures the steps required to upload an XSL file.

1. Sign in using a using with DBA.ADMIN privileges.
2. Go to **Business Process Automation > Integration > Integration Manager > Upload an XML/CSV Transmission**.
3. Browse to select XSL file (must have ".xsl" file extension).
4. **Upload**.
5. Ensure response shows file has been uploaded to correct directories in the application servers.

21 Appendix: Interface Additional Detail

Actual Shipments

If the service provider sends a new order release as part of the actual shipment, Transportation and Global Trade Management Cloud creates an order release and order base for the new release.

An actual shipment is required to print shipment documentation such as a bill of lading or Domestic Packing List.

Actual Shipment Workflow Considerations

Changes to an existing shipment using the Actual Shipment interface can be configured to trigger additional business logic by way of agent workflow. When the shipment contains multiple orders or is one of a 'graph' of multiple shipments for different modes or service providers, this workflow can be quite complex and involve modification of other objects. Consequently, although simple modifications **CAN** be achieved using the Actual Shipment interface, e.g. addition of a new Shipment Reference Number, these cases should instead make use of the Generic Status Update interface.

Structural Changes to Shipment Ship Units in Release 6.0

Prior to Transportation and Global Trade Management Cloud release 6.0, shipment ship units were shared among different shipments/legs of a multi-leg movement. With the multi-tier execution enhancement in release 6.0, this has been changed to not allow sharing of the shipment ship units across multiple shipments. As a result of this change, updates to Shipment Ship Unit details (quantities or attributes) will not automatically be reflected across all legs when a change is made to a shipment ship unit. There are several options for making the appropriate changes to each of the impacted legs:

- Update Each Shipment Manually
  - Update each of the shipments impacted with a separate ActualShipment xml. Each ActualShipment xml would update the appropriate ship units on that shipment.
- Leverage Integration Saved Query
  - Leverage the IntSavedQuery within the Shipment/ShipmentHeader and Shipment/SShipUnit to search for the GIDs of the shipment(s) and ship unit(s).
  - This option is beneficial when the shipment and/or ship unit GIDs are not all known in the integration layer.
  - This option is limited in that any other fields set in the ActualShipment would be applied to all shipments identified by the query. A similar situation exists for the Ship Units on the shipments.

- Use Propagation for Ship Unit Changes
    - This is useful to propagate the count and quantity changes to upstream or downstream shipments.
    - Initiated by setting the Shipment/ShipmentHeader/IntCommand/IntCommandName = "PropagateShipUnitChanges". The propagation option (upstream, downstream, or both) is specified by the IntCommand/IntArg element. Refer to the GLog XML Schema for the specific values to be used.
    - This option has limitations on the types of changes it will handle and propagate. Several are listed as follows:
        - Cannot Add SEquipment From Another Shipment: You cannot add a SEquipment that already exists on another shipment to a shipment with the propagation. The SEquipment addition should be done in a separate ActualShipment without the propagation.
        - Cannot Change SEquipment for Existing Ship Unit: If there are more than one SEquipment on a Shipment, you cannot change the SEquipment for an existing Ship Unit with the propagation.
        - Cannot Add New SEquipment for Existing Ship Unit: You cannot add a new SEquipment and assign to an existing Ship Unit with the propagation. The SEquipment changes should be done in a separate ActualShipment without the propagation.
        - Limited Support to Add New SEquipment and New Ship Unit: You can add a new SEquipment and assign to a new Ship Unit. The logic will determine the new SEquipment and persist it before initiating the propagation logic.
        - Can Add a New Ship Unit: The propagation supports adding a new Ship Unit to the Shipment. The Ship Unit must be assigned to an existing SEquipment on the shipment.
        - Cannot Use DR Transaction Code In Ship Unit: You cannot use the DR transaction code in the Ship Unit to dereference the Ship Unit from the Shipment. Since the Ship Units are no longer shared among shipments, you should consider using the D transaction code to delete the ship unit. If there is a need to use the DR transaction code, it should be done in a separate ActualShipment without the propagation.

> **Note:** When migrating to Transportation and Global Trade Management 6.0 from a previous release, review the implementation of the Actual Shipment integration into Transportation and Global Trade Management to determine the impacts of this change.

Updating Parts of a Shipment

When sending an actual shipment to Transportation and Global Trade Management Cloud you often want to update parts of an existing shipment. Generally, the TransactionCode of the shipment (ShipmentHeader/TransactionCode) provides the guiding rule for the child elements. Here are some examples:

Sample Elements

Element	Description
Shipment/ShipUnit/ShipUnitContent	If TransactionCode=IU, a Line Number that does not exist will be added, otherwise updated. You cannot delete an individual line.
Shipment/ShipmentHeader/ShipmentRefnum	If TransactionCode=IU, a new Qualifier Value pair will be added. You can delete (and replace) using the GenericStatusUpdate interface.
ShipmentHeader/Remark	If TransactionCode is IU and the RemarkSequence does not exist, then Transportation and Global Trade Management Cloud will automatically generate a sequence number and add the remark.

Element	Description
	If TransactionCode is IU and the RemarkSequence does exist, Transportation and Global Trade Management Cloud updates with a new RemarkQualifier and RemarkText. If you supply neither a RemarkSequence nor a RemarkQualifier, Transportation and Global Trade Management Cloud adds the RemarkText as new Remark. You can delete (and replace) using the GenericStatusUpdate interface.

In the Shipment element, if you set the transaction code to RC and set the ReplaceChildren/ManagedChild element to "ShipmentStop", Transportation and Global Trade Management Cloud deletes all shipment stops for that shipment and replaces the deleted shipment stops with the shipment stops from your transmission.

Note: This does not apply to shipment stops marked IsPermanent (same as Permanent check box in Transportation and Global Trade Management Cloud web interface).

In the Shipment element, if you set the transaction code to RC and set the ReplaceChildren/ManagedChild element to ShipmentStopDetail, Transportation and Global Trade Management Cloud replaces the existing ship units with the ship units in your transmission.

Note: This does not apply to existing ShipmentStopDetails marked IsPermanent.

Note: Within ShipmentStopDetail, the removal of the reference to the ShipUnitGID(s) will not remove the S_Ship_Unit from the system. Only the reference to the object is removed.

In the Shipment element, if you set the transaction code to U and the Shipment ID is missing from either the transaction or the database, you will receive an error.

When a new shipment referencing a ship unit is added with missing ship unit data, then the ship unit data is pulled from the database. When a new shipment referencing a ship unit is added with new data, then the ship unit data passed in through integration is used.

Adding Stops

There is no way to insert a new stop to a shipment via shipment actuals unless the new stop has a stop number that does not already exist on the shipment (like adding stop #3 to a 2-stop shipment, or adding stop #2 to a shipment with stops 1 and 99).

Adding Ship Units

An added ship unit should be linked to an order release that is on a shipment (this order release must be planned on the initial shipment) and should be linked to the initial pickup location. If the check box on the Shipment Header indicates "Propagate Updates," the Transportation and Global Trade Management Cloud integration layer will call business logic to add the new ship unit to subsequent stop on the initial shipment and all affected succeeding shipments.

