Oracle® Transportation Management

Administration Guide

Release 6.5.3

Part No. F82868-02

October 2024

Copyright Notice

Oracle® Transportation Management Administration Guide, Release 6.5.3

Part No. F82868-02

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Contents

Figures

Send Us Your Comments

Oracle® Transportation Management Administration Guide, Release 6.5.3

Part No. F82868-02

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: otm-doc_us@oracle.com

If you have problems with the software, contact Support at https://support.oracle.com or find the Support phone number for your region at http://www.oracle.com/support/contact.html.

Preface

This document contains information regarding the post-install configuration and maintenance of the following products: Oracle Transportation Management, Global Trade Management, and Transportation Intelligence.

1. Architecture Overview

Oracle Transportation Management is built for interactive environments that leverage the internet as an information backbone to capture reference data such as updates to carriers, carrier rates, shipping schedules, locations, ports, and other logistics sources. The underlying architecture is a unique multi-enterprise data model with n-tier web and application servers, partitioned database server, native XML language, extensible style sheets, advanced data security, and optimal caching.

Oracle Transportation Management is highly distributed and collaborative. It is written in Java and uses Enterprise Java Beans, XML, HTML, and XSL technologies. The design is based on a thin HTML client/fat application architecture so that applications are easily deployable and accessible by multiple enterprises. It supports comprehensive integration via XML APIs to all of the application logic.

Server Overview

Each server is designed to perform specific functions that are briefly described in the following sections. From a user's perspective, all that you need to use to access Oracle Transportation Management is a standard Web browser (see the Oracle Transportation Management Technical Architecture Guide for a list of supported browsers).

Web Server

The web server controls access to the Oracle Transportation Management user interface by servicing static content and directing servlet calls to the application server. It uses the Oracle HTTP Server (OHS) to redirect requests to the application server, receiving back HTML information for use with a supported browser.

Application Server

The application server controls the application logic and communicates that logic between the web server and the Oracle Database. The application server runs Oracle WebLogic Server software to manage system-level details and operates at the center of the Oracle Transportation Management multi-tier architecture. In this architecture, business logic is executed in the application server, rather than in client applications. The resulting "thin" client, allows the client to manage the presentation layer, the application server to manage the business logic, and the back-end data services to manage the data.

The application server can be viewed as two distinct components, running in a single WebLogic application:

- A **User Interface (UI)** component that responds to servlet requests from the web server. The UI Component is built on the WebLogic servlet container and returns dynamic HTML pages to the user.
- A **Business** component that performs transportation-specific functionality. The business component is built on WebLogic's J2EE implementation to control business transactions.

Prior to OTM 6.4.3, the UI component was serviced by a dedicated WebLogic domain. Communication between UI components and Business components relied on J2EE session beans. Starting with 6.4.3, both the UI and business components have been integrated into a single WebLogic domain structure.[1]

Database Server

The database server is the central repository for all the data entered in Oracle Transportation Management. The database server communicates with the application server to deliver the appropriate data to the user based on the Oracle Transportation Management application logic. The database server also communicates with the integration server to handle the import and export of Oracle Transportation Management data with other systems.

Please see the Technical Architecture Guide for the supported databases.

Integration Server

The integration server is responsible for:

- Accepting Oracle Transportation Management XML from an Enterprise Application Integration (EAI) solution.
- Persisting XML transmission to a staging table.
- Accepting requests from the Oracle Transportation Management application server to build Oracle Transportation Management XML from staged data.
- Sending Oracle Transportation Management XML to an EAI solution.

The EAI solution is used as a secure data pipeline and data transformation engine and is responsible for:

- Sending and receiving EDI flat files to and from an EDI Van.
- Transforming EDI flat files to Oracle Transportation Management XML.
- Login security (communication is permitted only from registered IP addresses).
- Encryption security (using https).
- Guaranteed delivery (only if a partner server is on the remote side).

The integration server's job is to safely store XML transmissions and transform the XML transmissions into internal tables. The Oracle Transportation Management application server is responsible for initiating jobs that move data to and from staging tables and notifies the application server when new data has arrived in the staging table. The application server must notify the Integration server when it has moved new data to the staging table.

The integration server does not perform data validation. Data validation occurs when data is transferred from the staging table to the application server tables. The staging table does not enforce referential integrity or constraints.

The Integration server and the application server communicate to the application server using Enterprise Java Beans (EJB) session bean calls. When an integration event occurs, the integration server notifies the application server using a session bean call. When the application service needs to communicate with the outside world, it does so directly, using an integration solution that provides a method to build transmissions and forward them to the EAI solution.

[1] This has a significant impact on scalability, particularly the ability to dedicated user functionality to dedicated server clusters. Please see the Scalability Guide for more information.

2. Configuring Optional Components

These components include:

- Replicated Online Database (ROD)
- Configuring Maps
- Configuring OTM to Send Faxes
- Configuring OTM to Connect to External Web Services Using SSL
- PC*Miler & PC*Miler Rail Services
- PCMiler WorldWide and PCMiler Rail
- Oracle Spatial server
- HERE Location Based Services Platform
- Rand McNally MileMaker
- User-defined External Rating and Distance Engines
- SMC RateWareXL & CarrierConnect Services

Configuring Oracle Transportation Management to Use a Replicated Online Database

To integrate an Oracle Transportation Management installation with a Replicated Online Database (ROD), append the following properties to the **CUSTOM** property set:

Table 2-1

Key	Value	Description
ods_feature	True	
glog.ods.dbserver	otmrod.company.com	Fully qualified domain name of the ROD server
glog.database.ods.sid	Otmrod	Port that Oracle is using on the ROD server (usually 1521)
glog.ods.connectstring	Otmrod	Connect string for the ROD database
glog.ods.database.port	1521	Service name of the ROD database
glog.database.ods.schema	Glogowner	

Once you make these changes, restart your Oracle Transportation Management instance.

Note that you can make the ROD property changes directly in the **Custom Properties** section of the **glog.properties** file. See Chapter 4 for more information.

Configuring Maps

By default, all OTM maps use workbench maps. Workbench maps are used in the following locations:

- PUBLIC workbench layouts which may include:
    - Dispatch Board (Fleet Management > Dispatch Board)
    - Network Workbench (Shipment Management > Itinerary Management > Network Workbench)
    - Planning Workbench (Operational Planning > Planning Workbench)
- Mapping actions which may include:
    - Map Bulk Plan Results
    - Map Fleet Bulk Plan Results
    - Map Inbound Shipments
    - Map Order Releases
    - Map Outbound Shipments
    - Map Shipments
- All user-created workbench layouts that contain a map component

See the “About Workbench Layouts” help topic for more details.

You can integrate Oracle Transportation Management with HERE Platform for Business, ALK/Trimble Maps, or Oracle Map Cloud Services (formerly eLocation) by setting the properties mentioned in the rest of this section. The installer does not configure these additional properties.

HERE

After acquiring map licensing from HERE, the following properties need to be set to enable this feature. Add the following properties to the **CUSTOM** property set:

Table 2-2

Key	Value	Description
here.app_id	appid	Application ID to connect to HERE map services
here.app_code	appcode	Application code to connect to HERE map services

The Application ID and code *must* be made in the **CUSTOM** property set. This is due to these values being secure licensing information needed to authenticate against the mapping software. These values are stored in the Oracle Transportation Management wallet (as if passwords). Any text entry in a properties file will be ignored.

HERE: https://company.here.com/here/

ALK/Trimble

After acquiring map licensing from ALK, the following property needs to be set in order to enable this feature. Add the following properties to the **CUSTOM** property set:

Table 2-3

Key	Value	Description
alk.api_key	apikey	Application key to connect to ALK map services

The key *must* be made in the **CUSTOM** property set. This is due to this value being secure licensing information needed to authenticate against the mapping software. These values are stored in the Oracle Transportation Management wallet (as if passwords). Any text entry in a properties file will be ignored.

ALK Technologies, Inc.: http://www.alk.com/

Oracle Map Cloud Services

After acquiring map licensing from Oracle Map Cloud Services, the following properties need to be set in order to enable this feature. Add the following properties to the **CUSTOM** property set:

Table 2-4

Key	**Value**	**Description**
elocation.mapviewer_url	MapViewer url	This property specifies the MapViewer URL, which connects to the map services provided by Oracle Maps Cloud Service (formerly known as eLocation). By default, this property contains no values. Once you get the map license from Oracle Map Cloud Services, set this property to https://elocation.oracle.com/mapviewer.
elocation.elocation_url	eLocation url	This property specifies the Oracle Maps Cloud Service URL, which connects to the map services provided by Oracle Maps Cloud Service (formerly known as eLocation). By default, this property contains no values. Once you get the map license from Oracle Map Cloud Services, set this property to https://elocation.oracle.com/elocation.

Oracle Map and Cloud Services: https://elocation.oracle.com

Configuring Oracle Transportation Management to Send Faxes

Fax Generation in Oracle Transportation Management

Fax generation is controlled via a plug-and-play interface, FaxHandler. Classes that implement this interface support the following two methods:

```
public MailAddress getAddress();
// returns the mail address for fax generation

public void handle(MailEvent event, FaxTopic faxTopic);
// modifies our standard text e-mail to conform to outgoing fax
// specifications.
```

The fax handler is specified via the property glog.fax.handler=<fax handler class name>

Oracle Transportation Management supplies a standard fax handler, glog.server.workflow.notify.DefaultFaxHandler, to support most fax generation software. This handler embeds routing information into one of the following 3 mail components:

- message body
- subject line
- to alias (i.e. the last name of the to party)

The embedding is accomplished with a picture string describing the modified format of the above component. You can specify original information from the text email to use in the modified component including:

- firstName
- lastName
- company
- phone
- rawPhone (punctuation or space)
- message
- subject

Replacements are represented by one of these identifiers surrounded by parenthesis; for example, {firstName}. Newline characters are preserved. To transform an email message body to one supporting FAXMaker, the picture string would be:

```
::{firstName},{company},{lastName},,{rawPhone}\n{message}
```

The following properties control the default fax handler:

```
glog.fax.defaultHandler.routingMode=<message|body|to>  (default: message)
glog.fax.defaultHandler.routingPicture=<picture string>
     (default: ::{firstName},{company},{lastName},,{rawPhone}\n{message})
```

Note that these changes for default fax handling may be made in the **CUSTOM** property set or directly in the **glog.properties** file. See Chapter 4 for more information.

Configuring OTM to Connect to External Web Services Using SSL

When OTM uses SSL to create a secure connection to an external web service, the certificate needs to be verified. The external web services we use,e.g. ALK PC*Miler Web Services, ALK PC*Miler Rail Web Services and SMC3 RateWareXL Web Services, all have the certificates with host name having wild cards like *.alk.com, *.here.com and *.smc3.com. By default, WebLogic on which OTM runs, does not support wildcard certificates. Follow the below steps for connecting to these services and enable OTM for supporting wild card certificates.

The following lines need to be added to the weblogic.conf on the application server below the last used jvm.arg statement.

- jvm.arg=-Djava.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
- jvm.arg=-Dssl.SocketFactory.provider=com.sun.net.ssl.internal.SSLSocketFactoryImpl
- jvm.arg=-DUseSunHttpHandler=true

Note: Be sure and make backup copies before making any changes.

Configuring ALK PC*Miler or PC*Miler Rail Services

To integrate an Oracle Transportation Management installation with ALK PC*Miler or PC*Miler Rail Service, append the following properties to the **CUSTOM** property set:

Table 2-5

Key	Value	Description
glog.ExternalDistanceEngine.PCMilerWS.AuthorizationKey	Authorizationkey	Authorization key to connect to ALK PC*Miler Web Services
glog.ExternalDistanceEngine.PCMilerRailWS.AuthorizationKey	Authorizationkey	Authorization key to connect to ALK PC*Miler Rail Web Services

Note that these property changes may ***not*** be made directly in the **Custom Properties** section of the **glog.properties** file. All of these properties are stored securely in the dedicated wallet (**wallet.otm**). They can be directly modified via **mkstore** where the wallet key is identical to the property key. See the Security Guide for more information on directly modifying wallets.

Configuring PCMiler WorldWide or PCMiler Rail

Install the PCMiler product as described in your PCMiler user's manual. Install the PCMiler Canada Postal Codes, if desired (PCMiler WorldWide only).

To integrate an Oracle Transportation Management installation with PCMiler WorldWide, append the following properties to the **CUSTOM** property set:

Table 2-6

Key	Value	Description
pcmiler.host	pcmiler.company.com	Fully Qualified Domain Name of the PCMiler WorldWide server
pcmiler.port	8145	PCMiler WorldWide port (usually 8145)

For PCMiler Rail, append the following properties to the **CUSTOM** property set:

Table 2-7

Key	Value	Description
pcmiler.rail.host	pcmilerrail.company.com	Fully Qualified Domain Name of the PCMiler Rail server
pcmiler.rail.port	2001	PCMiler Rail port (usually 2001)

Note: PCMiler must be configured to return its distance in miles.

Note that these changes for PCMiler products may be made directly in the **Custom Properties** section of the **glog.properties** file. See Chapter 4 for more information.

Configuring Oracle Spatial Server

To integrate an Oracle Transportation Management installation with an Oracle Spatial server for external distance engine, append the following properties to the **CUSTOM** property set:

Table 2-8

Key	**Value**	**Description**
glog.ExternalDistanceEngine.OracleSpatialEngine.protocol	http/https	Protocol to be used for connecting to Oracle Spatial Engine
OracleSpatial.host	elocation.oracle.com	Fully Qualified Domain Name of the Oracle Spatial server
OracleSpatial.port	80 (for HTTP protocol) 443 (for HTTPS protocol)	Port number of the Oracle Spatial server
OracleSpatial.US_Canada .routeServlet	/routeserver/servlet/ RouteServerServlet	The full path and name of the Route servlet for US/Canada that the Oracle Spatial server hosts
OracleSpatial.WesternEurope .routeServlet	/routeserver_eu/servlet/ RouteServerServlet	The full path and name of the Route servlet for Western Europe that the Oracle Spatial server hosts

Key	Value	Description
OracleSpatial.geocodeServlet	/geocoder/gcserver	The full path and name of the GeoCode servlet that the Oracle Spatial server hosts

Configuring HERE Location Based Services Platform

To integrate an Oracle Transportation Management installation with HERE Location based Service Platform for geo-coding, distance, and time services, append the following properties to the **CUSTOM** property set.

Specify the authentication type using the property glog.ExternalDistanceEngine.HERE.authenticationType. This takes values as either AppIdAppCode or ApiKey.

> **Note**: The authentication type of AppIdAppCode is deprecated by HERE. Any existing customers who are using app_id and app_code for authentication can set the value as AppIdAppCode. Since HERE deprecated this authentication, it is better to set the value as ApiKey.

Table 2-9

Key	Value	Value (HERE Routing)	Description
here.app_id	appid	NA	Application ID to connect to HERE Location based Services Platform
here.app_code	appcode	NA	Application code to connect to HERE Location based Services Platform

Key	Value	Value (HERE Routing)	Description
here.api_key	API Key	API Key	API Key to connect to HERE Location based Services Platform
here.route.host	https://route.api.here.com/routing/7.2/calculateroute.xml (for example)	https://router.hereapi.com/v8/routes	specify routing URL for distance and time calculation
here.geocode.host	https://geocoder.api.here.com/7.2/geocode.xml (for example)	https://geocode.search.hereapi.com/v1/geocode	specify geo-coding URL

Note that Application ID and Code property changes may ***not*** be made directly in the **Custom Properties** section of the **glog.properties** file. All three of these properties are stored securely in the dedicated wallet (**wallet.otm**). They can be directly modified via **mkstore** where the wallet key is identical to the property key. See the Security Guide for more information on directly modifying wallets.

Configuring Rand McNally MileMaker

Install the MileMaker server as described in your user’s manual.

To integrate an Oracle Transportation Management installation with MileMaker, append the following properties to the **CUSTOM** property set:

Table 2-10

Key	Value	Description
milemaker.host	otmmm.company.com	Fully Qualified Domain Name of the MileMaker server
milemaker.port	1031	MileMaker server port (usually 1031)

Note that these changes for MileMaker may be made directly in the **Custom Properties** section of the **glog.properties** file. See Section 4 for more information.

Configuring User-defined External Rating and Distance Engines

This section has been broken out into its own document. Please see the **External Programming Interface Guide** for more information.

Configuring SMC^3 RateWareXL and CarrierConnect Services

To integrate an Oracle Transportation Management installation with the SMC^3 RateWareXL or SMC^3 RateWareXL with CarrierConnect Services, append the following properties to the **CUSTOM** property set:

Table 2-11

Key	Value	Description
glog.RatingEngine.RatewareXL.Username	SomeUserName	SMC RateWareXL Username
glog.RatingEngine.RatewareXL.Password	SomePassword	SMC RateWareXL Password
glog.RatingEngine.RatewareXL.License	qwerty12345	SMC RateWareXL License Key

Note that these property changes may ***not*** be made directly in the **Custom Properties** section of the **glog.properties** file. All three of these properties are stored securely in the dedicated wallet (**wallet.otm**). They can be directly modified via **mkstore** where the wallet key is identical to the property key. See the Security Guide for more information on directly modifying wallets.

3. Starting and Stopping the Oracle Transportation Management Servers

Starting Oracle Transportation Management Servers

Oracle provides a shell script to start and stop the various components. The script may be used as-is or may be modified as needed (in which case it is recommended that you save the modified file to a different filename). The shell script is suitable for use with most `/etc/init.d`-type automatic startup environments. After installation the shell script can be found at:

- `<otm_install_dir>/install/webserver/glogweb-wl`

  Note: The script will be referred to in the instructions below by the name as given above. Please adjust the instructions given below as needed to reflect any name changes or location changes of the script being used for your specific environment.

To start Oracle Transportation Management, please do the following:

1. Start the Oracle Transportation Management database, including any listeners
2. Start the Oracle Transportation Management server
   a. Start the Oracle Transportation Management daemon (e.g. `<otm_install_dir>/install/webserver/glogweb-wl start`).
   b. It can take several minutes for the server's application server to fully start up, depending on your hardware. You can monitor the <`otm_install_path>/logs/console/console.log.0` file to determine when WebLogic has been fully initialized. Look for lines similar to the following:

```
INFO | 2015/07/21 07:31:41 | <Jul 21, 2015 7:31:41 AM PDT> <Notice>
<WebLogicServer> <BEA-000360> <The server started in RUNNING mode.>
INFO | 2015/07/21 07:31:41 | <Jul 21, 2015 7:31:41 AM PDT> <Notice>
<WebLogicServer> <BEA-000365> <Server state changed to RUNNING.>
INFO | 2015/07/21 07:31:41 | -- OTM Event: serverReady
```

3. To check to see if everything has started up, point to http://<web server name> in your web browser and log in.

   Note: While the weblogic server is starting up, OHS will respond to all requests with a '503 (Service Unavailable)' error message. This will go away once the application server has fully started up.

Stopping Oracle Transportation Management Servers

To stop Oracle Transportation Management, please do the following:

1. Stop the Oracle Transportation Management server
   a. Stop the Oracle Transportation Management daemon (e.g. `<otm_install_dir>/install/webserver/glogweb-wl stop`)
2. Stop the Oracle database, if necessary.

If you are going to restart Oracle Transportation Management, wait one minute between stopping and restarting the servers. This gives the server the time it needs to fully shut down.

4. Advanced Configuration: User-defined Properties

Much of the configuration of Oracle Transportation Management involves managing properties used by the application. These properties are initially distributed in a set of hierarchical property collections, where one collection can include another. This allows the overriding of staged properties for a particular installation.

The following types of property collections are supported:

- **Property File**: A file residing on the application server that holds property settings and instructions.
- **Property Set**: A group of records in the database where each record represents a property setting or instruction.

A property collection may include one or more child collections.

If a property is defined in more than one collection, or more than once in a single collection, the following rules determine precedence.

1. A property defined in a parent collection overrides the property defined in a child collection.
2. In a given collection, a property defined later in the collection overrides one defined earlier.

A "sibling" collection is the exception to rule #1. If a property file or property set is included as a sibling, its properties act as if they are properties in the enclosing collection. I.e., they take precedence over properties defined before the inclusion.

Property Files

A property file is a flat file where each line represents a comment, a property instruction or a property setting. The line syntax is:

Table 4-1

Line Syntax	Use	Description
#<comment>	Comment Line	
<key>=<value>	Sets a property	Sets a property value. Note that if the property key contains spaces, it needs to be enclosed in double-quotes. E.g.: "my prop"=2.
!include <file path>[,sibling]	Includes properties and instructions from another file.	Reads the specified property file from the provided file path. The path may be relative (to the current file) or absolute. If sibling is specified, the property settings in the included file take precedence over those declared before the !include instruction

Line Syntax	Use	Description
!propertySet <property set>[,sibling]	Includes properties and instructions from a property set.	Reads the specified property set from the database. If sibling is specified, the property settings in the included property set take precedence over those declared before the !propertySet instruction.
!remove <key>[=<value>]	Removes previous property settings.	Removes any previous settings for the specified property key. For multi-valued properties (see below), !remove <key> removes all values of the property. To remove a single value, include the value in the instruction.

While property files can be used to set any property, they are designed to hold properties needed to startup the servers (e.g. database connection properties) and property defaults.

Property Sets

Most properties used by Oracle Transportation Management can be configured using *property sets,* instead of property files. A property set is a collection of property instructions residing in the database rather than in a file. Property sets can be modified via the Property Set Manager. By default, this is available only to the **DBA.ADMIN** user via **Configuration and Administration** > **Property Management** > **Property Sets**.

Property sets support three instructions:

Table 4-2

Instruction	Key	Value	Description
Set	Property Key	Property Value	Sets a property value
Include	Property Set ID	[sibling]	Reads the specified property set from the database. If sibling is specified, the property settings in the included property set take precedence over those declared before the Include instruction.

Instruction	Key	Value	Description
Remove	Property Key	[Property Value]	Removes any previous settings for the specified property key. For multi-valued properties (see below), removes all values of the property. To remove a single value, include the value in the instruction.

The following property sets are included in the standard Oracle Transportation Management installation:

- **CUSTOM**. This holds modified properties, overriding staged values in the various **glog.*properties** files**/**
- **APP_WORKFLOW_THREADING**[2]. This property set is deprecated.

The **CUSTOM** property set can be used to enter updated properties via the Property Set Manager rather than directly in the file. This can be useful in an environment where business administrators have no access to the properties files but need to make property changes that impact various logic within the application.

Property Set Limitations

Property sets cannot be used to set all properties in OTM. Certain properties are needed to startup the server, setup database connections and control user security. Allowing users to set these properties within a property set could destabilize the system or relax needed security.

From the viewpoint of property sets, properties can be divided into three levels of access:

- **Dynamic**: These properties can be modified in property sets and take immediate effect.
- **Reserved**: These properties may impact the ability for the system to properly restart and read the database, or impact security rights. They cannot be modified in property sets. Reserved properties are defined by **glog.properties.reserved** metadata in the **glog.metadata.reserved.properties** and **glog.metadata.reserved.onpremise.properties** files. Note that some properties are reserved in a Cloud environment but are dynamic in an On Premise installation.
- **Internal**: These properties are reserved for internal use by Oracle. Do not modify these properties unless specifically directed by Technical Support.

[2] This property set was needed to hold workflow thread counts using the deprecated property: **glog.workflow.topicGroup**. The **APP_WORKFLOW_THREADING** property set is provided only for backward compatibility. It is deprecated as of release 6.4.2.

Reserved properties can only be changed in a property file. If staged directly to a property set outside of the application (e.g. via direct SQL), they will not be displayed in the property set manager and will be ignored. Reserved properties cannot be viewed from any screen diagnostic.

Dynamic Password properties can be modified in a property set. When viewed, the value of a password property is always masked. When modified, the value is stored in a secure Oracle Wallet, not the database table holding other property values (i.e. **PROP_INSTRUCTION**).

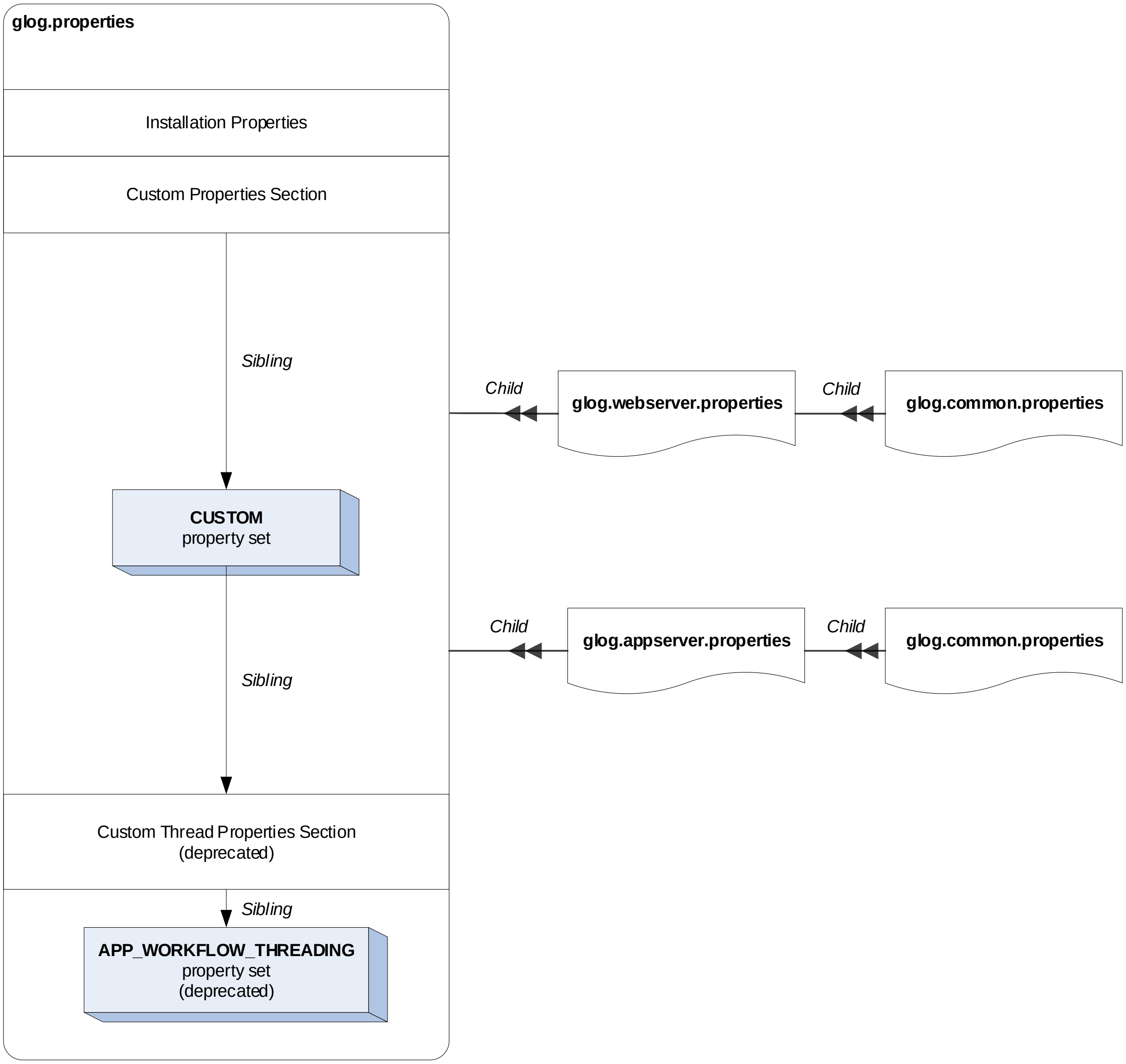

Figure 4-1: Properties Hierarchy

Property Management

Oracle Transportation Management ships with a set of staged property files and property sets, comprising a property hierarchy. The root of the hierarchy is the `glog.properties` file in the `<otm_install_path>/glog/config` directory. Figure 4-1 provides an overview of the property hierarchy:

- Properties entered in the Custom Properties section of the `glog.properties` file take precedence over installation properties and other staged properties files.
- Properties entered in the **CUSTOM** property set take precedence over those in the Custom Properties section.
- Custom Thread Properties and the **APP_WORKFLOW_THREADING** property set should only be used for backward compatibility with **glog.workflow.threadGroup** properties.

When modifying properties, use property sets whenever possible. Property sets can be modified via Oracle Transportation Management screens and, if changes are limited to dynamic properties, do not require a restart.

When making changes via a property set, use the **CUSTOM** property set or **Include** a property set specific to the implementation. This property set should be included as a sibling if you want it to override properties in the property files.

When making changes to a property file, always modify the **glog.properties** file. Changes to other files in the property tree may be overridden by a patch or upgrade.

When making changes to **glog.properties**, make changes in the **Custom Properties** section of the **glog.properties** file. This is the section in **glog.properties** after:

```
#---------------------------------------------------------------------
# Custom Properties - Beginning
#---------------------------------------------------------------------
# Place all changes, new properties, and custom properties here.
# This section will be used during GC3 Upgrades
#---------------------------------------------------------------------
```

and before:

```
#---------------------------------------------------------------------
# Custom Properties - End
#---------------------------------------------------------------------
```

This ensures property macros and multi-valued properties that you may need have been declared before your changes are processed. In addition, properties declared outside of the Custom Properties section may be overwritten in a future upgrade.

Property Values

To better understand property precedence, it's important to consider two basic types of properties in Oracle Transportation Management:

- Single-valued. This is a property that has a single value for a given key.
- Multi-valued. This is a property that supports multiple values for a given key.

While most configurable properties are single-valued, a list of multi-valued properties can be found in `glog.metaprop.multiple.properties` under `glog.properties.multiple` entries. The `glog.web.security.url.http.trustedDomain` property, for example, is multi-valued: each value specifies a trusted domain for HTTP web page redirects.

Single-Valued Properties

Assume a single-valued property is defined in more than one property collection and on more than one line in a given property collection. The declaration in the top-most property collection takes precedence over declarations in included collections[3]. The declaration lower in a property collection takes precedence over one defined above it. As an example, assume we have the following property files:

glog.properties

```
a.property=value1
!include standard.properties
a.property=value3
```

standard.properties

```
a.property=value2
```

The value of **a.property** is set to **value3** since the values in `glog.properties` take precedent over those in `standard.properties` and the line with **value3** is below the line with **value1**. Only if both declarations in `glog.properties` were removed would the value of **a.property** be set to **value2**.

Multi-Valued Properties

Multi-valued properties work in a more additive way. Assume that **a.property** was declared to be multi-valued. Then the values in both `glog.properties` and `standard.properties` are used. The values for **a.property** will be **value1**, **value2** and **value3**. If you needed to override (i.e. remove) **value2**, use the **!remove** directive. This directive can be used to remove all values declared to date or to remove a specific value. In the above example, we can remove **value2** as follows:

glog.properties

```
a.property=value1
!include standard.properties
!remove a.property=value2
a.property=value3
```

standard.properties

```
a.property=value2
```

The resulting list of values for **a.property** is now **value1** and **value3**.

Trusted Domain Example

A common example of multi-valued property use is the specification of trusted domains for HTTP redirects. The multi-valued properties that define these domains are given by:

```
glog.web.security.url.http.trustedDomain=<domain>
```

When there are more than one domain, you need to list each one as a separate property. Assume we have trusted domains A and B. Simply adding the properties:

```
glog.web.security.url.http.trustedDomain=A
glog.web.security.url.http.trustedDomain=B
```

is insufficient because the default value for HTTP trust is:

[3] Unless the file is included as a sibling. Sibling files effectively append their property settings onto the file that included them.

```
glog.web.security.url.http.trustedDomain=<all>
```

establishing an initial trust for all HTTP redirects.

To limit this trust to two domains, you must remove the initial declaration and then add the new ones:

```
!remove glog.web.security.url.http.trustedDomain=<all>
glog.web.security.url.http.trustedDomain=A
glog.web.security.url.http.trustedDomain=B
```

Note that the !remove directive only works on properties that have been loaded. It must be declared after the initial value has been included.

Property Macros

A property macro is a property whose value is used in the values for other properties. Any property can be used by another via the following syntax:

```
<property>=[<prefix>]$<macro>$[<suffix>]
```

where <macro> is a previously declared property. When evaluating property macros, the property macro must have been declared prior to its use. E.g., assume you have the following property files:

glog.properties

```
a.property=$a.macro$
!include standard.properties
another.property=$a.macro$
```

standard.properties

```
a.macro=test
```

The evaluation of **a.property** has no value for **a.macro**. It is set to **'$a.macro$'**. But the value for **another.property** is set properly to **test**.

Password Properties

As of release 6.4.2, clear text and encoded passwords are not stored in property files or property sets. Instead, password values are stored in secure wallets. See the Security Guide for more information on the use of wallets in Oracle Transportation Management.

Password properties are defined in **glog.metaprop.password.properties**, based on the metadata **glog.properties.password** entries. Any attempt to set a text or Base64 encoded value for a password property will be ignored. Instead, password properties should have a value of the form:

```
{w<wallet key>
```

where <wallet key> is a key to a wallet.

All password properties are staged with defaults pointing to the correct wallet key. To change the actual password value stored in the wallet, either:

- use the Property Set screen to modify the property. For password properties, the screen knows to store the value in the wallet rather than in the property set. The property set value is unchanged; it still points to the correct wallet key.
- use the Oracle Wallet management tool, **mkstore**, to directly modify the wallet.

Component-specific Property Values

Though the UI and Business components have been combined into a single WebLogic application, each component maintains a separate cache of property values. In rare cases, it's useful to set a property value for the UI that differs from the value for Business functions. This can be done by prefixing the property value with either **{web}** for a UI-specific value or **{app}** for a Business-specific value.

E.g., when configurable multiple servers for scalability, the **glog.scalability.thisMachineURL** must be blank for the UI component but set to the T3 URL of the server for the Business component. To do this, you would set the properties as:

```
glog.scalability.thisMachineURL={web}
glog.scalability.thisMachineURL={app}t3://myhost:8001
```

Enabling Property Sets after Upgrade from Releases Prior to 6.3.4

When migrating from versions prior to 6.3.4, property sets are not enabled by default. Entries in the **APP_CUSTOM**, **WEB_CUSTOM** and **CUSTOM** property sets are ignored.

To enable property sets for implementations migrated from these older releases, edit the `glog.properties` file for each application server and insert the line

```
!propertySet CUSTOM,sibling
```

before the line

```
!include glog.webserver.properties
```

Disabling Property Sets

Property sets can be disabled without modifying property files through the use of system properties. This can be useful when running a command-line tool that cannot access the database due to a firewall setup, but needs other OTM properties. To disable all property set loading, set the following system variable when invoking OTM or one of its tools:

```
-Dglog.propertySets.enabled=false
```

It may also be useful to allow property sets but suppress any errors generated by missing property sets. If a property set is being designed, for example, it may be helpful to add it first to the property file via a !propertySet directive and stage it later in the database. To suppress the default errors logged to the console, set

```
-Dglog.propertySets.reportErrors=false
```

More Information

For more information and complete documentation of all properties, please consult the online help topic "Property Files".

5. Advanced Configuration: Application Server

Note: Default paths are used below and may differ from your configuration.

Changing the WebLogic Configuration

There are two methods to modify WebLogic configuration in the OTM server domain:

- Directly modify the "`config.xml`" file in the Weblogic domain and restart the servers.
- Use the Weblogic console to modify configuration for an active server.

Since WebLogic configuration changes can impact the ability of Oracle Transportation Management to correctly restart and run, we recommend the following steps be taken to guarantee a workable state can be restored in event of a configuration error.