To add a ship unit to a shipment, the following must be done in the ActualShipment XML interface:

1. Specify a flag to indicate that new ship unit should be applied to downstream shipments.

   ```
   AcutalShipment.Shipment.ShipmentHeader.IntCommand.IntCommandName = "PropagateShipUnitChanges"
   ```

2. Indicate that Drop-off stop should be determined for the shipment, but not to propagate the ship unit changes.

   ```
   ActualShipment.ShipmentHeader.IntCommand.IntCommandName = "DetermineShipUnitDropoff"
   ```

3. Specify the new ship unit.

   ```
   ActualShipment.ShipmentHeader.Shipment.ShipUnit.ShipUnitGID =
   ```

4. Specify the Transaction Code (optional).

   ```
   ActualShipment.ShipmentHeader.Shipment.ShipUnit.TransactionCode = "I" or "IU"
   ```

5. Specify the pickup stop for the ship unit.

   ```
   ActualShipment.ShipmentHeader.Shipment.ShipmentStop.ShipmentStopDetail.Activity = "P"
   ActualShipment.ShipmentHeader.Shipment.ShipmentStop.ShipmentStopDetail.ShipUnitGID =
    ShipUnit.ShipUnitGID
   ```

6. ShipmentFrom and ShipmentTo Locations can in the Shipment.ShipUnit element are ignored. They will be based on the release.
7. Assign an SEquipment for the new ShipUnit via one of the following options:
8. Using the ShipUnit.SequipmentGID element, do the following:
   a. Allow integration to assign it by using the following:

      ```
      ActualShipment.ShipmentHeader.Shipment.ShipUnit.SEquipmentGIDQuery.SequipGIDMatchOption = "Any"
      ```

   b. Query the SEquipmentGID using the follwing:

      ```
      ActualShipment.ShipmentHeader.Shipment.ShipUnit.SEquipmentGIDQuery.IntSavedQuery
      ```

   c. Have the business logic assign it by not specifying the element in the ship unit.

Updating Ship Units

The updating of ship units means packaged items can be deleted or added to those ship units. Quantities from existing items can also be changed.

The following options are available via integration:

In the SShipUnit XML interface:

```
SShipUnit.TransactionCode = "RC"
SShipUnit.ReplaceChildren.ManagedChild = "ShipUnitContent"
```

In the ActualShipment XML interface:

```
Shipment.ShipmentHeader.TransactionCode = "RC"
Shipment.ShipmentHeader.ReplaceChildren.ManagedChild = "ShipUnitContent"
```

In the Shipment.ShipUnit.SShipUnit XML interface:

```
ActualShipment.Shipment.ShipUnit or ActualShipment.Shipment.SShipUnit
SShipUnit adds the ability to query for the ShipUnitGID if it's not known
```

Deleting Ship Units

Deleting a ship unit only removes the link between the shipment stop and the ship unit, as well as the link between the ship unit and equipment. The actual ship unit will not be deleted from the database. integration will also attach the ship unit remark, "Ship Unit Not Picked Up" to the ship unit.

A ship unit can be marked for removal from the shipment via the TransactionCode as follows:

```
ActualShipment.Shipment.ShipUnit.TransactionCode = "DR"
```

where "DR" corresponds to "Delete Reference." The ship unit will be removed from the shipment, but not deleted from Transportation and Global Trade Management Cloud.

Alternatively, you can delete ship units from a shipment using the IntCommand via integration. You can either delete all the ship units from the shipment, or only those that are marked as non-permanent. When used, the ship unit record, its shipment stop detail record, and any corresponding equipment, is deleted.

Specify the integration command as follows:

To remove all ship units:

```
<IntCommand>
 <IntCommandName>RemoveAllShipUnits</IntCommandName>
</IntCommand>
```

To remove only non-permanent ship units (where IsPermanent = 'N'):

```
<IntCommand>
 <IntCommandName>RemoveNonPermanentShipUnits</IntCommandName>
</IntCommand>
```

To remove orphaned ship units, use the command below. This specifies that the ship units that have been removed from the shipment via the DR transaction code should be deleted if no other shipments refer to them. Without this command, those ship units are left in the system and can later be added to other shipments.

```
<IntCommand>
 <IntCommandName>DeleteOrphanedShipUnits</IntCommandName>
</IntCommand>
```

Alternative Interfaces for Updating Ship Units

For alternatives to using this interface to update ship unit information, see SShipUnit and TransOrder.

Tips for Shipments As Work or SAWs

Element	Description
ShipmentHeader2	In Shipment/ShipmentHeader2, the most important element is ShipmentAsWork and it should almost always be set to "Y". The exception is when: 1. The shipment is new AND

Element	Description
	2. There is at least one release associated with the shipment OR the shipment.ShipmentHeader2. AutoGenerateRelease = "Y" To avoid confusion, set the Perspective element to "B." If Perspective is not specified, it will default to "B". All other elements in this branch are completely optional from a schema and business perspective.
ShipmentHeader, use correct rate	In the Shipment/ShipmentHeader, if you can get them, it is good to provide the RateOfferingGID and the RateRecordGID. This helps Transportation and Global Trade Management Cloud use the rate for the service provider that actually took the load. If you can't get information for these elements, the ServiceProviderGID would be the next best thing to use.
sPlannedTimeFixed	If you want to send in old shipments (past dates) and want Transportation and Global Trade Management Cloud to rate with correct rates (pertaining to correct effective/expiration dates), then you may want to use the Shipment/ShipmentStop/ArrivalTime/EventTime/IsPlannedTimeFixed flag set to "Y". You only need to insert the ArrivalTime element at the first stop (only) and that the date here should be the same as the StartDate of the shipment.
SEquipment	Transportation and Global Trade Management Cloud requires at least one SEquipment object. When you insert a new SAW, Transportation and Global Trade Management Cloud creates a default SEquipment if you do not provide one. If there are several ship units, the same (created equipment) is specified for each ship unit. The only thing really required in SEquipment is the SEquipmentGID. To avoid problems later, include the SEquipment element and set the SEquipmentGID to the same value as the ShipmentGID. This makes it easier to identify and manage the SEquipment, if there is ever a need in the future to specify multi-equipment, or update the shipment with additional ship unit information. **Note:** When you update a SAW with a new ship unit, you must include the SEquipmentGID. Transportation and Global Trade Management Cloud cannot create one for you.
TransOrder	The Shipment/TransOrder element is outbound only so you cannot include it. Any TransOrderHeader info should be specified in the Release/TransOrderHeader element.
Locations	When specifying source and destination locations in the Shipment/Release/ShipFromLocationRef and Shipment/Release/ShipToLocationRef elements, refer to locations already defined in Transportation and Global Trade Management Cloud instead of defining new ones (using Shipment/Release/ShipFromLocationRef/LocationRef/Location). This saves you the effort of providing all the Location elements. To refer to existing locations, use the Shipment/Release/ShipFromLocationRef/LocationRef/LocationGID element.
ShipUnits	Set ShipmentHeader2/AutoGenerateRelease to "Y" to avoid having to populate both Shipment/Release/ShipUnit and Shipment/ShipUnit. **Note:** Transportation and Global Trade Management Cloud generates an error if AutoGenerateRelease is set to "Y" and you still include the Shipment/Release element.

Order-Centric Modifications

Most modifications via this interface are based around shipment ship units (SShipUnit). In these cases, all weight, volume, quantities, and rating are based on shipment ship units. If you would like to modify shipments based on order information, you can do so by using the following sub-elements in the ShipmentHeader element:

- ShipmentModViaOrderLine
- ShipmentModViaOrderSU

When these two elements are used in the ShipmentHeader, the following logic will be used instead of the standard Shipment Interface logic:

1. Transportation and Global Trade Management Cloud will only interact with the order line level or order ship unit information instead of the shipment ship unit level information.
2. The logic addressing shipment modifications will change the number of order ship units involved and allocate the delta in the ship unit counts across multiple ship units.
3. The modified order ship unit count will be properly propagated and the related business objects (shipments and order movements) will be updated across legs.
4. The modified gross weight and volume will be updated per ship unit. This would then be reflected in the shipment total gross weight and volume, which impacts the shipment cost. This should only be applied when the AffectsCurrentLegOnly element is set to 'N'.