Manually Make the Changes to the "`config.xml`" File

1. Back up the original "`config.xml`" file first:

```
$ cd <bea_home_dir>/user_projects/>/domains/otm/config
$ cp config.xml  config.xml_20130417_1421
```

2. Make the changes directly to the "`config.xml`" file using the editor of your choice.
3. Restart the application.

Make Changes Using the WebLogic Console

1. Back up the original "`config.xml`" file first:

```
$ cd <bea_home_dir>/user_projects//domains/otm/config
$ cp config.xml  config.xml_20130417_1421
```

2. Make the necessary changes via the WebLogic console and click on "Activate".

Using WebLogic Connection Pools

By default, Oracle Transportation Management creates and manages all database connection pools used by the application. Most of these connection pools are small and are dedicated to specific functional use cases like security. The following pools, however, receive the majority of work in the system, including user queries and agent updates:

- PRIMARY_JTS: This pool is used for J2EE transaction management within nested session beans. It is a simple WebLogic connection pool, created on startup by Oracle Transportation Management. As a true J2EE pool, PRIMARY_JTS manages the lifecycle of its connections outside of the application.
- LOCAL_JTS: This pool is used for data updates, relying on a J2EE transaction implementation within Oracle Transportation Management. The lifecycle of connections within the local pool is completely managed by the application.

To enhance performance and/or fault tolerance, it may be useful to redirect these data sources to a connection pool created and managed by the WebLogic console. E.g., to properly account for RAC node failure, an ActiveGrid connection pool is required. The PRIMARY_JTS and LOCAL_JTS pools should ideally be redirected to an ActiveGrid pool.

Assume we want to redirect a staged data source to a WebLogic connection pool with JNDI name **MyPool**. The proper steps are:

1. Back up the `config.xml` file in the `<bea_home_dir>/user_projects/>/domains/otm/config` directory. Create and configure the MyPool connection pool within the WebLogic console.
2. Login to the application as DBA.ADMIN.
3. Create a Data Source MYPOOL for the MyPool connections (Configuration and Administration > System Administration > Data Source Manager).
4. Set the Manager for this data source to External JTS.
5. Set the JDBC URL for this data source to jndi:MyPool. The jndi: prefix informs the system that it should use JNDI to lookup the data source from WebLogic rather than create and manage a pool from the application.
6. Edit the DEFAULT Data Source Profile.
7. Reassign one or more of the Data Functions to the MYPOOL data source. Note that all of the data sources may be reassigned except for the OBJECT_LOCK_BACKUP data source. This data source is a backup in case the main connection pool is exhausted and requires a new connection be established outside of a pool.

During system startup, the WebLogic connection pools are not yet available. Any data source assigned to a **jndi:** URL is temporarily redirected to one of two application-managed connection pools. These pools are property-driven and are small. Once the server has started up, connections are retrieved from WebLogic.

Note that redirecting all data sources to an existing WebLogic connection pool can be streamlined by performing the following steps:

1. Back up the config.xml file in the `<bea_home_dir>/user_projects/>/domains/otm/config` directory.
2. Create and configure the MyPool connection pool within the WebLogic console.
3. Edit the glog.properties file and set **otm.db.url=jndi:MyPool**.
4. Make sure to set the **otm.db.server**, **otm.db.port**, **otm.db.sid**, **otm.db.user** and **otm.db.password** properties for thin connection use on startup.
5. Restart the application server(s)

With this configuration all data sources used by the system will retrieve connections from the single **MyPool** connection pool.

6. Advanced Configuration: UI/Web Server

Note: Default paths are used below and may differ from your configuration.

Note: Whenever you edit the Oracle Transportation Management property files, only edit the glog.properties file. Any changes that you make to any other files will be overwritten every time an update is applied.

Creating and Installing Secure Socket Layer (SSL) Certificates

This information has moved to the Oracle Transportation Management Security Guide.

Enabling Reverse-Proxy Support

Oracle Transportation Management supports some reverse-proxy solutions. To enable reverse-proxy support, follow the directions in the “Load Balancer” section.

Proxying OTM Resources through the OTM Web Server

Depending on which components are deployed, and how they are deployed, it may be desirable to allow access to OTM resources (Web Services, TI/GTI, and Oracle Analytics Publisher) via the OTM OHS server. This can be done by having the OTM OHS server act as a reverse proxy to the other resources.

To enable this, do the following:

1. Backup and edit the `<bea_home_dir>/user_projects/domains/$DOMAIN_NAME/config/fmwconfig/components/OHS/$OHS_COMPONENT_NAME/mod_wl_ohs.conf` file on your web server, where $WEB_DOMAIN_NAME and $OHS_COMPONENT_NAME are the values specified when the Web server was installed.
2. Locate the existing section with the `ProxyPass` and `ProxyPassReverse` directives. After the next blank line, add the following lines, depending upon which resources you would like to proxy:
    a. Web Services – the port number to use is the Application Server port number

    ```
    ProxyPass         /GC3Services  http://<app server host>:<app server port>/GC3Services
    ProxyPassReverse  /GC3Services  http://<app server host>:<app server port>/GC3Services
    ```

    b. TI/GTI – the port number to use is the Oracle Analytics Server (OAS) managed server port number

    ```
    ProxyPass         /analytics  http://<OAS host name>:<OAS port>/analytics
    ProxyPassReverse  /analytics  http://<OAS host name>:<OAS port>/analytics
    ```

    c. BIPublisher – the port number to use is the Oracle Analytics Publisher (OAP) managed server port number

    ```
    ProxyPass         /xmlpserver  http://<OAP host name>:<OAP Port>/xmlpserver
    ProxyPassReverse  /xmlpserver  http://<OAP host name>:<OAP Port>/xmlpserver
    ```

```
ProxyPass         /bi-security-login  http://<OAP host name>:<OAP
Port>/bi-security-login
ProxyPassReverse  /bi-security-login  http://<OAP host name>:<OAP
Port>/bi-security-login
ProxyPass         /biinfer  http://<OAP host name>:<OAP Port>/biinfer
ProxyPassReverse  /biinfer  http://<OAP host name>:<OAP Port>/biinfer
ProxyPass         /dv  http://<OAP host name>:<OAP Port>/dv
ProxyPassReverse  /dv  http://<OAP host name>:<OAP Port>/dv
ProxyPass         /oraclediagent  http://<OAS host name>:<OAS
port>/oraclediagent
ProxyPassReverse  /oraclediagent  http://<OAS host name>:<OAS
port>/oraclediagent
ProxyPass         /mapviewer  http://<OAS host name>:<OAS port>/
mapviewer
ProxyPassReverse  /mapviewer  http://<OAS host name>:<OAS
port>/mapviewer
```

3. Save the file and restart the OTM Web tier.
4. Backup and edit the `<bea_home_dir>/user_projects/domains/$DOMAIN_NAME/config/fmwconfig/components/OHS/$OHS_COMPONENT_NAME/httpd.conf` file on your web server, where $WEB_DOMAIN_NAME and $OHS_COMPONENT_NAME are the values specified when the Web server was installed.
5. Locate the existing line <Location /GC3>, add the following lines above it, depending upon which resources you would like to proxy
   a. Web Services – the port number to use is the Application Server port number

   ```
   <Location /GC3Services>
      Require all granted
   </Location>
   ```

   b. TI/GTI – the port number to use is the Oracle Analytics Server (OAS) managed server port number

   ```
   <Location /analytics>
      Require all granted
   </Location>
   ```

   c. BIPublisher – the port number to use is the Oracle Analytics Publisher (OAP) managed server port number

   ```
   <Location /xmlpserver>
      Require all granted
   </Location>
   <Location /bi-security-login>
      Require all granted
   </Location>

   <Location /biinfer>
      Require all granted
   </Location>
   <Location /dv>
      Require all granted
   </Location>
   <Location /oraclediagent>
      Require all granted
   </Location>
   <Location /mapviewer>
      Require all granted
   ```

```
</Location>
```

An alternative to the above the uses the mod_wl_ohs module is to do the following:

1. Back up and edit the `<bea_home_dir>/user_projects/domains/$DOMAIN_NAME/config/fmwconfig/components/OHS/$OHS_COMPONENT_NAME/mod_wl_ohs.conf` file on your web server, where $WEB_DOMAIN_NAME and $OHS_COMPONENT_NAME are the values specified when the Web server was installed.
2. Locate the existing section with the `ProxyPass` and `ProxyPassReverse` directives. After the next blank line, add the following lines, depending upon which resources you would like to proxy:
    a. Web Services – the port number to use is the Application Server port number

    ```
    <Location /GC3Services>
      SetHandler weblogic-handler
      WebLogicHost <app server host>
      WebLogicPort <app server port>
    </Location>
    ```

    b. TI/GTI – the port number to use is the OAS managed server port number

    ```
     <Location /analytics>
      SetHandler weblogic-handler
      WebLogicHost <OAS host name>
      WebLogicPort <OAS port>
     </Location>
    ```

    c. BIPublisher – the port number to use is the OAP managed server port number

    ```
    <Location /xmlpserver>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    </Location>
    <Location /bi-security-login>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    </Location>
    <Location /biinfer>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    </Location>
    <Location /dv>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    </Location>
    <Location /oraclediagent>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    </Location>
    <Location /mapviewer>
      SetHandler weblogic-handler
      WebLogicHost <OAP host name>
      WebLogicPort <OAP Port>
    ```

```
</Location>
```

3. Save the file and restart the OTM server.
4. Backup and edit the `<bea_home_dir>/user_projects/domains/$DOMAIN_NAME/config/fmwconfig/components/OHS/$OHS_COMPONENT_NAME/httpd.conf` file on your web server, where $WEB_DOMAIN_NAME and $OHS_COMPONENT_NAME are the values specified when the Web server was installed.
5. Locate the existing line <Location /GC3>, add the following lines above it, depending upon which resources you would like to proxy
   a. Web Services – the port number to use is the Application Server port number

      ```
      <Location /GC3Services>
         Require all granted
      </Location>
      ```

   b. TI/GTI – the port number to use is the Oracle Analytics Server (OAS) managed server port number

      ```
      <Location /analytics>
         Require all granted
      </Location>
      ```

   c. BIPublisher – the port number to use is the Oracle Analytics Publisher (OAP) managed server port number

      ```
      <Location /xmlpserver>
         Require all granted
      </Location>
      <Location /bi-security-login>
         Require all granted
      </Location>
      <Location /biinfer>
         Require all granted
      </Location>
      <Location /dv>
         Require all granted
      </Location>
      <Location /oraclediagent>
         Require all granted
      </Location>
      <Location /mapviewer>
         Require all granted
      </Location>
      ```

It is recommended that you use the ProxyPass approach due to issues with the hostname being incorrect for HTTP redirect results from the above servers.

Enabling Single Sign On (SSO) Support

This information has moved to the Oracle Transportation Management Security Guide.

Enabling IP Address in Log-in History

This information exists in the Oracle Transportation Management Security Guide.

Load Balancer

A hardware load balancer can be used to spread the incoming requests among multiple application servers (see the Oracle Transportation Management Scalability Guide for more details on running multiple application servers). The load balancer should be configured so that once a user logs into a given web server, that same web server is always given back to that user within some configurable time-period. After that time has elapsed, the web server given back to the user can once again be determined by current load, after which the server returned should once again be fixed for a given time-period. This is commonly referred to as a persistent or sticky session, and is usually accomplished by inserting a cookie or other header into the request. It can also be done by IP address, though this is a less desirable alternative. Exact configurations and options depend on the load balancer in use.

Some load balancers can also be configured to dynamically remove a web server from the rotation if it is down for whatever reason, and dynamically add it back in when it recovers. This is usually done by having the load balancer attempt to connect to a specific URL on the web server every N seconds or minutes, and there is a servlet provided to handle this specifically. Configure your load balancer to load the `glog.webserver.test.TestServlet` servlet, e.g.:

```
http://www.company.com/GC3/glog.webserver.test.TestServlet
```

If the servlet loads properly and returns a response code of 200, then everything is fine. If any other response code is given, or if a connection cannot be established, then that should be considered an error condition and the server should be dropped out of rotation. Test out the URL manually with a browser on a known-working system before using it to configure the load balancer; it should return the word “OK” on the screen.

In each <VirtualHost> section of the httpd.conf file, you need to change line:

```
RewriteCond %{HTTP_HOST} !otm.host.com
```

to be multiple lines that lists every acceptable FQDN host names or IP Address. For example, if it is needed to access the OTM server using URLs of http://10.1.1.2, the Load Balancer at URL https://lb.host.com and also the OTM host directly using http://otm.host.com, then

```
RewriteCond %{HTTP_HOST} !otm.host.com
```

will need to be changed to:

```
RewriteCond %{HTTP_HOST} !otm.host.com
RewriteCond %{HTTP_HOST} !lb.host.com
RewriteCond %{HTTP_HOST} !10.1.1.2
```

Terminate SSL at the Load Balancer

Complete this section to configure the system to terminate SSL at the load balancer.

> **Note**: After making this change you will only be able to log into the instance using the load balancer URL. Logging into the direct URL after the change is made will show blank screens and other access errors due the redirects used by OTM. Be sure and make backup copies of the files being modified.

1. Go to the `<OTM Home>/glog/config` directory and run the following command.

   ```
   grep ohs.config.dir glog.properties
   ```

   A directory will be returned.

2. Go to the `ohs.config.dir` and edit the `httpd.conf file`. Look for the following line.

   ```
   LoadModule ossl_module "${PRODUCT_HOME}/modules/mod_ossl.so"
   ```

3. Add the following line and comment:

   ```
   # Added to terminate SSL at the load balancer
   LoadModule certheaders_module "${PRODUCT_HOME}/modules/mod_certheaders.so
   ```

4. Go to the bottom of the same file and look for the > sections, one should show your http port at the top of the section and one for https port further down. The http section needs to be updated.

   ```
   Listen <Port>
   NameVirtualHost *:<Port>
   <VirtualHost *:<Port>>
    ServerName xxxxxxxxxx
    DocumentRoot <OTM Home>/web/htdocs
    DirectoryIndex index.htm
    <IfModule ossl_module>
     SSLEngine off
    </IfModule>
     <Directory "<OTM Home>/web/htdocs">
     Options Indexes FollowSymLinks
     AllowOverride None
     Require all granted
    </Directory>
     Header set Cache-Control maxage=21600
     ExpiresActive On
     ExpiresDefault "access plus 6 hours"
   </VirtualHost>
   ```

   After the ExpiresDefault "access plus 6 hours" line add the following lines.

   ```
   SimulateHttps On
   AddCertHeader HTTPS
   ```

   The section will look like this when done.

   ```
   Listen <Port>
   NameVirtualHost *:<Port>
   <VirtualHost *:<Port>>
    ServerName xxxxxxxxxx
    DocumentRoot <OTM Home>/web/htdocs
    DirectoryIndex index.htm
    <IfModule ossl_module>
     SSLEngine off
    </IfModule>
     <Directory "<OTM Home>/web/htdocs">
     Options Indexes FollowSymLinks
     AllowOverride None
     Require all granted
    </Directory>
     Header set Cache-Control maxage=21600
     ExpiresActive On
     ExpiresDefault "access plus 6 hours"
     SimulateHttps On
     AddCertHeader HTTPS
   </VirtualHost>
   ```

Save the changes

5. In the same directory edit the mod_wl_ohs.conf file.

```
<IfModule weblogic_module>
 # WebLogicHost <WEBLOGIC_HOST>
 # WebLogicPort <WEBLOGIC_PORT>
 # MatchExpression *.jsp
WebLogicHost <WEBLOGIC_HOST>
 DynamicServerList Off
#WebLogicPort <WEBLOGIC HTTPS Port>
WebLogicPort <WEBLOGIC HTTP Port>
<Location /GC3>
  SetHandler weblogic-handler
  WebLogicHost localhost
WebLogicPort <HTTP Port>
#WebLogicPort <HTTPS Port>
  Header set X-Content-Type-Options "nosniff"
  Header set X-XSS-Protection "1; mode=block"
#SecureProxy On
#WlSSLWallet ""
</Location>
 <Location /OTM-ADF>
  SetHandler weblogic-handler
  WebLogicHost localhost
WebLogicPort <WEBLOGIC HTTP Port>
#WebLogicPort <WEBLOGIC HTTPS Port>
  Header set X-Content-Type-Options "nosniff"
  Header set X-XSS-Protection "1; mode=block"
  ExpiresActive Off
#SecureProxy On
#WlSSLWallet ""
</Location>
</IfModule>
```

Add the following line before each of the </Location> tags.

```
WLProxySSLPassThrough ON
```

The file will look like this when done.

```
<IfModule weblogic_module>
 # WebLogicHost <WEBLOGIC_HOST>
 # WebLogicPort <WEBLOGIC_PORT>
 # MatchExpression *.jsp
WebLogicHost <WEBLOGIC_HOST>
 DynamicServerList Off
#WebLogicPort <WEBLOGIC HTTPS Port>
WebLogicPort <WEBLOGIC HTTP Port>
<Location /GC3>
  SetHandler weblogic-handler
  WebLogicHost localhost
WebLogicPort <WEBLOGIC HTTP Port>
#WebLogicPort <WEBLOGIC HTTPS Port>
  Header set X-Content-Type-Options "nosniff"
  Header set X-XSS-Protection "1; mode=block"
#SecureProxy On
#WlSSLWallet ""
  WLProxySSLPassThrough ON
```

```
</Location>
 <Location /OTM-ADF>
  SetHandler weblogic-handler
  WebLogicHost localhost
WebLogicPort <WEBLOGIC HTTP Port>
#WebLogicPort <WEBLOGIC HTTPS Port>
  Header set X-Content-Type-Options "nosniff"
  Header set X-XSS-Protection "1; mode=block"
  ExpiresActive Off
#SecureProxy On
#WlSSLWallet ""
  WLProxySSLPassThrough ON
</Location>
</IfModule>
```

6. Save the changes.

Additional Changes for 6.5.1 using WebLogic 12.2.1.4

If your FQDN's have common elements, like a .com at the end of the FQDN, the common values they can be added outside the ( )'s.

```
xxxxxx.xxxxxxxxxx.com
yyyyyyy.yyyyyyyyyy.oraclevcn.com

RewriteCond %{HTTP_HOST} !(xxxxxx.xxxxxxxxxx|yyyyyyy.yyyyyyyyyy.oraclevcn).com
RewriteCond %{HTTP_HOST} !<IP Address of the instance> <-- This may need to be
added to allow access to the TestServlet
```

If the FQDN's do not end with anything in common the full FQDN for OTM and the Load Balancer will need to be used.

```
xxxxxx.xxxxxxxxxx.com.sa
yyyyyyy.yyyyyyyyyy.oraclevcn.com

RewriteCond %{HTTP_HOST}
!(xxxxxx.xxxxxxxxxx.com.sa|yyyyyyy.yyyyyyyyyy.oraclevcn.com)
RewriteCond %{HTTP_HOST} !<IP Address of the instance> <-- This may need to be
added to allow access to the TestServlet
```

7. Save the changes.
8. Go back two directories and you should be in the OHS directory. Do a ls -l and you should see 3 directories, instances, mbeans and another one, typically named otm643 or otm651.
9. Go into the otm643 or otm651 directory, depending on the version you are running, and make the same changes to the httpd.conf and mod_wl_ohs.conf file in this directory.
10. Log into the WebLogic console and enable the "WebLogic Plugin". You will need the "weblogic" user password to access the console. Use the following commands to retrieve the "weblogic" user password from the shared wallet.
    a. Go to the `<OTM Home>/install` directory and run this command to find the wallet location.

       ```
       grep OTM_WALLET_HOME gc3env.sh
       ```

    b. In the same directory run this command to display the "weblogic" user password.

       ```
       ./otmstore -wrl <OTM_WALLET_HOME Directory>/wallet.shared -viewEntry
       ohs.admin -viewEntry otmwls.admin
       ```

Both entries should return the same value.

c. Access the WebLogic console login screen.

```
http://<OTM FQDN>:<WEBLOGIC HTTP Port>/console
```

d. Log in as the "weblogic" user with the <Wallet Password> from the last command.

e. Once logged in you can use the following steps to enable the WebLogic Plugin.

    i. Click on the domain name in the left hand pane.

    ii. Click on "Web Applications" in the right hand pane

    iii. Click on "Lock & Edit" on the left hand pane

    iv. Click on the "WebLogic Plugin Enabled" checkbox in the right hand pane.

    v. Click on the "Save" button.

    vi. Click on the "Activate Changes" button.

11. Restart the instance and log into the instance using the load balancer URL.

Installing Translations

1. Log in as the Oracle Transportation Management user.
2. Change the directory to `<otm_install_path>/glog/oracle/script8`.
3. Run the following command: `install_lang.sh <lang id> [<country id>]`.

`where valid <lang id> and <country id> values are`:

Table 6-1

Language	Language ID	Country ID
Chinese (Simplified)	zh	CN
Chinese (Traditional)	zh	TW
French (Canadian)	fr	CA
French (European)	fr	
German	de	
Italian	it	
Japanese	ja	
Korean	ko	
Polish	pl	
Portuguese (Brazillian)	pt	BR
Russian	ru	
Spanish (European)	es	
Spanish (Latin American)	es	MX

Language	Language ID	Country ID
Thai[4]	th	
Turkish	tr	

Transportation Intelligence Translations

Oracle Transportation Intelligence translations are installed when the Transportation Intelligence application is installed; please see the Installing Other Languages for Oracle Transportation Intelligence Reports section of the Oracle Transportation Management Install Guide for more information.

Notifications Translations

To have notifications translated, set the following property in the application server's `glog.properties` file or the `CUSTOM` property set:

```
glog.notify.localizeStylesheet=true
```

Modifying the Session Timeout

This information has been moved to the Oracle Transportation Management Security Guide.

[4] The Thai translations will only be available as a post-release patch.

7. Advanced Configuration: Database

> **Note**: Default paths are used below and may differ from your configuration.
>
> **Note**: Whenever you edit the Oracle Transportation Management property files, only edit the glog.properties file. Any changes that you make to any other files will be overwritten every time an update is applied.

Warning against Modifying Data in Database

Except as explicitly noted in this and other Oracle Transportation Management documentation, the data in an Oracle Transportation Management database should *never* be added to, modified or deleted by an outside application while Oracle Transportation Management is running. This could lead to unrecoverable data corruption and is not a supported use-case.

Configuring Oracle Transportation Management to Use a Different Database

At some point, you may need to point your Oracle Transportation Management instance to another database for testing, development, or fail-over purposes. Follow the directions below to configure Oracle Transportation Management to use this new database. It is assumed that the new Oracle Transportation Management database has already been created and populated with data using the database installation directions earlier in the Installation Guide. The database should also be at the same patch level as your Oracle Transportation Management instance to prevent any incompatibility problems.

1. Shutdown your Oracle Transportation Management web and application servers as described earlier in this guide.
2. Edit the file `<otm_install_path>/glog/config/glog.properties` and change the following properties. This should be done on each Oracle Transportation Management web and application server.

   ```
   dbserver=<db_server_fqdn>
   glog.database.sid=<oracle_sid>
   glog.database.connectstring=<oracle_connectstring>
   glog.database.port=<oracle port>
   ```

   Note that these properties are reserved and cannot be set as properties in a property set.
3. Restart your Oracle Transportation Management web and application servers as described earlier in this guide.
4. You may also need to edit your tnsnames.ora files on each Oracle Transportation Management server.

Real Application Clusters (RAC)

Oracle Transportation Management may be used in conjunction with RAC by using the SCAN Listener for the RAC instance in the connection information stored in the `<otm_install_path>/glog/config/glog.properties` file on all application and web servers:

```
dbserver=<SCAN listener host info>
glog.database.sid=<SCAN listener service name>
glog.database.connectstring=<SCAN listener TNS alias>
glog.database.port=<SCAN listener port>
```

Note that these properties are reserved and cannot be set as properties in a property set.

Special configuration in the RAC instance is needed to use a SCAN Listener. Please see the Oracle Real Application Cluster documentation for more details.

Copy Database Data

When moving from initial implementation to production, or duplicate production to development, there are needs to copy the whole database or part of Oracle Transportation Management data between production and development/test environment. Depending on the purpose and requirement, copying an Oracle Transportation Management database/data can be achieved using different methods. It is recommended that Support be consulted to decide the best way of moving data.

Copy Database Files

An Oracle Database can be cloned by copying datafiles and other configuration files to the destination environment (usually another server). Procedures of this method can be found from Oracle Database support/document.

Features/Limitations:

- Simple and straightforward. There is no need to pre-create an Oracle Database in the destination machine.
- Fast. The total time needed is governed by physically transferring datafiles from source machine to destination machine.
- Whole database copy. If there are data in the source database that the client does not want to copy to the destination database, then data cleaning process should be carried out in the destination database after the copy.
- Source and destination servers must be the same platform, same operating system and same Oracle Database version between.
- Requires DBA skills.
- Requires down time in the source database if there is no hot-backup.

Oracle Database Data Pump Utility

Using the Oracle Database data pump, the entire Oracle Transportation Management database structures can be copied from one database to another. To successfully copy an Oracle Transportation Management database, following guidelines are recommended:

1. Create all tablespaces used by Oracle Transportation Management first in the target database.
2. Check if all of Oracle Transportation Management required init.ora parameters are properly set in the target database.
3. Oracle Transportation Management database roles and schema users should be created in the target database before the import. This should be done by running Oracle Transportation Management supplied SQL scripts as described below.
4. Log in to the target database as user **SYS**.

   ```
   @create_glog_users.sql     -- Make sure no errors
   ```

5. Perform export on the source database using data pump. The easiest way is to do a FULL export. But if for any reason you do not want a FULL export, you can do a schema export for GLOGOWNER, REPORTOWNER, ARCHIVE_C, ARCHIVE_C_USER, GLOGLOAD, GLOGDBA and DIR_XML_USER, and GLOBALREPORTUSER schemas. Perform import on the target database using data pump. You can do a FULL import if the export was a FULL export. Alternatively, you can do a schema import.

6. After the import finished successfully, there are post-import SQL scripts to be run in the target database. The steps to run these scripts are described in the Oracle support document "Cloning Oracle Transportation Management Database Schemas (Doc ID 1313815.1) ". It is important to follow the instructions to run the scripts. Ensure you log in to the database as the correct users when running the scripts.

Refer to support document "Cloning Oracle Transportation Management Database Schemas (Doc ID 1313815.1)" for detail information related to copying database using data pump.

Domain Copy

Oracle Transportation Management provides utilities to copy domains between Oracle Transportation Management databases. There are two approaches to copy domains. The first one uses a PL/SQL procedure to generate INSERT statements with domain data to be copied from the source database. The insert statements can be run in the target database. The second approach uses Oracle Database TABLE mode export/import with WHERE clause to move domain data between source and destination databases. Please refer to the Oracle Transportation Management document titled "Data Management Guide" for details of this method.

Features/Limitations:

- Can be used between different platforms and different operating systems.
- Oracle Database version in the target environment must be equal or higher than that in the source environment if using export/import approach.
- Requires pre-creating an Oracle Database and installing an Oracle Transportation Management database in the target environment.
- Oracle Transportation Management table structures must be identical between the source and target database.
- Can just move one or more domain data between the databases.

Using WebLogic GridLink Data Source

Oracle Transportation Management can be configured to use GridLink Data Source defined in WebLogic.

Note: This configuration requires you to stop and start Oracle Transportation Management.

In order to set up to use an external data source, you need to do the following steps. Oracle Transportation Management may not start if all steps are not completed.

1. Create Data Source in WebLogic Console
2. Update Database for Data Source

Create Data Source in WebLogic Console

- Navigate to the WebLogic console.
- Login as system user.
- Follow WebLogic guidelines to Configure GridLink for RAC Data Source.
  - DO NOT create the Data Source name with prefix PRIMARY.
  - Remember the name of the Data Source and the JNDI name of this Data Source. In this example, the name of the Data Source is CUSTOM_DS and the JNDI name of the Data Source is CUSTOM

Update Database for Data Source

Create a new data source and assign data functions to this data source using SQL scripts.

Note: See the Changing Passwords section of the Oracle Transportation Management Security Guide for more information on encoding password for user glogdba.

```
insert into data_source
(data_source_gid,data_source_xid,data_database_gid,manager,jdbc_driver,jdbc_
url,schema,oracle_user,oracle_password,vpd_user,is_profiling_leaks,is_multi_
user,domain_name) values ('CUSTOM_DS','CUSTOM_DS','GC3','EXTERNAL
JTS','oracle.jdbc.OracleDriver','jndi:CUSTOM','glogowner','glogdba','<encode
d password>','DBA.ADMIN','N','Y','PUBLIC');

update data_source_profile_d set data_source_gid = 'PRIMARY_THIN' where
data_function_gid = 'STARTUP';

update data_source_profile_d set data_source_gid = 'CUSTOM_DS' where
data_source_gid = 'PRIMARY_JTS';

commit;
```

Multiple Oracle Transportation Management Application Server Instances

The new data source created using WebLogic console will not be available for all Oracle Transportation Management application instances. Repeat the above steps for each application server with same data source and JNDI name, target value will be different according to application server.

Note: Do not copy `config.xml` or the jdbc configuration file from one server to another. The target name has to match the target name for the specific Oracle Transportation Management instance.

Validate Oracle Transportation Management with Externally Defined Data Source

If Oracle Transportation Management does not start, then follow instructions to validate and reset the Oracle Transportation Management Data Source configuration.

- The new data source should not have any records in the CONNECTION_POOL table.
  - This query should return 0 rows. Replace the custom data source name to match the name defined in OTM earlier

    ```
    select count(1) from connection_pool where data_source_gid =
    'CUSTOM_DS' ;
    ```

- WebLogic console has options to test the Data Source. This test should be successful.

Reset Configuration

If there are errors and Oracle Transportation Management does not start, use these instructions to reset the configuration.

- Shutdown the Oracle Transportation Management instance.
- Run the following SQL scripts:

  ```
  update data_source_profile_d set data_source_gid = 'PRIMARY_JTS' where
  data_source_gid = 'CUSTOM_DS';

  delete from data_source where data_source_gid = ' CUSTOM_DS';

  commit;
  ```

- Restore the backup of the `config.xml` file. This file should not have any JDBC configuration

- Start the Oracle Transportation Management instance.

Edition Enabled Database Schemas

Oracle Transportation Management Database use edition "OTM643" as a default edition instead of ORA$BASE.

All of the OTM tables have been converted into Editioned Views. Original tables have been renamed to the table name with the "_T" suffix to facilitate this conversion.

Refer to "Using Edition-Based Redefinition" in Oracle Database Development guide for more information.

User Defined Database Schema

User may create additional schema on the database to store custom objects such as table, procedure, function or package. User defined schema need to be enabled to use edition for its objects to reference any OTM object.

Following SQL commands enable edition for user defined schema.

As SYS user

```
alter user <user-defined-schema> enable editions;
grant use on edition OTM643 to <user-defined-schema>;
```

In addition, user may want to create synonyms and grant privileges to read data from OTM editioning views.

```
Create synonym <user-defined-schema>.<Editioning view> for <OTM
schema>.<Editioning view>;
```

As OTM schema owner

```
Grant read on <OTM schema>.<Editioning view> to <user-defined-schema>;
```

Last two SQL need to be repeated for all required views. List of views can be obtained by querying dba_editioning_views.

Example: User defined schema MYSCHEMA’s package read data from SHIPMENT and ORDER_RELEASE views.

As SYS,

```
alter user MYSCHEMA enable editions;
grant use on edition OTM643 to MYSCHEMA;

Create synonym MYSCHEMA.SHIPMENT for GLOGOWNER.SHIPMENT;
Create synonym MYSCHEMA.ORDER_RELEASE for GLOGOWNER.ORDER_RELEASE;
```

As GLOGOWNER,

```
Grant read on GLOGOWNER.SHIPMENT to MYSCHEMA;
Grant read on GLOGOWNER.ORDER_RELEASE to MYSCHEMA;
```

Note: Do not reference base table (name ends with "_T"), instead use editioning views. List of views can be obtained by querying dba_editioning_views.

8. Configuring Transportation Intelligence

Configuring Oracle Transportation Management

Scheduling ETL through Oracle Transportation Management Process Management

You can schedule the ETL to run using the Run ETL process in Oracle Transportation Management process manager. Run ETL run the ETL from the replicated online database (ROD) or the online transaction processing (OLTP) database to the historic database (HD).

Since OTM now supports several business intelligence applications, there are properties that must be set to specify which business intelligence application the ETL loads. Depending on the following properties, the run ETL will either load either the GTI or TI tables or both:

- `allow_advanced_analytics` is the property that controls if TI is running. If this is set to TRUE, ETL loads TI tables only.
- `isAllowedGTIAnalytics` is the property that control if GTI is running. If this is set to TRUE, ETL loads GTI tables only.
- if both properties are set to TRUE, then ETL loads both TI and GTI tables.

Running the ETL

1. Log into Oracle Transportation Management as DBA.ADMIN.
2. Go to **Transportation Intelligence > Process Management > Run ETL**.
3. Schedule/run the Run ETL process.
4. Check the logs for errors.

   Note: Before running the ETL, you must complete the steps mentioned in the following section "Enabling Oracle Transportation Intelligence Agents in Oracle Transportation Management" to enable loading objects into Transportation Intelligence.

   Note: Functionality to upload targets via OTM has been. Please refer to the TI Reference guide: Adding your own Key Performance Indicators > Upload the E_FTI_TARGET.csv and the OTM/TI online help for complete details.

Loading an Oracle Transportation Management Object into Transportation Intelligence

For Oracle Transportation Management business objects to be loaded into Transportation Intelligence, they must have a status type similar to ##_HD_LOAD_STATUS with a status value of ##_READY_TO_LOAD. For example, order base with a statue type of OB_HD_LOAD_STATUS and a status value of OB_READY_TO_LOAD. By default, this status type is set to a value of ##_NOT_READY_TO_LOAD.

> **Note**: ## is object code, Shipment code is SH, Order Base code is OB and Order Release code is OR.

Setting the Status for Existing Objects:

The Oracle Transportation Management installation provides a sample script to set the status on existing objects. This script is located at:

```
<otm_install_path>/glog/oracle/script8/advanced_analytics/load_hd_status.sql.
```

You can configure this script based on your business process and run it in OLTP database by logging in with glogowner credentials.

Setting the Status for New Objects:

Oracle Transportation Management ships with several PUBLIC automation agents to set the status of shipments, order bases, and order releases to ##_READY_TO_LOAD to allow business objects to load into Transportation Intelligence. You need to enable these automation agents as described below.

Enabling Oracle Transportation Intelligence Agents in Oracle Transportation Management

To enable the automation agents required to load Oracle Transportation Management business objects into Transportation Intelligence, complete the following:

1. Log on to Oracle Transportation Management as DBA.ADMIN.
2. Go to **Business Process Automation > Agents and Milestones > Automation Agent**.
3. Search for and activate the following automation agents:
   - LOAD_ORDER_BASE_TO_HD (Default Event: Order base created)
   - LOAD_ORDER_RELEASE_TO_HD (Default Event: Order on shipment tendered)
   - LOAD_SHIPMENT_TO_HD (Default Event: Shipment tendered)
4. Change the event if necessary.

Unloading an Oracle Transportation Management Object from Transportation Intelligence

When an object is deleted from Oracle Transportation Management, you can also remove it from Transportation Intelligence. You can use the following agent actions to mark objects as soft deleted in the HD database.