Both of these elements will only be included once on the shipment. There is no need to repeat this data for both the pickup stop and the delivery stop. Since the Shipment Interface is defined the same on the inbound and the outbound, you can only specify one way for the modification to happen, either at the order line level or the ship unit level.

ShipmentModViaOrderLine

The ShipmentModViaOrderLine element will contain all of the counts, weights, and volumes for that order (order release, order release line, or order base) that is being shipped on this shipment across all the shipment ship units.

ShipmentModViaOrderSU

The ShipmentModViaOrderSU element, Transportation and Global Trade Management Cloud will loop through all the ship units that are on the shipment that have the same order ship unit GID (ob_ship_unit_gid or or_ship_unit_gid).

The inbound XML will accept this data into Transportation and Global Trade Management Cloud when you are doing a modify transaction. When the integration brings in this modification it will call business logic that will apply allocation rules and perform the appropriate updates.

Data Requirements

To send shipments and perform planning actions, you must make decisions about the way you want Transportation and Global Trade Management Cloud to perform certain actions.

Sending Shipments (Shipment as Work)

Send shipments that do not have orders associated with them to Transportation and Global Trade Management Cloud for processing using the Shipment interface. This type of shipment is known as a shipment as work or manual shipment; it can include order level information, but not necessarily. These shipments are not bundled, re-consolidated, or re-sequenced.

A shipment as work must have at least one pickup and one delivery location. A shipment as work is not associated with an itinerary.

> **Note:** To indicate that the shipment you are sending to Transportation and Global Trade Management Cloud is a shipment as work, enter Y in the ShipmentAsWork element.

When a Shipment as Work is received, Transportation and Global Trade Management Cloud can be set to automatically perform certain actions defined in public workflow agents in the Agent Manager.

To ensure best possible performance, you should let Transportation and Global Trade Management Cloud process your actual shipments in parallel. To do this, either send only one actual shipment per transmission or follow these steps:

- In the TransmissionHeader, set IsProcessInSequence to N.
- Send all the actual shipments in one Transmission.

See the *Shipment Manager help* for a detailed description of manual shipments.

If you insert a new shipment and omit the end_date, Transportation and Global Trade Management Cloud sets the end_date to the same date as the start_date.

Accrual Interface

When the ALLOCATION GENERATES ACCRUALS parameter is set to TRUE and the shipment status is ACCRUAL_ALLOWED, the allocation logic will generate an accrual record. The accrual record contains the difference between the current allocated freight cost and the previously transmitted freight cost. The delta between the two is used because a single order may be on multiple shipments, which are approved for payment in different time periods. These accrual records are sent as part of this interface.

Invoice Interface

Invoices can be automatically matched to shipments based on the Service Provider ID and Shipment Reference Number fields. If more than one shipment is found for an invoice, the invoice must be reviewed manually and assigned to a shipment, it must be rejected. After an invoice is approved, a voucher gets created. A voucher represents what the planner agrees to pay for the shipment.

Use the Financials managers to create and modify invoices and customer bills.

In some cases, you may need to send the Invoice interface outbound. This is true when sending a bill to yourself for internal invoice or billing purposes. When the Invoice interface is used outbound, then it will include all shipment details, as well as any associated order information.

Consolidated Invoices

1. You must send each invoice, parent and child, as a separate transaction.
2. Parent invoices must enter Transportation and Global Trade Management Cloud before any child invoices.
3. Child invoices may be sent inbound referencing a parent in one of two ways:
   - Populate the invoice number on the child to be that of the parent and integration will look up the parent id based on the invoice number.
   - Populate the parent invoice ID on the child.

Job Interface

A job offers a workspace that brings together the objects and activities required of them, including:

- The ability to group all objects related to a job and perform existing functions/actions against those objects, including buy shipments, sell shipments, non-freight related charges, and customer bills.
- The ability to manage jobs from various perspectives depending on responsibility. For example, export, import, both, or consolidations.
- The ability to manage settlement functions at the job level, including profitability, expenses, revenues, and billing.

> **Note:** The interface is supported on the outbound only.

The primary business objects in the Job Interface are:

1. Order releases
   - For each order release related to the job, JOB_ORDER_RELEASE_JOIN will be added to the XML. Although orders are not required to create a job, at least one order should be related to the job to send out the interface.
2. Buy side costs
   - The Buy Side Costs wrapper element contains two sub elements, Buy Shipments and Buy Allocation.
   - Buy shipments: Select all related orders. For each order, select all related buy shipments where the shipment job GID equals null or it equals the current job GID. There may be zero or more buy shipments.
   - Buy allocation: Each allocation will be selected for each order, where the allocation Shipment Job GID is equal to the current job or it is equal to null. There may be zero or more allocations.
3. Sell side costs
   - The sell side costs works exactly the same as the buy side, except the selection criteria is based on sell side perspective.
4. Bills
   - Each customer bill related to the job will be included in the XML. Bills can be found in the JOB_BILL table. Zero or more bills are required.
   - Because this interface can potentially be large, redundant data has been reduced across multiple data elements included in the interface. This includes:
   - The ability to only include the Order Release GID in the shipment, allocation, and bill elements.

- The ability to only include the Shipment GID in the allocation and bill elements.

Release Interface

An order release contains the following information:

- Order release ID that is automatically generated by Transportation and Global Trade Management Cloud.
- Order release name and type.
- Order base ID that references the base order from which the order release was created.
- Source and destination locations.
- Early/late pick up dates.
- Assigned or fixed itinerary.
- Current status.
- Package or non-package data attributes.

Release Method

Populate the ReleaseMethodGid to tell Transportation and Global Trade Management Cloud how to release ship units. TheReleaseMethodGid (also called Order Configuration) tells Transportation and Global Trade Management Cloud how to build ship units from order lines, or how to calculate ship unit information if ship units are entered. If ReleaseHeader/ ReleaseMethodGid is not populated, the default can be property controlled.

Business Number Generator (BNG)

You can send a transportation order to Transportation and Global Trade Management Cloud, without the entering values in the TransOrder GID, Ship Unit ID, or Order Release ID elements in the XML Transmission. Transportation and Global Trade Management Cloud generates values for these fields based on the default business number rule in place when the order comes into the system. You can set up the BNG to create numbers that fit your needs.

ShipmentStatus Interface (INE)

You can also send received ship unit quantities with this interface using the SStatusShipUnit element.

Insert New Shipment Status into Transportation and Global Trade Management Cloud

Required Data

Before you can send ShipmentStatus transmissions to Transportation and Global Trade Management Cloud, you must set up the following:

- User accounts for service providers in Transportation and Global Trade Management Cloud
- Shipment Status Codes
- Shipment Status Reason Codes
- Shipment Event Groups
- Shipment Reason Groups
- Corporations for service providers

What Data Goes into the Transmission?