- UNLOAD SHIPMENT FROM HD
- UNLOAD ORDER RELEASE FROM HD
- UNLOAD ORDER BASE FROM HD
- UNLOAD ORDER MOVEMENT FROM HD
- UNLOAD SELL SHIPMENT FROM HD
- UNLOAD QUOTE FROM HD
- UNLOAD INVOICE FROM HD
- UNLOAD BULK PLAN FROM HD
- UNLOAD ORDER ITEM FROM HD

You can create automation agents using the agent actions listed above. An example of an automation agent which marks order releases as soft deleted in the HD database is shown below:

1. Select an Agent Type of ORDER RELEASE.
2. Select an Agent Event of ORDER - REMOVED with Restrictions of INTEGRATION, INTERNAL, or USER. You do not need to select the Pre-persist option on the Restrictions page.
3. Add a single agent action of UNLOAD ORDER RELEASE FROM HD.

Mandatory Oracle Transportation Management User Role (VPD Profile) Configuration

Identify the Oracle Transportation Intelligence users. Assign the appropriate Business Intelligence Role to each user.

Oracle Transportation Intelligence offers additional external predicates from version 6.2. These are available under the following standard VPD Profiles shipped with the Oracle Transportation Management application:

- **FTI_DEFAULT**: This VPD profile is applicable for all Oracle Transportation Intelligence users who are not service providers in Oracle Transportation Management. This includes all the external predicates available in the existing DEFAULT Oracle Transportation Management VPD profile and the new external predicates specific to the Oracle Transportation Intelligence solution’s Historical Database tables.
- **SERVPROV**: This VPD Profile is applicable for all Oracle Transportation Intelligence users who are also service providers in Oracle Transportation Management. This includes all the external predicates available in the existing SERVPROV Oracle Transportation Management VPD profile and the new external predicates specific to the Oracle Transportation Intelligence solution’s Historical Database tables.

As a result, it is mandatory that you update the user role for all your existing Oracle Transportation Intelligence users to include either the FTI_DEFAULT or SERVPROV VPD Profiles, as applicable. Performing this manual step is mandatory to ensure the proper operational behavior of the Oracle Transportation Intelligence application.

If you are using a user-configured VPD profile in Oracle Transportation Management, alternatively you must manually include your additional external predicates over the FTI_DEFAULT or SERVPROV VPD profiles as applicable.

Populating the AD_TIME Table

The AD_TIME table is not populated by the Transportation Intelligence installation. If you want to use fiscal time as a dimension, you must populate the AD_TIME table as follows:

1. Populate the following columns in the AD_TIME table in the Oracle Transportation Management database schema. These columns correlate standard calendar dates to the fiscal dates.

a.	FISCAL_YEAR	VARCHAR2(50)
b.	FISCAL_QUARTER_ID	VARCHAR2(50)
c.	FISCAL_MONTH_ID	VARCHAR2(50)
d.	FISCAL_WEEK_ID	VARCHAR2(50)
e.	FISCAL_DAY	VARCHAR2(50)

2. Run the ETL.

Configuring Transportation Intelligence with OAM (SSO)

To configure single sign-on with Oracle Access Manager (OAM) in Oracle Analytics Server (OAS), make the following changes.

Enable Lightweight SSO as follows:

1. Stop the servers by running:

```
cd <BI Domain home>/domains/bi/bitools/bin
./stop.sh
```

2. Start `wlst.sh` from `ORACLE_HOME/oracle_common/common/bin/wlst.sh` and run the following command

```
enableBISingleSignOn('<BI Domain home>/domains/bi','<oam_log_off_url>')
```

3. Start the servers by running:

```
cd <BI Domain home>/domains/bi/bitools/bin
```

```
./start.sh
```

Create an Oracle Access Manager (OAM) asserter as follows:

4. Login to the WebLogic console.
5. Go to Security Realms > myrealm > Providers > New.
6. Select **OAM Identity Asserter** from the type drop down with available type as *OAM_REMOTE_USER.*
7. Set the **Control Flag** to *REQUIRED* and save.
8. Select the Providers Specific tab.
9. Set the **SSO Header name** as *OAM_REMOTE_CN* and save.

Creating Authentication Providers (OTM username):

10. Login to the WebLogic console.
11. Go to Security Realms > myrealm > Providers > New.
12. Select *ReadOnlySQLAuthenticator* from the **Type** drop down.
13. Set the **Control Flag** to *SUFFICIENT*.
14. In the Provider Specific tab, enter the queries as follows:
    a. **Data Source Name**: Enter name of data source created.
    b. **SQL Get Users Password**: `SELECT GL_PASSWORD FROM GL_USER WHERE UPPER(USERNAME) = UPPER(?)`
    c. **SQL User Exists**:

```
SELECT UPPER(USERNAME) FROM GL_USER WHERE UPPER(USERNAME) = UPPER(?)
```

    d. **SQL List Users**:

```
SELECT UPPER(USERNAME) FROM GL_USER WHERE UPPER(USERNAME) LIKE UPPER(?)
```

    e. **SQL List Groups**:

```
SELECT
DECODE(BI_ROLE_GID,'BIADMINISTRATOR','BIAdministrators','BICONSUMER','BIC
onsumers','BIAUTHOR','BIAuthors',BI_ROLE_GID) FROM GL_USER_BI_ROLE WHERE
GL_USER_GID = (SELECT GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE
UPPER(?)) UNION SELECT DECODE(BI_ROLE_GID, 'NO ACCESS', 'NO ACCESS',
'BIConsumers') FROM GL_USER_BI_ROLE WHERE GL_USER_GID = (SELECT
GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE UPPER(?))
```

    f. **SQL Group Exists**:

```
SELECT
DECODE(BI_ROLE_GID,'BIADMINISTRATOR','BIAdministrators','BICONSUMER','BIC
onsumers','BIAUTHOR','BIAuthors',BI_ROLE_GID) FROM GL_USER_BI_ROLE WHERE
GL_USER_GID = (SELECT GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE
UPPER(?)) UNION SELECT DECODE(BI_ROLE_GID, 'NO ACCESS', 'NO ACCESS',
'BIConsumers') FROM GL_USER_BI_ROLE WHERE GL_USER_GID = (SELECT
GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE UPPER(?))
```

    g. **SQL Is Member**:

```
SELECT
DECODE(BI_ROLE_GID,'BIADMINISTRATOR','BIAdministrators','BICONSUMER','BICons
umers','BIAUTHOR','BIAuthors',BI_ROLE_GID) FROM GL_USER_BI_ROLE WHERE
GL_USER_GID = (SELECT GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE
UPPER(?)) UNION SELECT DECODE(BI_ROLE_GID, 'NO ACCESS', 'NO ACCESS',
'BIConsumers') FROM GL_USER_BI_ROLE WHERE GL_USER_GID = (SELECT GL_USER_GID
FROM GL_USER WHERE UPPER(USERNAME) LIKE UPPER(?))
```

h. **SQL List Member Groups**:

```
SELECT BI_ROLE_GID FROM ( SELECT
DECODE(BI_ROLE_GID,'BIADMINISTRATOR','BIAdministrators','BICONSUMER','BICons
umers','BIAUTHOR','BIAuthors',BI_ROLE_GID) BI_ROLE_GID,GL_USER_GID FROM
GL_USER_BI_ROLE UNION SELECT distinct DECODE(BI_ROLE_GID, 'NO ACCESS', 'NO
ACCESS', 'BIConsumers') BI_ROLE_GID,GL_USER_GID FROM GL_USER_BI_ROLE WHERE
BI_ROLE_GID IN ('SERVPROV','KPIMANAGER','LSP HOST','LSP CUSTOMER') ) WHERE
GL_USER_GID = (SELECT GL_USER_GID FROM GL_USER WHERE UPPER(USERNAME) LIKE
UPPER(?))
```

i. Un-check **Descriptions Supported** check box.

j. Clear the queries against ‘SQL Get User Description’ and ‘SQL Get Group Description’.

Important Note

Providers should be in sequence of:

1. SQL Authenticator Provider (Sufficient)
2. OAM Identity Asserter (Required)
3. Default Authentication (Sufficient)

Configuring Oracle Data Integrator (ODI)

ETL Performance Tuning

When you have very large OLTP/ROD source databases, such as ETL source, there is a chance of the ETL running slowly. You can complete the steps listed below to improve ETL performance.

> Note: These steps may slow down inserts/updates of data, so only complete this section when it is absolute necessary to optimize ETL.

Creating Indexes

First, you create indexes as follow:

1. Connect to the source database as glogowner and run the following:

   ```
   <OTM_INSTALL_HOME>/script8/fti_indexes.sql
   ```

2. Connect to the source database using a sys user and run following commands:

   ```
   alter system set optimizer_index_caching=0 scope=BOTH;
   alter system set optimizer_index_cost_adj=100 scope=BOTH;
   ```

3. Install and configure the ODI Agent in the target (hdowner) database instance machine.

Changing the ODI Agent's Heap Size

Next, you change the heap size of the ODI agent as follows:

1. Open one of the following files depending on the operating system:
    a. Linux: `/oracledi/agent/bin/odiparams.sh`.
2. Change following parameters values based on available RAM.

```
set ODI_INIT_HEAP=512m
set ODI_MAX_HEAP=1024m
```

3. Go to **Topology Manager > Physical Architecture > Oracle > ORACLE_OTM/ORACLE_FTI**.
4. Increase Array Fetch Size/Batch Update Size based on the environment. You can increase the sizes to 100 and check the performance.
5. If you observe no improvement in performance, then increase to some other numbers and try again.
6. Continue to increase these numbers until you see performance improvement. Eventually the performance increase will level off.

Configuring OAS

Setting up an OAS Agent

To set up an OAS Delivers Agent, complete the following:

Note: In Transportation Intelligence, these are called Alerts.

1. In OAS enterprise manager, set up the SMTP server. Refer to the OAS documentation for details.
2. All Business Intelligence Roles that ship with Oracle Transportation Intelligence have the correct privileges to access Oracle Delivers and create agents (alerts). However, if an Oracle Transportation Management user cannot access the Alerts page, you need to enable alerts as follows:
    a. Log into Oracle Transportation Management as DBA.ADMIN.
    b. Click **Transportation Intelligence > Alerts**.
    c. Click the **Administration** link in the upper right corner of the Oracle Business Intelligence page.
    d. Under **Security**, click **Manage Privileges**.
    e. Under **Access**, make sure the Business Intelligence Role specified on your Oracle Transportation Management user is listed next to **Access to Delivers**.
    f. Log out of Oracle Transportation Management.
3. Log back into Oracle Transportation Management as the user that will create alerts.
4. Click **Transportation Intelligence > Alerts**.
5. In the upper right of the Oracle Business Intelligence page, click the **down arrow next to Signed In As USER.ADMIN** and click **My Account**.
6. In the **My Account** page, click **Deliver Options** tab.
7. Set up a **device** and **delivery profile**. Refer to the Oracle Business Intelligence help for details.
8. Click **OK** to save the device.

9. Follow the tabs to **set up an agent**. Refer to the Oracle Business Intelligence help and/or documentation for details.

9. Configuring Global Trade Intelligence

Configuring Oracle Transportation Management

Scheduling ETL through Oracle Transportation Management Process Management

You can schedule the ETL to run using the Run ETL process in Oracle Transportation Management process manager. Run ETL run the ETL from the replicated online database (ROD) or the online transaction processing (OLTP) database to the historic database (HD).

Since OTM now supports several business intelligence applications, there are properties that must be set to specify which business intelligence application the ETL loads. Depending on the following properties, the run ETL will either load either the GTI or TI tables or both:

- `allow_advanced_analytics` is the property that controls if TI is running. If this is set to TRUE, ETL loads TI tables only.
- `isAllowedGTIAnalytics` is the property that control if GTI is running. If this is set to TRUE, ETL loads GTI tables only.
- if both properties are set to TRUE, then ETL loads both TI and GTI tables.

Running the ETL

1. Log into Oracle Transportation Management as DBA.ADMIN.
2. Go to **Global Trade Intelligence > Process Management > Run ETL**.
3. Schedule/run the Run ETL process.
4. Check the logs for errors.

   Note: All of the GTM objects included in GTI are automatically loaded.

Loading an Oracle Global Trade Management Object into Global Trade Intelligence

By default, the Global Trade Management objects supported by GTI are all loaded into GTI. Please see the Data Flow to Global Trade Intelligence online help topic for complete details.

Unloading a Global Trade Management Object from Global Trade Intelligence

You can unload (soft delete) data from the Global Trade Intelligence historical database (HD). When data is deleted from GTM/OTM, you can mark that record as deleted in the Global Trade Intelligence HD. The record remains in the HD, but is it filtered out using the GTI_DEFAULT VPD profile.

There are several PUBLIC automation agents and agent actions intended for use with Global Trade Intelligence to enable this functionality. Please see the Data Flow to Global Trade Intelligence online help topic for complete details.

Mandatory Oracle Transportation Management User Role (VPD Profile) Configuration

Identify the Oracle Global Trade Intelligence users. Assign the appropriate Business Intelligence Role to each user.

Oracle Global Trade Intelligence offers additional external predicates. These are available under the following standard VPD Profile shipped with the Oracle Transportation Management application:

- **GTI_DEFAULT**: This includes all the external predicates available in the existing DEFAULT and FTI_DEFAULT Oracle Transportation Management VPD profiles and the new external predicates specific to the Oracle Global Trade Intelligence solution’s Historical Database tables.

As a result, it is mandatory that you update the user role for all your existing Oracle Transportation Intelligence users to include the GTI_DEFAULT profiles, as applicable. Performing this manual step is mandatory to ensure the proper operational behavior of the Oracle Global Trade Intelligence application.

If you are using a user-configured VPD profile in Oracle Transportation Management, alternatively you must manually include your additional external predicates over the GTI_DEFAULT VPD profile as applicable.

Populating the AD_TIME Table

The AD_TIME table is not populated by the Global Trade Intelligence installation. If you want to use fiscal time as a dimension, you must populate the AD_TIME table as follows:

1. Populate the following columns in the AD_TIME table in the Oracle Transportation Management database schema. These columns correlate standard calendar dates to the fiscal dates.

a.	FISCAL_YEAR	VARCHAR2(50)
b.	FISCAL_QUARTER_ID	VARCHAR2(50)
c.	FISCAL_MONTH_ID	VARCHAR2(50)
d.	FISCAL_WEEK_ID	VARCHAR2(50)
e.	FISCAL_DAY	VARCHAR2(50)

2. Run the ETL.

Configuring Global Trade Intelligence with OAM (SSO)

Refer to Configuring Transportation Intelligence with OAM (SSO) earlier in this guide.

Configuring Oracle Data Integrator (ODI)

ETL Performance Tuning

When you have very large OLTP/ROD source databases, such as ETL source, there is a chance of the ETL running slowly. You can complete the steps listed in ETL Performance Tuning to improve ETL performance.

> Note: These steps may slow down inserts/updates of data, so only complete this section when it is absolute necessary to optimize ETL.

10. Advanced Configuration: General

Note: Default paths are used below and may differ from your configuration.

Note: Whenever you edit the Oracle Transportation Management property files, only edit the `glog.properties` file. Any changes that you make to any other files will be overwritten every time an update is applied.

General Instructions on Upgrading Oracle Transportation Management from One Release to Another

Unless otherwise specified in release-specific instructions, please follow these steps when upgrading to a new Oracle Transportation Management version:

1. Migration from any release earlier than 6.4.3 of Oracle Transportation Management is not supported; if not already on version 6.4.3 you must upgrade to 6.4.3 before proceeding.
2. Install Oracle Transportation Management version 6.5.1.
3. Migrate your data from the old version to the new version according to the new version's Migration Guide.

   Note: These steps should be executed on a test instance that mimics your production instance before running them in the production instance itself. This will allow you to identify any issues and construct a solution for them before affecting your production environment.

Branding

Oracle Transportation Management upgrades typically include updated web files (XSL, html, jpg, gif, JS, etc.) to fix specific reported problems and to provide minor enhancements requested by customers.

Since each Oracle Transportation Management upgrade includes a list of updated files, you should apply the upgrade to your test site, re-brand the updated files, and then test everything. Once it has passed your testing, you should apply the upgrade and your new branded files to your production site. This procedure ensures that you have all of the fixes included in the latest upgrade and that your branding is not broken on your production site.

Changing Logging Options

This information has moved to the Oracle Transportation Management Security Guide.

Changing Notification Settings

The Oracle Transportation Management server sends out a variety of notifications to users. Changing these settings involves modifying the glog.properties file on your Oracle Transportation Management application server, or modifying the `CUSTOM` property set, as described below.

```
glog.mail.smtp.host=smtp.company.com
```

This setting defines the SMTP server that Oracle Transportation Management uses when sending email and fax notifications.

```
glog.workflow.notify.advisor.email=OTMAdvisor@company.com
```

This setting defines the email address that email and fax notifications will appear to come from. This email address should be valid and this email box should be monitored, so that bounced emails and delivery failures are caught. FAXmaker will also send fax success/failure responses to this email address.

```
glog.workflow.notify.advisor.fax=800-555-1212
```

This setting defines the fax number that fax notifications appear to come from.

```
glog.workflow.notify.FAXmaker.email=fax@company.com
```

This setting defines the e-mail address of the fax server. The fax server should monitor the corresponding mailbox and convert the incoming emails to faxes.

Units of Measure

Oracle Transportation Management stores all amounts in two units of measure: the actual unit of measure and a storage default unit of measure. The storage default amount is stored in the database "BASE" columns and is used to support querying amounts (i.e. Shipment Total Weight) which have different actual units of measure. The storage default is designated by an indicator on the unit of measure for each unit of measure type (weight, volume, distance, etc.). The default for storage default uses U.S. standard units of measure.

In most cases, the unit of measure displayed on the user interface is controlled by a user preference. If a user does not have a user preference defined, the unit of measure is controlled by the Display Default indicator on the unit of measure for each unit of measure type. The default for Display Default uses U.S. standard units of measure.