1. To ensure that Transportation and Global Trade Management Cloud processes multiple ShipmentStatus transactions in the order you intend, set IsProcessInSequence to Y in the TransmissionHeader.
2. Identify which object (shipment, shipment group etc.) the status applies to. Set StatusLevel, ShipmentStatusType, ServiceProviderAlias, and ShipmentRefnum or IntSavedQuery.
3. ShipmentStatusType must be set to one of Shipment or ShipmentGroup. Note that it is case sensitive. The integration logic assumes that it is a ShipmentGroup if the value is not matched to Shipment.
4. Optionally, identify which equipment the shipment status refers to with SStatusSEquipment element.
5. Include the time when the event occurred. To be sure that Transportation and Global Trade Management Cloud can interpret the time correctly, include the TimeZoneGID element. Alternatives to doing this is:
   - If you cannot include the TimeZoneGID, Transportation and Global Trade Management Cloud can set the time zone to the time zone of the Location where the event occurred.
   - If you cannot do this either, set the TimeZoneGID to Local. In this case, Transportation and Global Trade Management Cloud saves and displays the event date as entered, ignoring user preferences.
6. Enter your status information. In some cases, shipments can only have events added to them if they are of a certain status.
7. Identify at what SSStop (number or location name) the event (shipment status) occurred. The LocationID = Location Reference Number and the LocationRefnumQualifierGID = Location Reference Qualifier in the Location Manager.
8. If you have a Shipment Agent Type with the Recalculate Estimated Stop Times action, then you must include a RATE_GEO element for Transportation and Global Trade Management Cloud to be able to recalculate your estimated stop times and/or re-drive your shipment. If you omit the RATE_GEO element, Transportation and Global Trade Management Cloud only resets the stop times you provide.
9. See the ShipmentStatus Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application like XML Spy.

Send Shipment Status from Transportation and Global Trade Management Cloud

You can forward a received ShipmentStatus transmission to an external system with an agent. See the agent action called SEND SHIPMENT STATUS XML.

Match Events to an Object

You can use one of these methods:

- If you need the object (shipment or shipment group) to match many reference numbers, use IntSavedQuery.
- If you need the object to only match one out of a set of reference numbers, use ShipmentRefnums, or ShipmentGroupRefnums.

IntSavedQuery

If you specify the IntSavedQuery element, only that query is applied. You can define a query to search for shipments or shipment groups that shipment status applies to. To do this, set:

- IntSavedQueryGID to specify which query you want to use. If the query you specify here does not return any results, Transportation and Global Trade Management Cloud generates an error message. No other queries are applied. You must have created this query in Power Data beforehand.
- IntSavedQueryArg to arguments that can be referred to in the queries. For example, BM=YELLOW-0000007. If you omit this element, your IntSavedQueryGID must point to a query that uses XPath instead.
- IsMultiMatch to N to forbid multiple records to be returned from the query. If your query happens to return multiple records, Transportation and Global Trade Management Cloud generates an error message.

Refnums

If you omit the IntSavedQuery element, Transportation and Global Trade Management Cloud tries to match your shipment status with:

- The ShipmentRefnum elements to the shipment_refnum table in the database.
- The SSEquipment/ EquipmentIdentificationNum element to the S_Equipment.Equipment_Number field in the database
- The standard integration saved query INT_SHIPMENT_STATUS_GID_1
- The standard integration saved query INT_SHIPMENT_STATUS_GID_2

You can optionally enforce a rule that a give shipment may have only one shipment reference number with a given qualifier. The update_flag column in the shipment_refnum_qual table indicates if the rule is in effect or not. The valid values for the update_flag are:

- UPDATE_OK: Only one value is allowed for a give qualifier, the value of which can be modified.
- UPDATE_NOT_OK: Only one value is allowed for a give qualifier, the value of which cannot be modified.
- MANY: a given shipment can have multiple values for the same qualifier.

Match Events to a Shipment Stop

For Transportation and Global Trade Management Cloud to match an event to a stop on a shipment, you must include the SSStopSequenceNum element.

Another way of matching event to shipment stop is to include the LocationID where the event occurred and the LocationRefnumQualifierGID in SSStop/SSLocation. This only works if you have enabled this feature in your glog.properties file. As long as Transportation and Global Trade Management Cloud can match your LocationID to a stop number, your shipment status saves as if you had supplied a stop number.

> **Note:** If Transportation and Global Trade Management Cloud cannot match the event to a location, Transportation and Global Trade Management Cloud still saves the information but not for a specific stop and only as informational. You can also have Transportation and Global Trade Management Cloud send you a TransmissionReport if the LocationID is missing altogether (controlled by glog.properties). Transportation and Global Trade Management Cloud also set the time zone for the event to local.

Correspondingly, if the event is not related to a shipment stop to begin with, Transportation and Global Trade Management Cloud saves the event as informational with a local time zone.

A single stop related shipment event can be applied to multiple shipments, regardless of whether their stop numbers or location IDs are the same. This will allow for situations where you want to apply a single shipment stop event to stop 2, but stop 2 of shipment 1 and stop 1 of shipment 2 are both Philadelphia. Stop related events are applied to all the shipments specified in the ShipmentStatus interface. To work successfully, the ShipmentStatus XML must include an IntSavedQuery element that will return two shipments. Logically, this is similar to having specified the ShipmentStatus message multiple times in the Transmission XML.

TransOrder Interface (INO)

Create, modify, or delete order information through the TransOrder interface.

Insert New Order and Release Order Line

This procedure shows you how to:

- insert a new order
- release all or part of an order line
- build shipments from the order release

Required Data

To send an order to Transportation and Global Trade Management Cloud, certain information related to the order must already exist in Transportation and Global Trade Management Cloud. For example, you must have a valid itinerary, rate, locations, and so on.

Setup

You control validation of incoming transmissions with the `glog.integration.validation` property.

What Data Goes into the Transmission?

1. Set the TransactionCode to I. A transaction code of UI or IU works too.
2. If you do not want to enter values for the TransOrderGID, TransOrderLineGID, ShipUnitGID, or OrderReleaseGID elements, you can have Transportation and Global Trade Management Cloud automatically generate GIDs. Automatic generation of GIDs only works for a transaction code of I.

   > **Note:** If a transaction code of IU is used, then a TransOrderGID must be provided.

3. Populate the ProcessingCodeGID to tell Transportation and Global Trade Management Cloud how to plan the shipments from the order release.
4. Populate the TransOrderLineDetail element, including the PackagedItemCount, WeightVolume/Weight, and WeightVolume/Volume under TransOrderLineDetail/TransOrderLine/ItemQuantity/ to specify the order lines. You can set all but one of them to 0, if your setup uses the same kind of quantity to release.
5. Set TransOrderLineDetail/TransOrderLine/ItemQuantity/IsShippable = N.
6. Populate the amount to release in the TransOrderHeader/ReleaseInstruction/QuantityToRelease element.

   If you omit the ReleaseInstruction, set TransOrderLineDetail/TransOrderLine/ItemQuantity/IsShippable = Y to have Transportation and Global Trade Management Cloud create an order release for all of your order lines.
7. If your order base is coded in a format other than the supported format you need to transform your TransOrder transmission into the Transmission schema, you can use Transportation and Global Trade Management Cloud's transform feature to do this.
8. If you want to change the level of validation for this transmission, you can include a processing instruction to set the desired level. By default, TransOrder validation is turned on.
9. See the TransOrder Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application.

Transmission Results

1. Transportation and Global Trade Management Cloud receives your transmission and starts to process it internally.
2. Transportation and Global Trade Management Cloud starts the public Order Base - Insert agent.

   If the current date is outside the effective date/expiration date window of your TransOrder, the agent cannot create order releases. You must release the TransOrder via the process manager. There you can release all orders which have release instructions, but whose release has not been processed.

   If you use the UI or IU transaction codes and the record exists already, Transportation and Global Trade Management Cloud starts the public Order Base - Modify agent instead.
3. Transportation and Global Trade Management Cloud raises events that in turn can trigger Notifications to be sent.

Error Messages

If you receive a TransmissionReport, check for integration messages.

Insert New TransOrder and Release ShipUnit

Auto-releasing ship units is consistent with order base lines. This procedure shows you how to:

- insert a new order
- release all or some ship units on the order base
- build shipments from the order release

Required Data

To send an order to Transportation and Global Trade Management Cloud, certain information related to the order must already exist in Transportation and Global Trade Management Cloud. For example, you must have a valid itinerary, rate, locations, and so on.