The Display Default and Storage Default settings can be modified by running an action on the corresponding unit of measure. In addition, new units of measure and corresponding conversions can be created. The Unit of Measure page is located at **Configuration and Administration > Power Data > General > Unit of Measure.** This page is only available when you are logged in as DBA.ADMIN. For more information on this topic refer to the help topic "Unit of Measure".

Changing Currency Settings

> **Note**: The CSV records in the CURRENCY_EXCHANGE_RATE.csv are provided as samples and must not be used for currency conversions. You must load your own exchange rates using the CSV or Exchange Rate Inbound transmissions.

By default, Oracle Transportation Management uses US Dollars when saving costs to the database. Also by default, Oracle Transportation Management triangulates all currency conversions through US Dollars.

This example illustrates how Oracle Transportation Management stores a shipment cost record with the currency storage default set to two different currencies:

Total actual cost of shipment is 1000 JPY. If Oracle Transportation Management's currency storage default is USD (current default in all Oracle Transportation Management installations), Oracle Transportation Management stores this cost as follows:

Table 10-1

Total_actual-cost	Total-actual-cost_currency_GID	Total_Actual_cost_base
1000	JPY	7.76

If instead Oracle Transportation Management's currency storage default is GBP:

Table 10-2

Total_actual-cost	Total-actual-cost_currency_GID	Total_Actual_cost_base
1000	JPY	5.31

In the first instance, the rate of 7.76 represents the USD value of 1000 JPY converted at the current rate in Oracle Transportation Management (128.77) while in the second instance, the rate of 5.31 represents the GBP value of 1000 JPY converted at the current JPY/GBP rate in the system (188.08).

When to Change Currency Storage Default

There are two scenarios where you would like to change your currency storage default: either you only use one currency other than USD or you use multiple currencies but not USD.

Rates in One Single Currency

In this case, you only need to set your currency storage default to the currency you use. For example, Oracle Transportation Management stores a 100 GBP shipment cost as 100 in both the total cost and the total cost base fields so no currency conversion is needed.

1. Run the following insert statement (example using GBP as new storage default) as the DBA user:

   ```
   INSERT INTO UOM (TYPE, UOM_CODE, UOM_SHORT_DESCRIPTION,
   UOM_LONG_DESCRIPTION, UOM_RANK, IS_STORAGE_DEFAULT, IS_DISPLAY_DEFAULT,
   DOMAIN_NAME) VALUES ('CURRENCY', 'GBP', 'UK POUND', 'UK POUND', 1, 'Y', 'Y',
   'DBA')
   ```

Rates in Several Currencies

In this case, you need to:

1. Decide what the currency storage default is for your Oracle Transportation Management installation.
2. Set your currency storage default by running the following insert statement (example using GBP as new storage default) as the DBA user:

   ```
   INSERT INTO UOM (TYPE, UOM_CODE, UOM_SHORT_DESCRIPTION,
   UOM_LONG_DESCRIPTION, UOM_RANK, IS_STORAGE_DEFAULT, IS_DISPLAY_DEFAULT,
   DOMAIN_NAME) VALUES ('CURRENCY', 'GBP', 'UK POUND', 'UK POUND', 1, 'Y', 'Y',
   'DBA')
   ```

Oracle Transportation Management still needs currency rates to convert between the currencies you use. You can download rates from the IMF website. (This populates the DEFAULT rate in the CURRENCY_EXCHANGE_RATE table.) The problem is that all rates from the IMF are against USD. You have two alternatives to using USD centric IMF rates:

- Write an API that uses another source than the IMF to bring in exchange rates stated against your currency storage default.

  or

  Update the rates you need manually or use a CSV process while again entering your needed rates against your preferred currency. You can do this nightly, monthly or in any other frequency.
- In the `glog.`properties property file or the `CUSTOM` property set, set `glog.currency.base` to your currency storage default (e.g. EUR). This makes Oracle Transportation Management triangulate through the currency of your choice.

This means that Oracle Transportation Management will have all DEFAULT rates stated against your base currency and triangulates using your base currency.

Accounting for Missing Time Zones

The UTC pl/sql package depends on java running in Oracle to do time zone conversions. Some time zones are not accounted for in the java.util.TimeZone class. The solution to this problem involves a new (and optional) "missing_time_zone" table, which has the following structure:

Table 10-3

Column	Value
TIME_ZONE_GID	NOT NULL VARCHAR2(128)
ALT_TIME_ZONE_GID	NOT NULL VARCHAR2(128)
ADJ_FACTOR	NOT NULL NUMBER

So for example, if you had a location that specified a time zone of "FOO", and you wanted "FOO" to be treated the same as "America/New_York", you would specify a missing_time_zone record of:

```
time_zone_gid = 'FOO'
alt_time_zone_gid = 'America/New_York'
adj_factor = 0
```

The adj_factor is added to the time_zone_offset. So if the time_zone_offset for America/New_York was -5, and you wanted 'FOO' to have an offset of -4, you could set the adj_factor to 1.

Some of the missing Australian time zones are accounted for in the java code itself. The defaults established in the code for the Australian time zones can be overridden via the missing_time_zone table.

Country Code Conversion from ISO Alpha-3 Code to Alpha-2 Code

1. Prior to version 6.4.2, Oracle Transportation Management uses ISO alpha-3 code format for the country code. Country code format can be change to ISO alpha-2 code as described below. This process will change 3 character country code to 2 character country code in the Oracle Transportation Management database. Prior to running this process, ensure that the two character country code record exists on the Country Code table for the country codes which were not supplied by Oracle Transportation Management. Go to `<glog_home>/glog/oracle/script8` directory of the application server
2. Log into database as user *GLOGOWNER* using SQL PLUS and run following SQL command: `@convert_cc3_cc2.sql`

3. Review and resolve any errors shown in the log file `convert_cc3_cc2.log`.

   Note: Oracle Transportation Management uses ISO alpha-2 code format for the country code starting with 6.4.2.

Data Validation

Data Validation allows you to alter default user exit validation behavior through Oracle Transportation Management. The user exit validation is performed through pkg_user_exit package. This increases the transparency of the user exit validation. You are able to alter and compile the package.

A validation package is provided, pkg_user_exit, which only serves as a template. It is called by persistence engine and performs the post-persistence validation when the property parameter glog.persistence.callout is set to On. By default, the package does not do anything when executed. It has only one public stored procedure: to_exit.

The stored procedure delegates the validation works to other private stored procedures, which should be developed by users according to their business requirements. However, the public stored procedure in package spec should never be modified.

Process Flow

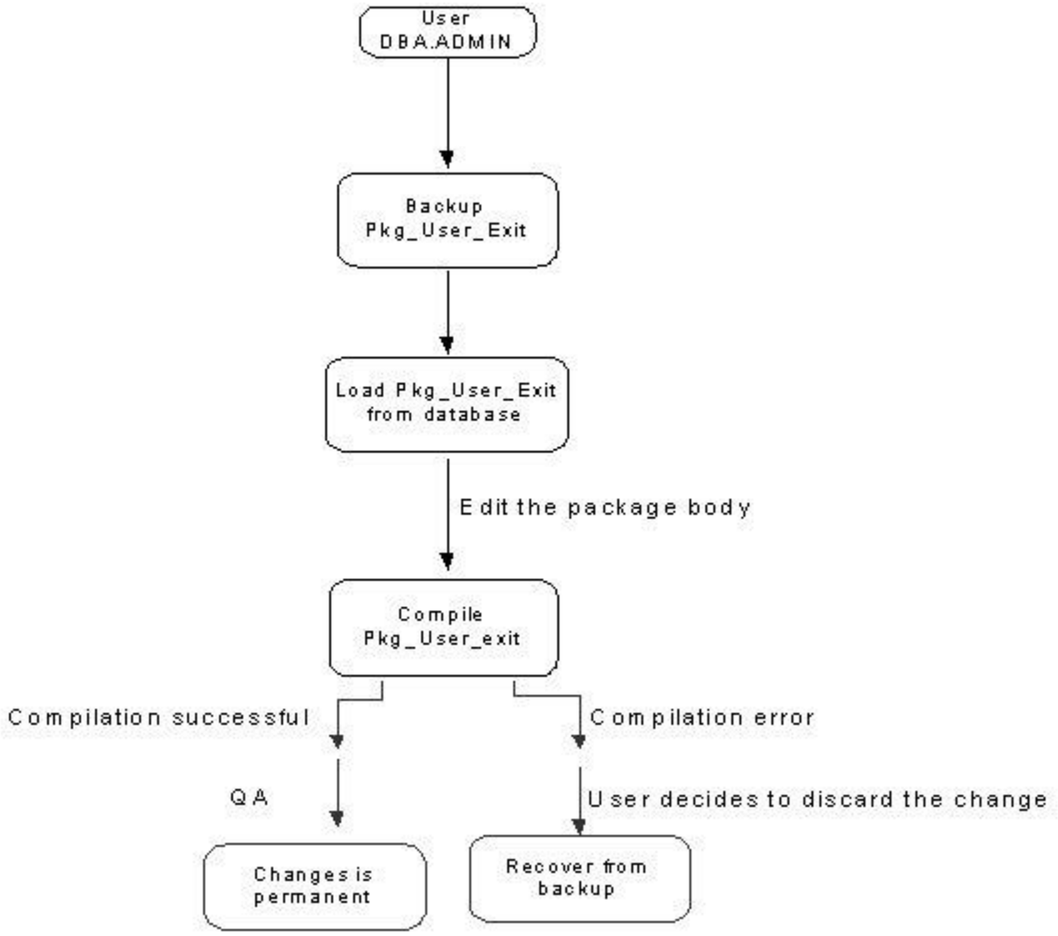

Figure 10-1: Process Flow

APIs for Compiling pkg_user_exit

UserExitCompiler Class

The UserExitCompiler should be able to load pkg_user_exit package from database and compile it to the database. Additionally, it can perform backup and recovery of the package. The backup and recovery allows you to restore the package to previously functional state when the edited package compilation is failed or the user decides to discard the altered package. To avoid security related issues, this class can only modify pkg_user_exit package, which is only restricted to package body. Furthermore, only the user DBA.ADMIN is allowed to edit the package. In the class, the following public methods will be included. The return of the methods is either status of the action or error messages when the database action is failed.

- public String get ()- Load the package from database.
- public String getBackup ()- Load backup copy from database.
- public String Backup ().
- public String Restore ().
- public String put (String packageString) - compile the package into database .

Pkg_SQL_executor Package

Class UserExitCompiler connects database through GLOGDBA. However, GLOGDBA does not have privileges to create database objects in Glogowner schema. To circumvent the issue, a new package called Pkg_SQL_executor is created to compile altered pkg_user_exit package in Glogowner schema. The privilege to execute the package is granted to GLOGDBA. The package is executed from UserExitCompiler.

HTTP Integration

Oracle Transportation Management uses HTTP to communicate with a number of downstream systems. Canonical XML data is packaged into an HTTP request and processed asynchronously through a set of workflow threads. If the downstream system is slow to respond, these threads can backup and impact system performance.

By default, the system waits indefinitely for a downstream system response. This is to minimize the number of failed XML transmissions requiring reprocessing. Timeouts, however, can be specified in two ways:

- If the transmission is directed via an External System specification, the connection and read timeouts are given by the **Socket Timeout** on the External System. If 0, the system waits indefinitely.
- If the transmission is specified with an ad-hoc URL, the following properties control timeout:

  ```
  glog.httppost.connectTimeout=<timeout in seconds>
  glog.httppost.readTimeout=<timeout in seconds>
  ```

To have the properties take precedence over the External System timeouts, set:

```
glog.httppost.allowPerSystemTimeouts=false
```

BPEL Integration

BPEL Flows

Oracle Transportation Management provides BPEL integrations with Oracle E-Business Suite. These flows can be found in `<otm_install_path>/utils/integration/bpel/flows`. Each flow should have a README.TXT file in the zip file with further instructions. There are several other flows that ship with the other Oracle products. Refer to the respective product documentation for the details for those flows.

Oracle Transportation Management Properties Files for Oracle E-Business Suite Integration through BPEL

If integrating with Oracle E-Business Suite through Oracle BPEL Process Manager, you need to include/uncomment the ebs.properties file in the glog.properties file on the Oracle Transportation Management web and application servers. These files are usually under `<otm_install_path>/glog/config`. In the `glog.properties` file, you would uncomment the following[5]:

```
#!include ebs.properties
```

Your line would look like this:

```
!include ebs.properties
```

Once this change has been made, restart your Oracle Transportation Management instance.

Oracle Transportation Management Properties Files for BPEL Integration

If integrating with the Oracle BPEL Process Manager, but not necessarily integration with Oracle E-Business Suite, you will need to enable a property on the Oracle Transportation Management web and application servers. In the `glog.properties` file or the `CUSTOM` property set, add the following:

```
glog.integration.enableParserInServlets=true
```

Once this change has been made, restart your Oracle Transportation Management instance.

Recurring Processes

This information has been moved to the Oracle Transportation Management Security Guide.

SMTP Integration

Oracle Transportation Management uses SMTP to communicate with an external mail server. Mail requests are routed to a set of dedicated threads that processes the SMTP request and response. If the mail server is slow to respond, these threads can backup and impact system performance.

[5] Note that this must be done in the `glog.properties` file. There is no method to include a property file from within the `CUSTOM` property Set.

By default, the system waits indefinitely for a mail server response. This is to minimize the number of failed e-mails. Timeouts, however, can be specified by setting the following properties:

```
glog.mail.smtp.readTimeout=<timeout in milliseconds>
glog.mail.smtp.writeTimeout=<timeout in milliseconds>
glog.mail.smtp.connectTimeout=<timeout in milliseconds>
```

Note that this section used to hold information regarding SMTP authentication. This has been moved to the Oracle Transportation Management Security Guide.

11. Advanced Configuration: LDAP

This information has been moved to the Oracle Transportation Management Security Guide.

12. Performance Monitoring

Performance monitoring is a necessary part of running a complex multi-tier application like Oracle Transportation Management. In most cases, a sudden change in performance is caused by a recent configuration change or an increase in transactional data. A formal change management process is critical for maintaining a stable production system and for quickly diagnosing performance issues should they happen. All system related configuration changes should be tested before applying to production. The date and time of the change should also be tracked. In a situation where this process is not being followed, a query of the insert/update date on important configuration tables may reveal a recent change that is causing an issue. It is also important to maintain benchmark statistics (i.e. orders and shipments processed in a typical day) to be able to quickly rule out a significant increase in data volume.

Operating System Monitoring Tools

You can monitor system performance using the `ps`, `top`, `sar`, or similar commands. You should monitor general trends, such as CPU usage, memory usage, and memory swapping (out to virtual memory). All processes for Oracle Transportation Management should run under the user setup during the installation process. This can be helpful for identifying the processes.

The Oracle HTTP Server (OHS) service starts up multiple instances of OHS to handle incoming requests. Each of these will show up as “httpd” or “httpd.worker”, and can be monitored for performance statistics using general Linux tools.

WebLogic

The WebLogic service starts up one JVM. Unfortunately this can be hard to find if you have multiple JVMs running on the same machine. Usually you can identify WebLogic by the amount of memory it is using, as normal WebLogic instances use large amounts of memory (1.6 GB or more). You can monitor the performance statistics of WebLogic using general Linux tools. You can also get detailed performance statistics from the WebLogic console -- refer to the WebLogic documentation for more information on performance monitoring.

SAR

The “`sar`” command-line utility displays CPU activity. Running “`sar 10 10`” will run the `sar` command ten times in ten second intervals. The output will show basic CPU, %nice, %system, %iowait and % idle information.

```
Linux 2.4.21-37.0.1.ELsmp (spapp01)     01/30/2008

07:24:18 AM       CPU     %user     %nice   %system   %iowait     %idle
07:24:28 AM       all      2.88      0.00      0.28      0.00     96.85
07:24:38 AM       all      3.65      0.00      0.60      0.10     95.65
```

Oracle Transportation Management activity will show up under the %user column, high activity in the user column would be expected on an unresponsive system. If you see significant utilization in the system or iowait columns it could be an indication of other internal processes or swapping memory to and from disk that may need to be investigated further.

TOP

The “`top`” command-line utility produces a frequently-updated list of processes on most Linux-based platforms. By default, the processes are ordered by percentage of CPU usage, with only the highest CPU consumers shown. The top command shows how much processing power and memory are being used. Running the “`top`” command will give you information on the system memory and swap space usage. The output on the screen is broken down into two sections, the top section tells you the current system time, up time, number of users and average load.

```
top - 08:57:58 up 97 days, 23:53, 3 users, load average: 0.06, 0.03, 0.00
Tasks: 79 total, 1 running, 78 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 2049976k total, 2034984k used, 14992k free, 1848k buffers
Swap: 2048248k total, 1021960k used, 1026288k free, 14252k cached
```

It also shows you the total amount of memory available on the system, amount used and free. The same information is shown for the swap space allocated on the system, swap used, swap free and swap cached.

The bottom section of the output shows the top 20 processes listed by PID and user name, priority and other information. For many Oracle Transportation Management instances Java will likely show up as being the highest consumer of memory and CPU on the system.

> **Note**: If you see other processes that are using significant system resources you should ask a system administrator to follow up.

Oracle Transportation Management Tools

Configuration Data Collection

The Configuration Data Collection tool captures important configuration data that can be used in diagnosing a performance issue. Users have the option of displaying the information on the screen, or they can have the data collected to them via FTP or via email.

Performance Data Collection

The Performance Data Collection utility can be used to capture performance statistics that can be used in diagnosing a performance issue. Users have the options of displaying the information on the screen or having it sent via FTP or email to a contact they have defined in the system.

Object and Method Instrumentation

This is an internal tool that OTM support and development can use to understand performance bottlenecks during bulk plan. The information gathered can only be analyzed by the development team. You can generate the instrumentation information and send it to support for development analysis.

Object instrumentation provides information regarding the memory usage of objects which is useful for understanding the memory consumption during bulk plan runs. To turn on object instrumentation, two instrumentation properties must be set as follows:

- glog.instrumentation.on = true
- glog.instrumentation.instrumentor = glog.util.instrumentation.ObjectCounts

The other two possible values for the instrumentor are:

- glog.util.instrumentation.ObjectDistribution
- glog.util.instrumentation.ObjectTrace

The instrumentor value "glog.util.instrumentation.ObjectCounts" provides current/max/total object instance counts whereas the value “glog.util.instrumentation.ObjectDistribution” provides ObjectCounts functionality along with a distribution of current objects over time. This allows identification of memory leaks. The other value "glog.util.instrumentation.ObjectTrace" provides ObjectDistribution functionality along with a distribution of current objects over distinct stack traces.

The method instrumentation can be accessed via the following servlet:

- glog.webserver.instrumentation.InstrumentationServlet

Method Profiling in OTM

To better analyze performance issues on production sites (and for complex use cases such as bulk plan), methods, subsystems, and code blocks in OTM can be profiled. Two levels of profiling are supported:

- Collection of method call counts, total time, average time, and maximum time
- Distribution of method calls by their OTM stack frame

The OTM stack frame is the collection of profiled methods called prior to execution of a particular method. For example, if bulk plan, plan orders and rating engine are profiled, the OTM frame for the rating engine might be Bulk Plan > Plan Orders.

- Collecting OTM stack frames provides two analyses of method calls:
- Counts and times by caller frame - this shows how a methods count and time are distributed among the caller methods
- Counts and times of child methods - this shows the contribution of total method time made by each profiled method it calls

The following property settings are required to enable the profiler.

- glog.instrumentation.profilerOn = true
- glog.instrumentation.profiler = glog.util.instrumentation.ProfileTimes

or

- glog.instrumentation.profiler = glog.util.instrumentation.ProfileFrameTimes

The profiler statistics can be accessed through the servlet "glog.webserver.instrumentation.ProfleServlet". This UI displays profiler information as a tree to help you identify performance bottlenecks. To collect profiler statistics for a particular bulk plan or action correctly, you will need to clear all the existing statistics on this UI prior to running the bulk plan or action. It is also important to make sure that another user is not running any processes at the same time.

Web Server Tools

Oracle HTTP Server

Please refer to the Oracle HTTP Server and Oracle WebLogic Server documentation for more information on performance monitoring.

Load Balancers

A test servlet is provided to assist in determining whether or not a web server should be dropped from or restored to a load balancer rotation.

Application Server Tools

Oracle WebLogic Server

Please refer to the Oracle WebLogic Server documentation for more information on performance monitoring.

JVM Tools

Console Logs

For the Oracle Transportation Management server, system monitoring utilities will only be partially effective since the memory of a Java application is typically static and limited by the Java heap size. They will be useful if multiple applications are being run on a single machine and the total physical memory may be limited. If the entire Java heap cannot be contained in physical memory then performance issues will be very likely due to memory swapping.

Memory monitoring within the Java heap can be performed by various Java monitoring tools or by reviewing the verbose garbage collection output in the console.log files. You should have garbage collection logging enable. This is done via one of the parameter settings in the "`weblogic.conf`" file for WebLogic on the application server. Console logs for the OTM server should be reviewed to understand the memory allocation and garbage collection.

Thread Dumps

Thread dumps are critical when diagnosing almost any performance issue. They are particularly useful when diagnosing an issue where the OTM server is unresponsive. Thread dumps capture the current state (executing, blocked, or inactive) and a stacktrace indicating the exact code that is being executed for every thread. Thread dumps should be taken in a set of three thread dumps taken consecutively at 30 second intervals. If the performance issue continues for an extended period of time, it is extremely useful to capture multiple sets of thread dumps.

If there is very little activity in the thread dumps the cause of the performance issue may be fairly apparent. However, if there are many active threads it may be very difficult to deduce the exact cause of the performance issue. In this case, capturing the thread dumps during multiple occurrences of the issue is necessary.

Heap Dumps

A heap dump is a dump of all the objects that are currently in the Java heap. The heap dump is a critical piece of information when diagnosing a memory related issues. The heap dump can be captured and then be analyzed off-line. Unfortunately the support and technique for capturing a heap dump varies by JVM and version of the JVM. Oracle Support will help with capturing a heap dump when it is necessary.

Database Monitoring Tools

Oracle Enterprise Manager includes several tools which are critical to monitoring the performance of the Oracle Database. Two of these tools are Automatic Workload Repository (AWR) and Automatic Database Diagnostic Monitor (ADDM), both of which are included in the Diagnostics Pack.

Automatic Workload Repository

By default snapshots of the relevant data are taken every hour and retained for seven days. The data is accessed via a report by running `awrrpt.sql` or `awrrpti.sql`. The automated workload repository administration tasks have been included in Enterprise Manager. The Automatic Workload Repository page is accessed from the main page via Administration > Workload Repository (under the Workload section).

Automatic Database Diagnostic Monitor

ADDM analyzes data in the Automatic Workload Repository to identify potential performance bottlenecks. For each of the identified issues it locates the root cause and provides recommendations for correcting the problem. An ADDM analysis task is performed and its findings and recommendations stored in the database every time an AWR snapshot is taken.

The best place to start viewing ADDM reports is Enterprise Manager. The Performance Analysis section on the Home page is a list of the top five findings from the last ADDM analysis task. Specific reports can be produced by clicking on the Advisor Central link, then the ADDM link. The resulting page allows you to select a start and end snapshot, create an ADDM task and display the resulting report by clicking on a few links. Alternatively the `addmrpt.sql` script can be used to create an ADDM report from SQL*Plus.

Network Monitoring

Due to the heavy traffic between the client, mid-tier, and database, a small change in network latency can have a significant impact on system performance. System monitoring tools will be helpful in diagnosing network performance issues. Generally speaking, network performance issues are one of the most difficult to identify. Maintaining benchmarks of good network performance that can be referenced during performance issues is a good practice.

Diagnosing a Performance Issue

Steps for Diagnosing a Performance Issue

1. Determine the scope of the problem.
    - Is this issue affecting one user or many users?
    - Can new users login to Oracle Transportation Management?
    - Are agents and integrations not processing?
    - Record the exact timeframe of when the issue occurred.
2. Monitor CPU, I/O, and network activity.
   Use a system monitoring tool to capture and review whether there is abnormally high CPU, network, or I/O activity on the Mid-tier/database.
3. Check the exception logs.
   Capture and review the exception.log file for unusual exceptions.
4. Review logging that is enabled.
   Review what logging is enabled via Logging Overview page and potentially suppress all logging.
5. Review workflow activity.
   Capture and review Event Diagnostics servlet or Event Queue Application Performance Diagnostic for a backlog of unprocessed events.
6. Review console.log files.
   Capture and review the console.log file looking for excessive garbage collection activity.

7. Capture thread dumps.
   Perform multiple sets of three thread dumps (30 seconds apart) on the web/applications servers.
8. Restart the server.
   If the problem persists, a restart may be necessary to resolve the issue.
9. Review change log for recent configuration changes.
   Most performance problems are caused by a recent configuration change. Review the log of recent changes and provide this information to Support.

Additional steps for Database Related Issue

1. Capture AWR/ADDM reports.
2. Review reports for SQL which may need tuning.
3. Check for stale and partial database statistics.

13. Performance Tuning

JVM

The JVMs supported by Oracle Transportation Management vary by platform. Each JVM has a set of parameters which can be used to tune the performance of the JVM. Most of these parameters affect the memory management process referred to as garbage collection. Oracle Transportation Management ships with a default set of these parameters to be used as a starting point for JVM tuning. Changing these parameters can have a significant impact on the overall performance of the system. Therefore, changes should be well documented and made incrementally with thorough performance testing.

Changing Memory Settings

As your server load varies, you may need to change the amount of memory that is allocated to the Oracle Transportation Management instance. You can change the amount of memory allocated to WebLogic for both the Web and Application servers. Increasing the amount of memory that WebLogic can utilize on the Web server allows more Oracle Transportation Management concurrent users and improves the speed of some integration tasks. Increasing the amount of memory that WebLogic can utilize on the Application server improves performance of business logic functions.

After making any memory changes, you need to stop and restart Oracle Transportation Management, as described starting on page in the chapter **Starting and Stopping the Oracle Transportation Management Servers**. The instructions below describe how to make these changes.

To change the amount of memory that WebLogic uses on the OTM server, backup and edit the `<otm_install_path>/webserver/weblogic.conf file` and modify the following lines:

```
jvm.arg=-Xms[SERVER_MEMORY]m
jvm.arg=-Xmx[SERVER_MEMORY]m
```

The string "`[SERVER_MEMORY]`" will actually be some integer value that represents the amount of memory (in megabytes) to use. For instance, if you changed this to 6.5GB, the line would read:

```
jvm.arg=-Xms6500m
jvm.arg=-Xmx6500m
```

Restart your Oracle Transportation Management instance to make these changes take effect.

Web Server

Simultaneous Users

If your installation is going to have a large number of simultaneous users it may be necessary to tune some Oracle HTTP Server and WebLogic settings. It is necessary to do performance testing and monitoring in order to tune these settings for an Oracle Transportation Management installation. Depending on the volume of users and the type of activity performed by these users it may also be necessary to setup additional webserver instances.

1. Change MaxClients in `<otm_domain>/config/fmwconfig/components/OHS/instances/otm65/httpd.conf`

The following properties in the httpd.conf file control the number processes/threads that will be used to process requests. Connection attempts beyond these limits will normally be queued, up to a number based on the "ListenBacklog" directive.

For example:

- # StartServers: number of server processes to start
- # MaxClients: maximum number of server processes allowed to start
- # MinSpareThreads: minimum number of threads which are kept spare
- # MaxSpareThreads: maximum number of threads which are kept spare
- # ThreadsPerChild: number of threads per server process
- # ServerLimit: maximum number of server processes
- # MaxRequestsPerChild: maximum number of requests a server process serves

  <IfModule mpm_worker_module>

```
StartServers  10
ServerLimit         40
MaxClients          1000
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild 0
```

See the documentation for Oracle HTTP Server for more information on these configuration settings.

Mid-Tier Server

Logging

Although logging is a vital function in Oracle Transportation Management, excessive logging is a very common cause of poor performance. This is particularly true of bulk planning processes. The Log Files and Logging Overview pages can be used to review what logs are currently enabled. You can also temporarily disable all logging by setting the following property:

```
glog.log.suppressAll=true
```

Setting this property can be a quick method of determining whether logging is the cause of a performance issue.

> **Note**: Log File and Logging Overview pages on the Configuration and Administration menu can only be used to display and configure details about log files enabled on the application server. Logging on the web server can only be controlled through `glog.properties`, the `CUSTOM` property set or Log File pages that are of type WEB.

Another way to determine if logging is impacting performance is to look at the rate at which log files are turning over. On the server running Oracle Transportation Management, you can perform a quick scan of the directory containing the Oracle Transportation Management logs files. If this reveals a log file that is turning over a 10MB log file several times per second, it is very likely to be impacting performance.

LogIds with a suffix of “Debug” or “Details” have the potential to log significant amounts of data and should be avoided unless directed to by Support. Ad-hoc logs are the most dangerous because they generate logging regardless of the user logged in. On the other hand, User logs only write to the log file when that particular user is logged in and using Oracle Transportation Management. In some scenarios user logs can still have a significant impact on performance, even if that particular user is not logged in. This logging happens because there is a certain amount of overhead in generating a log message. The overhead occurs before Oracle Transportation Management determines, based on the logged in user, that it does not need to write the message to the log file. For this reason, having many user logs with detailed logging enabled can have a significant impact on performance.

During server startup, a number of log IDs are automatically suppressed to improve startup performance and minimize initial log statements. This is controlled by the following multi-valued property:

```
glog.log.startupSuppress=<Log ID>
```

Certain errors during startup can be difficult to diagnose with these log IDs suppressed. E.g., you may want to display all SQL statements during system startup. To remove the SQL log suppression, add the following to the custom properties section of **glog.properties**:

```
!remove glog.log.startupSuppress=Sql
```

Oracle Transportation Management Business Object Caches

The Oracle Transportation Management Business Object caches are maintained by Oracle Transportation Management. The majority of Oracle Transportation Management Business Objects caches use a Least Recently Used (LRU) strategy to maintain the cache. When an LRU cache reaches its maximum, a one-for-one exchange is made for the new object and the least recently used object in the cache. Most static data used by Oracle Transportation Management business logic is maintained in one of these caches. The App-tier Caches utility page, located on the menu under Technical Support – Diagnostics and Tools – Caches, can be used to review statistics on these caches.

The size of a Business Object Cache can have a significant impact on performance. The efficiency of a cache is measured by its hit ratio. A low hit ratio is a possible indication of an undersized cache. If a cache has reached its capacity and the hit ratio is low (less than 0.80), performance may be impacted. Increasing the maximum size of this cache may increase system performance. Temporary changes can be made to the cache using the diagnostic screen, but the changes will revert to the default upon restart. To permanently change the size of a cache the appropriate glog.property must be set. For example, the size of the Rate Offering cache is set by the following property:

```
glog.cache.TRateOfferingCache.capacity=2000
```

It is important to note that increasing the size of the cache has the adverse effect of increasing memory usage, so changes should be done incrementally and with thorough testing.

Business Number Generator (BNG) Contention

The Business Number Generator is an Oracle Transportation Management mechanism for creating IDs based on a complex set of business rules. Since the IDs are based on a sequence, it is necessary to generate the IDs one at a time in order to prevent duplicate IDs. For this reason, a process may have to wait for another process to finish generating an ID. This waiting can manifest itself as a performance issue. The impact is even more significant in a scalability environment since the synchronization must be coordinated across application servers.

It is a good practice to eliminate unnecessary use of BNG. For example, the Bill of Lading 'BM' Shipment reference number can be disabled with the following property:

```
glog.shipment.createBMRefnum=false
```

The Oracle Database sequence number generator has less overhead than the BNG and should be used instead of the BNG when possible. For example, the `s_ship_unit_gid` can be changed from the BNG to an Oracle Database sequence number by setting the following property:

```
glog.server.bngenerator.oracleSequence.xid.S_SHIP_UNIT_XID.DEFAULT=S_SHIP_UNIT_
SEQUENCE
```

The following database sequences should be used instead of BNG in high volume transactional environment.

```
glog.server.bngenerator.oracleSequence.xid.S_SHIP_UNIT_XID.DEFAULT=S_SHIP_UNIT_
SEQUENCE
glog.server.bngenerator.oracleSequence.xid.SHIPMENT_XID.DEFAULT=SHIPMENT_SEQUEN
CE
glog.server.bngenerator.oracleSequence.xid.ORDER_MOVEMENT_XID.DEFAULT=ORDER_MOV
EMENT_SEQUENCE
glog.server.bngenerator.oracleSequence.xid.SHIP_UNIT_XID.DEFAULT=SHIP_UNIT_GID_
SEQUENCE
glog.server.bngenerator.oracleSequence.xid.ORDER_RELEASE_LINE_XID.DEFAULT=ORDER
_RELEASE_LINE_GID_SEQ
```

These properties follow the format shown below.

```
glog.server.bngenerator.oracleSequence.gid.<rule id>=<oracle database sequence>
```

Database sequence can be created using following steps if it is not supplied by Oracle Transportation Management.

Log into the database as glogowner using SQL PLUS and issue following SQL commands:

```
exec create_sequence.seq(P_TABLE, P_SEQUENCE,P_COLUMN);
grant select on P_SEQUENCE to APP_USER, EXT_USER;
@create_otm_synonyms.sql
```

For example, following commands will create sequence SHIP_UNIT_SEQUENCE, grant privileges, create synonyms and associate sequence SHIP_UNIT_SEQUENCE with column SHIP_UNIT_XID and table SHIP_UNIT.

```
exec create_sequence.seq('SHIP UNIT','SHIP UNIT SEQUENCE','SHIP UNIT XID');
grant select on SHIP UNIT SEQUENCE to APP USER, EXT USER;
@create_otm_synonyms.sql
```

DirectSQLUpdate

DirectSQLUpdate is an agent action which allows you to update an attribute of an Oracle Transportation Management business object using a SQL statement. It is equivalent to a database trigger, but the agent action is capable of refreshing Oracle Transportation Management business caches to reflect the modified data. Performance tuning of SQL statements in DirectSQLUpdate is important. It is also important to optimally configure the use of the cache refreshing options in the action. For any given DirectSQLUpdate, you have the choice to refresh the cache or not. When choosing to refresh the cache you have the further option to Parse SQL. If you have multiple DirectSQLUpdate actions which update the same Oracle Transportation Management business object in a single agent, you should configure the agent to only refresh the cache on the last update. An exception to this rule is if actions in the agent itself are dependent upon the modified data.

The option to Parse SQL should be used whenever possible. By default, the refresh logic needs to refresh all child data for the business object that is being updated. For instance, a DirectSQLUpdate action in a shipment agent needs to refresh the related shipment, shipment stops, ship units, etc. When SQL parsing is enabled, the agent will attempt to determine the specific business object being updated, and only refresh this object.

Workflow Thread Tuning

Oracle Transportation Management workflow is based on a set of business topics and workflow thread groups. A Topic is the Oracle Transportation Management terminology for a particular workflow process. For example, AutoMatchInvoice is a topic. Each Topic is associated with a thread group by a glog.property. For example, the following property associates the AutoMatchInvoice Topic with the agentPlanning Thread Group.

```
glog.workflow.topic.group.glog.server.agent.business.invoice.AutoMatchInvoice=a
gentPlanning
```

A Thread Group is a set of workflow threads dedicated to processing a set of workflow topics. Each thread group is given a number of threads that can be used to simultaneously process topics. When all threads in a group are busy processing a topic, all additional topics are placed in a queue until a thread becomes available. Oracle Transportation Management computes statistics for the amount of time a topic spends waiting to be processed and the amount of time spent processing the topic. These statistics can be reviewed with the Event Diagnostics page available on the DBA Diagnostics menu.