Setup

You control validation of incoming transmissions with the `glog.integration.validation` property.

What Data Goes into the Transmission?

1. Set the TransactionCode to I. A transaction code of UI or IU works too.
2. If you do not want to enter values for the TransOrderGID, ShipUnitGID, or OrderReleaseGID elements, you can have Transportation and Global Trade Management Cloud automatically generate GIDs. Automatic generation of GIDs only works for a transaction code of I.

   > **Note:** If a transaction code of IU is used, then a TransOrderGID must be provided.

3. Populate the ProcessingCodeGID to tell Transportation and Global Trade Management Cloud how to plan the shipments from the order release.
4. Populate the TransOrder/ShipUnitDetail element.
5. To be able to track your ship unites as they propagate through Transportation and Global Trade Management Cloud as order release ship units and shipment ship units, you might want to include a unique ID in the ShipUnitDetail/ShipUnit/ShipUnitContent/ItemQuantity/ItemTag1 element. Also, there is a TransOrderShipUnitGID element in Release/ShipUnit that can help you track ship units.
6. Set ShipUnitDetail/ShipUnit/IsShippable to Y to have Transportation and Global Trade Management Cloud create an order release for all your order base ship units.
7. If you omit the IsShippable element or set it to N, you need to populate the amount to release in the TransOrderHeader/ReleaseInstruction/QuantityToRelease element. With this option, you can specify the number of ship units to be released in the ReleaseInstruction/ShipUnitReleaseCount element.
8. You can override all dates and locations from the ShipUnitDetail with other settings in the ReleaseInstruction.
9. If you want to change the level of validation for this transmission, you can include a processing instruction to set the desired level. By default, TransOrder validation is turned on.
10. See the *Order Base help* for a description of the fields.
11. See the TransOrder Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application like XML Spy.

Transmission Results

1. Transportation and Global Trade Management Cloud receives your transmission and starts to process it internally.

2. Transportation and Global Trade Management Cloud starts the public Order Base - Insert agent.
3. It finds the unprocessed release instructions with a release date <= the current date.
4. If the current date is outside the effective date/expiration date window of your TransOrder, the agent cannot create order releases. You must release the TransOrder via the process manager. There you can release all orders which have release instructions, but whose release has not been processed.
5. If you use the UI or IU transaction codes and the record exists already, Transportation and Global Trade Management Cloud starts the public Order Base - Modify agent instead.
6. Transportation and Global Trade Management Cloud raises events that in turn can trigger notifications to be sent.

Modify Order Base with Lines

In this scenario, you can just update the information in an order base or you can update and release the full amount specified for the TransOrderLine.

Required Data

None

Setup

You control validation of incoming transmissions with the `glog.integration.validation.orderinterface` property.

What Data Goes into the Transmission?

1. Set the TransactionCode to U. A transaction code of UI or IU works too.
2. If your update is to delete only a couple of fields in the OrderBase, use the Value to Null Field symbol.
3. If you want Transportation and Global Trade Management Cloud to release all your order lines, set IsShippable = Y.

   With IsShippable=Y, you should omit the ReleaseInstruction element, Transportation and Global Trade Management Cloud releases the full weight, volume, or count (depending on glog.properties). If you set IsShippable=Y and include a ReleaseInstruction, Transportation and Global Trade Management Cloud releases your order twice. One full order release based on the parameter in glog.properties and another order release based on the ReleaseInstruction element.
4. If you want to change the level of validation for this transmission, you can include a processing instruction to set the desired level. By default, TransOrder validation is turned on.
5. See the *Order Base help* for a description of the fields.
6. See the TransOrder Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application like XML Spy.
7. To update date fields with NULL values, submit a value of '~' in the date element(s) of the inbound TransOrder XML.

Transmission Results

1. Transportation and Global Trade Management Cloud receives your transmission and starts to process it internally.
2. Transportation and Global Trade Management Cloud starts the public Order Base - Modify agent.

   If you use the UI or IU transaction codes and the record does not exist already, Transportation and Global Trade Management Cloud starts the public Order Base - Insert agent instead.

3. Transportation and Global Trade Management Cloud raises events that in turn can trigger Notifications to be sent.

Error Messages

If you receive a TransmissionReport, check for integration messages.

Modify ShipUnits

The following ways update the ship unit information (quantities, weights, volumes, etc.) on a shipment via integration:

- Use ActualShipment. This interface provides complete control of all the fields in the Shipment.
- Use SShipUnit.
- Send another TransOrder with the IsUpdateShipmentOnly element.

The TransOrder interface together with the IsUpdateShipmentOnly element supports uploading a slightly modified TransOrder and has it update only the Shipment/SShipUnit. IsUpdateShipmentOnly indicates that the TransOrder should update the shipment only, and not the order base information.

> **Note:** To update date fields with NULL values, submit a value of '~' in the date element(s) of the inbound TransOrder XML.

Using the IsUpdateShipmentOnly element can help you reduce the need to implement a separate SShipUnit or ActualShipment interface. The use of this flag with the TransOrder interface is restricted as follows:

- The original order base should have been created using the ShipUnitDetail (not the TransOrderLineDetail).
- The information you can update is restricted to the SShipUnit element. The TransOrderHeader is ignored, and none of the other Shipment related information is updated.
- The specific S_Ship_Unit(s) to be modified are identified by using the ShipUnitGID in the new TransOrder and searching for the related Release/ShipUnit (via the OB_SHIP_UNIT_GID on SHIP_UNIT table) and then the Shipment.ShipUnit(s) (via the SHIP_UNIT_GID field in the S_SHIP_UNIT table). The search requires those reference pointers to exist.

Delete Orders

Required Data

Setup

You control validation of incoming transmissions with the `glog.integration.validation.orderinterface` property.

What Data Goes into the Transmission?

1. Set the TransactionCode to D.
2. If you do not know the GID of the record you want to delete, you can use integration saved queries instead.
3. If you want to change the level of validation for this transmission, you can include a processing instruction to set the desired level. By default, TransOrder validation is turned on.

4. See the Order Base Manager for a description of the fields.
5. See the TransOrder Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application like XML Spy.

Transmission Results

1. Transportation and Global Trade Management Cloud receives your transmission and starts to process it internally.
2. Depending on what kind of record you are deleting an agent might start. For example, if you are deleting an Order Base, the public Order Base - Delete agent starts.
3. Transportation and Global Trade Management Cloud raises events that in turn can trigger notifications to be sent.

Error Messages

You cannot delete an order that is assigned to a shipment after a service provider accepts a tender on the shipment. If you try to do this, Transportation and Global Trade Management Cloud might send a TransmissionReport stating the problem.

If you receive a TransmissionReport, check for integration messages.

Bulk Plan Orders

Required Data

You must create a saved query that points out the order releases you want to include.

Setup

You control validation of incoming transmissions with the `glog.integration.validation.orderinterface` property.

What Data Goes into the Transmission?

If you can keep all your TransOrders within one transmission follow these steps:

1. Set IsProcessInSequence=N.

   This ensures maximum performance because Transportation and Global Trade Management Cloud can process TransOrders belonging to different order bases in parallel.
2. Include all TransOrders that should be bulk planned.
3. Create an order release for every TransOrder either with IsShippable=Y and omit the ReleaseInstruction, or with IsShippable=N and include a ReleaseInstruction.
4. Include a Topic element as the last element in the transmission to start the bulk planning. Set TopicArgName to 'savedQuery' and TopicArgValue to a Query_Name. The saved query must point out the Order Releases you want to include.
5. In the GLogXMLElement holding the Topic element, include a ProcessInfo element with WhenToProcess=END_OF_TRANSMISSION.

   This tells Transportation and Global Trade Management Cloud to wait to start the bulk planning until the end of the transmission.

> **Note:** Transportation and Global Trade Management Cloud plans all order releases that match the saved query, not just the ones within the transmission.

> **Note:** The Topic element must be the last element in the transmission. If it is not, Transportation and Global Trade Management Cloud will plan incorrectly.

If you cannot keep all your TransOrders within one Transmission follow these guidelines:

1. For every transmission with TransOrder, set IsProcessInSequence=N.
2. Create an order release for every TransOrder either with IsShippable=Y and omit the ReleaseInstruction, or with IsShippable=N and include a ReleaseInstruction.
3. For every TransOrder that Transportation and Global Trade Management Cloud should bulk plan later, set the ProcessingCodeGID to NOPLN.

   If you instead set the ProcessingCodeGID to PLN on each TransOrder in a transmission, Transportation and Global Trade Management Cloud bulk plans these orders on each transmission. Also, Transportation and Global Trade Management Cloud cannot supply a bulk plan history in this case.
4. When Transportation and Global Trade Management Cloud has received all TransOrders to be bulk planned, send a Topic element as the last element in the transmission to start the bulk planning. Set TopicArgName to 'savedQuery' and TopicArgValue to a Query_Name. The saved query must point out the Order Releases you want to include. If you want to supply your own bulk plan ID, in addition, set TopicArgName to 'bulkPlanID' and TopicArgValue to your desired bulk plan ID.

   To be reasonably sure that Transportation and Global Trade Management Cloud has received all your transmissions, allow sufficient amount of time between sending the last TransOrder Transmission and sending the Topic element.

   > **Note:** Transportation and Global Trade Management Cloud plans all order releases that match the saved query, not just the ones within the last transmissions.

   > **Note:** The Topic element must be the last element in the transmission or group. If it is not, Transportation and Global Trade Management Cloud will plan incorrectly.

Transmission Results

When Transportation and Global Trade Management Cloud completes the bulk planning, Transportation and Global Trade Management Cloud sends the results of the bulk plan in a BulkPlan element.

Error Messages

If you receive a TransmissionReport, check for integration messages.

Incrementally Release TransOrder Line from Existing TransOrder

In this scenario, you already have a TransOrder with a large amount of goods in a TransOrderLine in Transportation and Global Trade Management Cloud but now you want to release small amounts of that TransOrderLine with multiple subsequent TransOrders.

To incrementally release TransOrderLines from an order already in Transportation and Global Trade Management Cloud via an integration transmission, do the following:

Required Data

None.

Setup

You control validation of incoming transmissions with the `glog.integration.validation.orderinterface` property.

What Data Goes into the Transmission?

1. Make sure the public Order Base - Modify - Incremental Release agent is active.
2. Send a transmission of the record and enter the transaction code U in the TransactionCode element.
3. All your TransOrderLines must be marked IsShippable=N.
4. Always keep IsShippable=N between all these TransOrders.
5. Include a TransOrderHeader/ReleaseInstruction to release a fraction of the amount specified on the original TransOrderLine. If you omit the ReleaseInstruction element, Transportation and Global Trade Management Cloud only saves your order base since you have IsShippable set to N.
6. For each modified TransOrder you send, update the ReleaseInstruction/SequenceNumber and make it unique. If you do not, Transportation and Global Trade Management Cloud keeps the old releases but replaces the content of the release instruction.
7. See the Order Base Manager for a description of the fields.
8. See the TransOrder Diagram in the XML schema to learn which elements are required. To view the diagrams use an XML application like XML Spy.

Transmission Results

1. Transportation and Global Trade Management Cloud receives your transmission and starts to process your transmission internally.
2. Transportation and Global Trade Management Cloud starts the public Order Base - Modify - Incremental Release agent.
3. Transportation and Global Trade Management Cloud raises events that in turn can trigger notifications to be sent.

Error Messages

If you receive a TransmissionReport, check for integration messages.

Send TransOrder from Transportation and Global Trade Management Cloud

Required Data

None.

Setup

You must have created the external system you want to send to.

How To Send the Transmission?

In the Order Base Manager.

In the Process Manager, Send Integration Page.

What Data Goes into the Transmission?

Transportation and Global Trade Management Cloud includes all data specified for the order base.

Transmission Results

Error Messages

Processing Codes

When you send an order to Transportation and Global Trade Management Cloud, you can indicate whether you want Transportation and Global Trade Management Cloud to perform planning functions on it. If Transportation and Global Trade Management Cloud plans orders, it creates shipments from the orders and then executes the shipments as soon as it receives them. If you want to execute orders into shipments at a particular time or after you receive a certain number of orders, do not run the planning function.

Control the details of planning orders in Transportation and Global Trade Management Cloud in the Agent Manager.

In the ProcessingCodeGID element, enter one of the following values:

- **NOPLN**: Instructs Transportation and Global Trade Management Cloud not to plan shipments form the order. This is the default if you omit this element.
- **PLN**: Instructs Transportation and Global Trade Management Cloud to plan shipments from the order release. Transportation and Global Trade Management Cloud will plan multi-stop shipments if appropriate. You must have your TransOrder set up to create an order release for this to work.
- **MSPLN**: Obsolete.

Voucher Interface

Transportation and Global Trade Management Cloud sends voucher transmissions to the URL you specify in glog.properties.

You can create an agent that send the Voucher interface using the agent action Send Voucher Interface. You can also send this interface from the Invoice Manager.

> **Note:** You can send a voucher transmission that cancels or edits a previous voucher.

Transportation and Global Trade Management Cloud determines to whom a payment is due based on the involved parties defined on the order release.

The Shipment element is only included when generating vouchers for parent invoices. When generating a voucher for a child invoice, the Shipment element is not included.

You can optionally use the ShipmentGID element instead of the full Shipment element in order to reduce the size of the Voucher XML.

DataQuerySummary Interface

Contains the GID of a business object record.

Some external systems may not be prepared for Transportation and Global Trade Management Cloud to send large amounts of data. The DataQuerySummary interface provides a mechanism to send only a summary of the data. The external system can request the individual records from Transportation and Global Trade Management Cloud at appropriate times (e.g. idle times, overnight) by referencing the GID.

Send DataQuery Summary from Transportation and Global Trade Management Cloud

Required Data

Setup

You must have created the external system you want to send to.

How To Send the Transmission?

Mark the Send Summary check box when sending from one of these managers:

- Order Base Manager
- Order Manager (use for order releases)
- Buy Shipment Manager
- Sell Shipment Manager
- Shipment Group Manager
- Bill Manager
- Service Provider Manager
- Location Manager
- Rate Offering Manager
- Rate Record Manager
- Item Manager
- Invoice Manager
- In the Process Manager, Send Integration Page.

What Data Goes into the Transmission?

Transportation and Global Trade Management Cloud includes only the GID of the record.

Transmission Results

Error Messages

The Role of the Transaction Code

The TransactionCode specifies whether the information should be inserted, or updated/replaced. For the Refnum objects that have the qualifier and value as part of the primary key, the TransactionCode indicates whether the new qualifier/value pair should be added (Insert), or used to replace all of the current records with the same qualifier (Update).

For example, the Shipment_Refnum table has a composite primary key made up of the ShipmentGID, RefnumQualifier, and RefnumValue. Assume a shipment has the following ShipmentRefnum Qualifier/Value pairs in the system: CO/A-12345, CO/B-89387, CN/C-83920. If you send a new refnum qualifier/value of CO/D-23849 using the GenericStatusUpdate interface, the TransactionCode would affect the change as follows:

- TransactionCode = I: The new refnum would be added, resulting in all of the following being present in the table: CO/A-12345, CO/B-89387, CN/C-83920, CO/D-23849.
- TransactionCode = U: The current reference numbers with the same qualifier would be deleted, and replaced by the new one. In this case, the result would leave the following in the table: CN/C-83920, CO/D-23849.

The TransactionCode is only applicable for the Refnum and Remark elements. It is not used for the Status or Indicator elements, which are only intended to be updated.

Topic Interface

This inbound interface allows you to raise a topic and get Transportation and Global Trade Management Cloud to start processing an object. Currently Transportation and Global Trade Management Cloud supports BuildBuySideShipments and BuildSellSideShipments that allows you to start bulk planning. Transportation and Global Trade Management Cloud also supports clearing caches using the interface.

> **Note:** Make sure Transportation and Global Trade Management Cloud has released all your TransOrders before sending the Topic element to Transportation and Global Trade Management Cloud.

> **Note:** When including other transactions in the same transmission as the Topic transaction, make the Topic transaction the last in the transmission.

The table lists what each element should contain.

Examples of What Topic Interface Elements Should Contain

TopicAliasName	TopicArgName	TopicArgValue
BuildBuySideShipments	savedQuery	query_name, e.g. YELLOW_ORDER_REL
BuildSellSideShipments	savedQuery	query_name, e.g. YELLOW_ORDER_REL
glog.server.workflow.adhoc.ClearCaches	cache	partial or full string matching the cache name, e.g. RateOffering
	zone	zone name, e.g. Rating or Business
	exactMatch	true, if the cache match should be exact

SKU Interface

How to Structure Your Data

If you have large amounts of highly complex, nested inventory information, you should use the XML column in the SKU_DESCRIPTOR table to store that information, rather than using nested SKU_DESCRIPTOR records. However, it will not be possible to search for SKUs using the XML info. You will only be able to search on the non-XML columns in the SKU and SKU_DESCRIPTOR tables.

If you have small or medium amounts of less complex inventory information, you can use nested SKU_DESCRIPTOR records instead. Using this method, it will be possible to find a SKU by a sub-descriptor.

SKU Table

The following sample SKU record shows a part used to make Novelty phones. The Novelty stock code is 2002, which is used to form the XID. This corresponds to the packaged_item novelty.8946. The warehouse is novelty.wh1. The supplier is General Electric, who also currently owns the inventory.

```
SKU_GID = novelty.2002-wh1
SKU_xid = 2002-wh1
Packaged_item_GID = novelty.8946
Warehouse_location_GID = novelty.wh1
Supplier_corporation_GID = novelty.ge
Owner_corporation_GID = novelty.ge
Quantity_on_hand = 1800
Min_level = 100
Max_level = 2000
Domain_name = novelty
```

SKU_DESCRIPTOR table - BLOB

Transportation and Global Trade Management Cloud cannot show BLOBs in tree view of the inventory manager.

This section illustrates the relational approach to storing SKU descriptor and sub-descriptor information in using the SKU_DESCRIPTOR table. This method would be used when it is necessary to use standard SQL to search SKU descriptor data.

The example below shows a top-level SKU_DESCRIPTOR record. Notice that the parent_sku_descriptor_seq is null.

```
SKU_GID = novelty.2002-wh1
SKU_descriptor_seq = 1
SKU_descriptor_type = status
SKU_desriptor_value = held
SKU_descriptor_quantity = 1000
Parent_sku_descriptor_seq = null
Domain_name = novelty
```

The example below shows a level-2 SKU_DESCRIPTOR record. The parent_sku_descriptor_seq is set to 1, pointing to the previous example.

```
SKU_GID = novelty.2002-wh1
SKU_descriptor_seq = 2
SKU_descriptor_type = reason
SKU_descriptor_value = damaged
SKU_descriptor_quantity = 600
Parent_sku_descriptor_seq = 1
Domain_name = novelty
```

The example below shows a level-3 SKU_DESCRIPTOR record. The parent_sku_descriptor_seq is set to 2, pointing to the previous example.

```
SKU_GID = novelty.2002-wh1
SKU_descriptor_seq = 3
SKU_descriptor_type = batch
SKU_descriptor_value = 001
SKU_descriptor_quantity = 250
Parent_sku_descriptor_seq = 2
Domain_name = novelty

SKU_GID = novelty.2002
SKU_descriptor_seq = 4
SKU_descriptor_type = batch
SKU_descriptor_value = 002
SKU_descriptor_quantity = 300
Parent_sku_descriptor_seq = 2
Domain_name = novelty

SKU_GID = novelty.2002
SKU_descriptor_seq = 5
SKU_descriptor_type = batch
SKU_descriptor_value = 003
SKU_descriptor_quantity = 50
Parent_sku_descriptor_seq = 2
Domain_name = novelty
```

SKU_DESCRIPTOR Table - XML

If the SKU descriptor information need not be fully searchable using standard SQL, then the XML column in the SKU_DESCRIPTOR table may be used to represent the information at level 2 and below. In other words, it would be possible to use standard SQL to search for a SKU descriptor, but not for a SKU sub-descriptor.

An example situation where the XML method may not be appropriate would be where the top level SKU is a combination of shoes of different styles. The top level SKU_DESCRIPTOR records would have one row for each style.

The level 2 SKU_DESCRIPTOR would have counts of sizes within each style. A query to determine the total inventory of size 9 shoes across all styles would not be possible using the XML method. You can think of similar examples for the auto industry, i.e. find the inventory of all cars with anti-lock brakes, etc.

When using the XML method for representing detailed SKU_DESCRIPTOR information, each client implementation will be responsible for developing their own industry-specific XML schema for that information. By default, the UI will display this information in a nicely formatted manner. The UI provides a mechanism whereby you can install customer-defined XSL for formatting information. However, this XSL file is purely optional.

Below is a snippet of how the information from the previous section might appear in the database if the XML approach is used instead of the nested SKU_DESCRIPTOR method. In this case, the parent_sku_descriptor_seq column is always null, and the XML column is used instead. In this case, the top level status information is available relationally. However, the lower level descriptors within that status are represented inside the XML.

```
SKU_GID=novelty.2002-wh1
SKU_descriptor_seq = 1
SKU_descriptor_type = status
SKU_descriptor_value = held
SKU_descriptor_quantity = 1000
Domain_name = novelty
Xml =
<SkuDescriptor>
 <type>damaged</type>
 <value>001</value>
 <quantity>600</quantity>
<SkuDescriptor>
 <type>batch</type>
 <value>001</value>
 <quantity>250</quantity>
</SkuDescriptor>
<SkuDescriptor>
 <type>batch</type>
 <value>002</value>
 <quantity>300</quantity>
</SkuDescriptor>
… etc …
</SkuDescriptor>
```

CharterVoyage Interface

The CharterVoyage interface is used to specify the charter voyage for creating a consol shipment. It is supported on both the inbound and the outbound.

A charter voyage represents an ocean transport movement by a carrier from a loading port to a discharge port. Within a charter voyage, there are several Stowage Modes, which represent, at a conceptual level, separate "compartments" within the charter voyage. There is also capacity associated with each stowage mode as defined on a consol that you create for each stowage mode defined on the charter voyage. This capacity controls the orders that can be booked on the charter voyage.

For a charter voyage, and each of its defined stowage modes, you can create a consol that has a single empty shipment attached. For each voyage, one consol is automatically created for each stowage mode defined on the voyage. A shipment is also created for each consol at the same time.

Most of the elements included in the CharterVoyage interface follow the fields available in the Charter Voyage manager and the Charter Voyage Stowage Details.

Consol Interface

The Consol interface is used to specify the shipment consolidator. It is supported on both the inbound and the outbound. A consol can be created for a charter voyage or air schedule (flight).

A charter voyage consol represents the weight, volume, FEU/TEU capacities of a specific stowage mode on a specific charter voyage. It captures the allocated, maximum, committed, booked, and produced capacity values when the status of a consol is changed as a result of booking orders on a shipment that is related to the consol.

For a freight forwarder, the consol is considered a group of house bills or a set of sell shipments. All actions related to manipulating a consol should be performed from the perspective of a sell shipment. For example, adding freight to a consol would be performed by selecting sell shipments to add to consol.

For example, a freight forwarder starts with a group of house bills or a set of sell shipments. They have also reserved flights. For each flight reservation, there is a consol for defining the reserved capacity of the flight. The sell shipments are then booked to consols to create buy shipments.

Generic Status Update Interface

The Generic Status Update interface is used to modify one of a number of simple properties of a set of business objects. The properties that can be modified are:

- Status
- Indicator
- Reference Number
- Remark

The following table shows which properties are supported by each valid business object (the GenericStatusObjectType is the value that must be passed in the transaction XML):

Generic Status Update Interface Supported Objects

Business Object (GenericStatusObjectType)	**Status**	**Indicator**	**Refnum**	**Remark**
INVOICE_LINEITEM			x	x
INVOICE	x	x	x	x
LOCATION	x		x	x
OB_LINE	x		x	x
OB_ORDER_BASE	x	x	x	x
OB_SHIP_UNIT	x		x	x
ORDER_RELEASE	x	x	x	x

Business Object (GenericStatusObjectType)	Status	Indicator	Refnum	Remark
SHIPMENT	x	x	x	x
SHIPMENT_STOP			x	x
SHIP_GROUP	x	x	x	x
S_SHIP_UNIT			x	x
S_SHIP_UNIT_LINE			x	x
VOUCHER		x	x	x
SCHEDULE	x			
CHARTER_VOYAGE			x	
CLAIM_LINE_ITEM			x	
CLAIM	x		x	x
CONSOL	x			
ITEM	x		x	x
JOB	x		x	x
ORDER_RELEASE_LINE			x	x
DOCUMENT	x			
CONTAINER_GROUP	x			
DRIVER	x	x	x	x
DRIVER_TYPE				x
POWER_UNIT	x	x	x	x
POWER_UNIT_TYPE				x
OR_STOP				x
EQUIPMENT	x		x	x
EQUIPMENT_TYPE				x
WORK_INVOICE	x			
SHIP_STATUS_SPCL_SERVICE				x
ORDER_MOVEMENT	x		x	x
CONTACT	x	x	x	x
GTM_TRANSACTION	x		x	x
GTM_TRANSACTION_LINE	x		x	x
GTM_REGISTRATION	x		x	x

Business Object (GenericStatusObjectType)	Status	Indicator	Refnum	Remark
GTM_STRUCTURE			x	x

22 Appendix: General Properties

General Properties

General Properties

Property	Description
glog.integration.remoteQuery.wrapReplyInT	See the *glog.integration Properties help* for details.
glog.integration.TransmissionReport	Used to determine when to send a TransmissionReport if no AckOption element is present in an inbound AckSpec. "on error" – (Default) Only send if there are errors. "no" – Never send "yes" – Always send.
glog.integration.TransmissionReport.readRe	Used when sending a TransmissionReport via the HTTP POST communication method. true – block and read the HTTP return code from the receiving URL false – do not attempt to read any response.
glog.integration.transmissionreport.include	1 – include Summary Detail in the TransmissionReport 0 – (Default) Do not include Summary Detail
glog.integration.validation	See the *glog.integration Properties help* for details.
glog.integration.validation.{interface name}	See the *glog.integration Properties help* for details.
glog.integration.enableCaseChange	See the *glog.integration Properties help* for details.
glog.integration.casechange.element.{element name}	See the *glog.integration Properties help* for details.
glog.integration.dataqueue.inboundDXITyp	See the *glog.integration Properties help* for details.
glog.integration.dataqueue.inbound	See the *glog.integration Properties help* for details.
glog.integration.dataqueue.transport.{com method}	See the *glog.integration Properties help* for details.
glog.integration.dataqueue.inbound.useMe	See the *glog.integration Properties help* for details.
glog.integration.enableTimeZoneCompatib	See the *glog.integration Properties help* for details.

23 Appendix: Testing and External Systems

Testing Tips

It is often the case that through the use of Out XML Profiles or XSLT stylesheets on outbound messages, the final XML document leaving the Transportation and Global Trade Management Cloud application is modified from the default representation. Consequently it would be useful to have a way of testing the process to generate it before finally using in a Production scenario. The following section outlines the ways in which this can be achieved.

External System Servlet

The External System Servlet allows the emulation of an HTTP POST to an external system. A test External System can be created that would use the URL in the HTTP section to refer to the External System Servlet on the same application server instance.

Although the final External System may use a different transport protocol – SOAP or HTTP – the main objective for this tool is to validate the XML message content which can be exactly the same for all methods and so the HTTP POST emulation should be useful in most cases.

> **Note:** If the requirement is to see SOAP Envelope content then this would not be possible with the servlet and another method would be needed.

The servlet supports the forwarding or saving of the XML to the following:-

- Email address
- File system file
- Message Center
- HTTP (essentially only allows logging)

The processing for each option is controlled by URL parameters, defined below, that are specified on the URL used for the HTTP section in the External System.

External System URL Parameters

Parameter Name	Description
forwardingProtocol	(optional) Controls the choice of where the XML should be forwarded or saved as a file. If it is present it must be one of the following values: • email • http • messagecenter Depending on the option chosen other URL parameters may be required (see sections below).

Parameter Name	Description
	If it is not present, the XML will be saved as a file on the targeted server in the 'Integration Temporary Directory' (see glog.integration.tempdir property in the *General Properties appendix*).
logOn	(optional) 0 – Default. Do not output URL parameters to log file. 1 – Output all URL parameters to a file in Integration temp dir. The file name of the file will be "ExternalSystemServletLog<N>.txt". The <N> is between 0 and 6 corresponding to each day of the week. There are up to 7 of these files. The oldest file is overwritten each day.
logContent	(optional) 0 – Default. Do not log XML content to log file. 1 – Output XML to log file (name/location described above).
remoteFileExt	(optional) If a file is to be saved this file extension should be used otherwise it will have no extension. This property is also available when the forwardingProtocol is not specified which results in the file being saved on the server. See forwardingProtocol above for details.

The following sections describe the additional URL parameters used when a forwardingProtocol is specified.

email

- **from**: Valid email address the mail should appear to be sent from.
- **to**: Valid email address the mail should be sent to.
- **smtphost**: The SMTP server that will be used to transport the message. If not specified the default server will be used.
- **subject**: Subject to appear on email.

http

This option essentially acts as the "do nothing" option other than provide logging capability via the logOn and logContent parameters.

messagecenter

When sent as a Message Center message, the event will be sent as a GC3 ALERT event and the XML content is sent as the event message content.

- **to**: Valid Transportation and Global Trade Management Cloud user account the message should be sent to.
- **subject**: Subject to appear on the message.

Examples

Email:

```
http://< hostname_of_OTM_server>/GC3/glog.integration.servlet.ExternalSystemServlet?
forwardingProtocol=email&smtphost=internal-smtp.hostname.com&from=admin@example.com& to=target@example.com&
 subject=TestMessage&logOn=0&logContent=0
```