A long average wait time and/or a backlog of queued events may indicate a need to increase the number of threads in a thread group. It is possible to add or remove threads in the Event Diagnostic page[6]. Alternatively, changes to thread groups may be made with a change to the corresponding glog.workflow.queueThreads property[7]. For example:

```
glog.workflow.queueThreads.agentPlanning=4
```

This will change the number of agentPlanning threads to 4.

Adding more threads will not always increase performance. Another possibility is that some slower topics are in the same thread group as a higher volume, but short duration topic. The faster topics may be prevented from running by the slower topics. Adding more threads to this thread group may not have a significant impact on performance. However, moving the slow topics to a new thread group may improve overall system throughput by allowing the shorter topics to complete more quickly. The following properties are an example of moving a topic to a new thread group:

```
glog.workflow.queueThreads.slowEventGroup=1
glog.workflow.topic.group.glog.server.agent.business.shipment.SlowTopic=slowEve
ntGroup
```

[6] When the user adds or removes a thread from the Event Diagnostic page, he is prompted whether to persist these changes past the next system restart or temporarily change the thread count to test an alternate performance setup. If he elects to persist the changes, the **glog.workflow.queueThread.<queue>** value is automatically modified in the **CUSTOM** property set. Note that the diagnostic shows both the current # of threads and the persistent # of threads so a user can determine what changes will and will not have impact after the next restart.

[7] Note that the multi-valued properties **glog.workflow.topicGroup=**<queue>,<count> have been deprecated in favor of the **glog.workflow.queueThreads.**<queue>=<count> properties. This simplifies the specification of thread counts for a particular queue. As existing implementations may have overridden workflow thread counts in their property files or sets, the **glog.workflow.topicGroup** properties take precedence over **glog.workflow.queueThreads** properties.

When modifying workflow threads via property sets, be sure to place your properties in the APP_WORKFLOW_THREADING set.

Automatic Timeout for Stuck Threads

For any event queue, we can set an automatic timeout that effectively kills the thread and redirects work to a new thread of the same name. To use this feature, you need to set:

```
glog.eventQueueMonitor.polling=<polling interval in seconds>
glog.eventQueueMonitor.timeout.<queue>=<timeout in seconds>
```

The polling interval should generally be 1/2 the smallest timeout.

For example:

```
glog.eventQueueMonitor.polling=60
"glog.eventQueueMonitor.timeout.transport - SMTP"=120
"glog.eventQueueMonitor.timeout.transport - HTTP"=240
```

In this example, the system will check every minute on the health of the SMTP and HTTP threads. If an SMTP thread has been active longer than 2 minutes, it is interrupted and another SMTP thread is created in its place. If an HTTP thread has been active longer than 4 minutes, it is interrupted and another HTTP thread is created in its place.

Note: The double quotes around the property key since the queue names have spaces.

Multi-threading Bulk Plan Processing

Parts of the bulk plan process can be configured to process multiple threads simultaneously. Adding more threads to these thread groups may increase the performance of a very large bulk plan. The following are the two key properties to control this functionality.

```
glog.workflow.queueThreads.planningCommit=2
glog.workflow.queueThreads.planningBuild=2
```

There are additional properties to control the behavior of the multi-threading. The default values and a description of each of these properties are available in the glog.task.properties file.

Database

Analyzing Tables/Gathering Statistics

The Oracle Transportation Management database performs best when optimizer statistics are gathered regularly.

Oracle Database provides an auto optimizer stats collection task, which is enabled by default when a database is created, or when a database is upgraded from an earlier database release. This task gathers statistics on all objects in the database that have missing statistics or stale statistics (stale - the underlying object has been modified significantly, i.e. more than 10% of the rows). The auto task scheduler runs this job during a maintenance window (by default, 10pm to 6am).

Previously, Oracle Transportation Management (OTM) provided a database script (gather_table_stats.sql) to gather statistics. This script will be deprecated in the future releases. It is highly recommended to use the database-provided auto optimizer stats collection job and stop using the OTM provided script on database version 12c or higher. OTM does not provide a scheduler for the

OTM stats job, so the OTM-provided script will have to be scheduled using cron or another scheduling utility. Please check your scheduling utility to stop the OTM provided stat job from running.

How to Enable the Database Auto Stat Gather Job

1. Go to `<GLOG_HOME>/glog/oracle/script8` of the application server.
2. Log in as SYSDBA via SQL PLUS and run `@gather_stats_default_job.sql ENABLE`.

Database File Storage Performance

The two most commonly used forms of database file storage are: Storage Area Network (SAN) and Network-Attached Storage (NAS). NAS provides both storage and filesystem access whereas SAN provides only block-based storage and leaves file system concerns on the client side. Typical SAN protocols are SCSI, Fibre Channel, iSCSI, ATA over Ethernet, or HyperSCSI. A NAS on the other hand communicates over Ethernet and TCP/IP. Because NAS communicates over standard network protocols there is a stronger possibility of a poorly performing network affecting the data access rate. The latency from a database server to the NAS device should be less than 150 microseconds (0.15 ms) and preferably less than 100 microseconds (0.10 ms).

Full Export and Re-import

As data is added to tables the database may need to add additional database extents to store the data and indexes. The general consensus among database administrators is that an increasing number of extents does not affect performance of the database. However, in practice performance gains have been seen by completing a Full export and import.

SQL Tuning

Another common cause of performance issues is slow performing queries and high number of executions of queries. There are a number of mechanisms that can be used to identify these issues. The most effective is the AWR/ADDM reports provided by the Oracle Database. Another technique is to temporarily enable SQL logging in Oracle Transportation Management and review the logs for repetitive SQL statements.

Once a poorly performing SQL statement is identified, use DBMS_XPLAN procedures to get execution plans. (See the database document for further information about DBMS_XPLAN). Alternatively, an Explain Plan can be obtained within Oracle Transportation Management SQL Execution Interface by pre-pending the SQL statement with "explain plan for". Performing an explain plan outside of Oracle Transportation Management will supply more information about the query. When running it outside Oracle Transportation Management, it is important to execute the stored procedure call (`vpd.set_user()`) before running the query. Otherwise the explain plan will not be evaluated with VPD applied as it would be when run within the context of Oracle Transportation Management. Followings are the steps to get execution plans with VPD for query tuning.

Example – Query select count(*) from shipment performs poorly when run by user GUEST.ADMIN

1. Turn on application SQL log and find the exact query and vpd.set_user_r call before the query
2. Log into database as glogdba via sqlplus (do not log in as sys and change CURRENT_SCHEMA)
3. Execute vpd command as shown in sql log.
4. exec vpd.set_user_r('GUEST.ADMIN');
5. set autotrace on;
6. Run query - select count(*) from shipment

> **Note**: Query without VPD may perform well compare to with VPD because VPD adds default predicates to limit rows for the domain as well as adds custom predicate from the VPD profile if any exists. There is no point in comparing query run by DBA.ADMIN or without VPD to query with VPD (ordinary non-DBA user) because underlying queries are different. Always make sure that query performs as expected with VPD (ordinary non-DBA user) before putting it production.

An index can be added to improve query performance.

Determining the Origin of a SQL Statement

SQL statements that are part of Oracle Transportation Management originate from Oracle Transportation Management base code, user-defined saved queries, and reports. The performance tuning of user-defined reports and saved queries are the responsibility of the customer. Saved queries provide a mechanism to create a query which can be used in various parts of Oracle Transportation Management including Finders (Search pages), Business Monitors, and Automation Agents. The following query can be used to search for a SQL statement which may be generated from a saved query:

```
select saved_query_gid from saved_query where sql_find_all like ‘%querytext%’or
sql_check one like ‘%querytext %’
```

The ‘querytext’ should be just part of the SQL statement, since the SQL statement is slightly modified depending on where in Oracle Transportation Management the SQL statement is being used. For instance, the Business Monitor prefixes all SQL statements with “`select count(*)…`”.

Using CLIENT_INFO to Track SQL Statements

Each database session in Oracle provides a 64 character field to hold context information. Oracle Transportation Management can be configured to add information to the field holding information on the current or last-run SQL statement in the session. This information can include:

- The application user that issued the statement
- If from a finder or agent, the saved query name driving the statement
- The log ID of the process running the statement
- The thread name where the statement is running

By querying the **CLIENT_INFO** field of the **GV$SESSION** view, you can better determine the application source of a long-running statement.

The following table describes properties that control the tracking of SQL. By default, this tracking is turned off to avoid performance overhead.

Table 13-1

Property	Description	Default
glog.sql.clientInfo.enabled	True, if statement tracking is available.	false

Property	**Description**	**Default**
glog.sql.clientInfo.enabledUseCases	A comma-delimited list of use cases tracking statements. Valid use cases include: • **agentQuery**: agent saved queries • **uiQuery**: finder saved queries • **queryDefault**: all other queries • **directSqlUpdate**: DML statements from the DIRECT SQL UPDATE agent action • **updateDefault**: all other DML statements • **directSqlCall**: procedure calls from the DIRECT SQL UPDATE agent action • **callDefault**: all other stored procedure calls E.g., to enable statement tracking for agent and finder queries only, set `glog.sql.clientInfo.enabledUseCases=agentQuery,uiQuery`	uiQuery, agentQuery, directSqlUpdate, directSqlCall
glog.sql.clientInfo.pattern	A pattern string to map data into the CLIENT_INFO field. The pattern should be in the form: `<Information Type>[:<max length>][,<Information Type>[:<max length>]]...` where the Information Type is one of: • **user** • **savedQuery** • **logId** • **thread** and the Max Length is the maximum character length to copy from the information. Note that fields are truncated from the ends to include the most relevant information possible unless the Max Length is negative. An example pattern might be: **user:20,savedQuery:40,logId:-99999**	user:20,savedQuery:20, logId,thread:-99999
glog.sql.clientInfo.separator	The separator character used between Information Types in the CLIENT_INFO value.	|
glog.sql.clientInfo.useXid	True, if users and saved queries should suppress their domain names.	false

Property	Description	Default
glog.sql.clientInfo.saveLastInfo	By default, the CLIENT_INFO field is cleared once the query or DML statement completes. If this property is set to true, the CLIENT_INFO field is not cleared; it represents the last statement run in the session.	false

Using Timeouts with SQL Statements

To avoid performance delays associated with long-running SQL statements, Oracle Transportation Management supports optional timeouts for SQL queries, updates and stored procedure calls. These timeouts can be varied by use case.

The following properties control SQL timeouts:

```
glog.sql.query.timeout.<query use case>=<# of seconds>
glog.sql.update.timeout.<update use case>=<# of seconds>
glog.sql.call.timeout.<procedure use case>=<# of seconds>
```

If set to 0, the statement will never time out.

The following table summarizes valid uses cases by statement type:

Table 13-2

Statement Type	Use Case	Description	Default, in seconds
Query	uiQuery	Finders, Managers,Business monitors	300
Query	agentQuery	Agent conditions, IF agent action	900
Query	default	All other query statements	1800
Update	directSqlUpdate	DIRECT SQL UPDATE agent action	1800
Update	default	All update statements	1800
Call	directSqlUpdate	DIRECT_SQL_UPDATE agent action	1800
Call	default	All other stored procedure calls	1800

INIT.ORA Parameters

The values of INIT.ORA tuning parameters can have a significant impact on performance. The Oracle Transportation Management Installation Guide (Database Administration section) provides the latest set of recommended settings for Oracle Transportation Management. These settings are guidelines that have been established based on feedback from various customer scenarios. There is not one set of parameters that will be optimal for all installations. These parameters need to be tuned for each Oracle Transportation Management instance by an experienced Database Administrator.

Tuning a RAC Database Installation Using Services

RAC is the multi-node clustering capability for the Oracle Database. Oracle Transportation Management is certified to work in a RAC configuration. A RAC configuration will allow the database to scale, particularly in a situation where the database experiences high CPU. The default configuration of RAC is to load-balance the work across all nodes in a cluster. In some scenarios this configuration may result in significant contention between the RAC nodes due to the global buffer cache. Evidence of this issue can be seen in the AWR/ADDM reports with high wait times due to the "gc cr request" and/or "gc buffer busy" events.

An Oracle Transportation Management large, multi-threaded bulkplan process is a scenario that may experience this problem when attempting to commit a large number of shipments across multiple nodes in a cluster. It is possible to achieve a performance improvement by dedicating a particular Oracle Transportation Management process to a sub-set of the nodes in the cluster in order to reduce contention. This can be accomplished by defining services on the RAC nodes and then associating the service in the database connect string of particular nodes in the Oracle Transportation Management Application Server. This assumes that Oracle Transportation Management has been configured to use Oracle Transportation Management scalability and that particular processes are being handled by particular applications servers in the Oracle Transportation Management cluster.

The following is an example of how these services could be defined and used:

Table 13-3

Service Name	Primary Instance(s)	Available Instance(s)
otm_oltp	rotm021, r0tm022	rotm023
otm_batch	rotm023	rotm021, rotm022

To use these services, users need to configure new connect strings in the corresponding glog.properties file[8] for each application server like the following:

otm_oltp

```
glog.database.dbaOnly.t2client.connectionURL=jdbc:oracle:thin:@(DESCRIPTION=(AD
DRESS=(PROTOCOL=TCP)(HOST=otm-rac01-
vip)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=otm-rac02-
vip)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=otm-rac03-
vip)(PORT=1521))(LOAD_BALANCE=yes)(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME
=otm_oltp)))\
```

[8] Note these properties are reserved and cannot be added via property sets.

otm_batch

```
glog.database.dbaOnly.t2client.connectionURL=jdbc:oracle:thin:@(DESCRIPTION=(AD
DRESS=(PROTOCOL=TCP)(HOST=otm-rac01-
vip)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=otm-rac02-
vip)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=otm-rac03-
vip)(PORT=1521))(LOAD_BALANCE=yes)(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME
=otm_batch)))
```

Hash Partitioning Indexes on RAC Databases

Oracle Transportation Management installations on the RAC database may experience high cluster waits such as "gc buffer busy" on the indexes for the columns whose data is populated by database sequence generated number. Converting such indexes into hash partitioned index reduces cluster waits. Perform following steps to convert about twenty eight indexes into hash partition. Oracle Transportation Management application and web servers should be off line during this procedure.

> **Note**: Converting regular indexes into hash partitioned index may affect range scan on the corresponding column. If user community is heavily searching column data by range then you may want to keep index same as it is.

1. Make sure all constraints are valid by running following query on the database.

   ```
   SELECT owner,  status, constraint_type, COUNT(*) FROM dba_constraints WHERE
   owner IN ('GLOGOWNER','REPORTOWNER') GROUP BY owner, status,
   constraint_type;
   ```

2. Shut down all web and application servers.
3. Go to the Application servers `<otm home>/glog/oracle/script8` directory. From the script8 directory log into the database as "glogower" using SQLPlus and run the following SQL command:

   ```
   @gen_hash_indexes.sql
   ```

   - This command will create script create_hash_indexes.sql in the same directory
4. Review create_hash_indexes.sql. This file will have about 28 indexes for hash partitioning. You may remove indexes which are not good candidate for hash partitioning.
5. Using SQL PLUS, log in as GLOGOWNER and run

   ```
   @create_hash_indexes.sql
   ```

6. Review generated log file create_hash_indexes.log and make sure that there are no errors.
7. Also query data dictionary to ensure that hash partitoned indexes are created, and all primary key and foreign key constraints are valid.

   ```
   SELECT table_name, index_name FROM dba_part_indexes WHERE owner IN
   ('GLOGOWNER','REPORTOWNER');

   SELECT owner,  status, constraint_type, COUNT(*) FROM dba_constraints WHERE
   owner IN ('GLOGOWNER','REPORTOWNER') GROUP BY owner, status,
   constraint_type;
   ```

8. Restart all web and application servers.

Changing Database Pool Size

As your server load varies, you may need to change the number of connections in the database pool. This may increase the performance of your Oracle Transportation Management instance.

After making any database pool changes, you may need to stop and restart Oracle Transportation Management, as described in the chapter **Starting and Stopping the Oracle Transportation Management Servers**. Also, before you change your pool size, consult a qualified Oracle DBA and ensure that your database can handle the new number of connections. If you set your pool connections higher than your database can support, Oracle Transportation Management will not be able to startup and function correctly. The instructions below describe how to make these changes.

1. Log into your Oracle Transportation Management system (`http://<otm_webserver_name>`) as the user DBA.ADMIN.
2. In the left frame expand Configuration and Administration > System Administration > Data Source Manager.
3. Select search.
4. Select the desired database pool.
5. You can now edit the initial capacity, maximum capacity, and capacity increment. If resources permit, you should set the initial and maximum capacity to the same value and save some overhead involved with expanding and shrinking the database pool.

Business Data Purge

Oracle Transportation Management business data can be purged from the database when the data gets old, or depending on business data requirements. Purging old and unused business data may enhance run time and migration performance as well as saving disk space. The following types of business data can purged:

- SHIPMENT
- ORDER
- ITEM
- SKU
- INVOICE
- WORK INVOICE
- DEVICE ASSOCIATION
- DEMURRAGE TRANSACTION
- GTM TRANSACTION
- GTM DATA VERSION
- DOCUMENT
- P_BID
- QUOTE

Business Data Purge is divided into two separate processes; both processes can be accessed via the Oracle Transportation Management user interface by logging in as an application administrator (DBA.ADMIN):

- Mark business data for purge (Mark For Purge): This process marks business objects for purge. You can mark or unmark business objects prior to executing data purge. As an application administrator, you can mark business objects for purge based on the age of the data, status information, or domain. You can set up the marking process to happen immediately, during the next scheduled process, or on a specific day and time.
- Data Purge: This process purges data marked for purge at the time of execution. It inserts data into tables located on ARCHIVE_C schema before deleting.

For more information on the Mark for Purge and Data Purge actions, see the online help topics "Mark For Purge", "Schedule Purge", and "Data Purge".

The Data Purge process can be configured using logic parameters associated with it. Logic configuration purge parameters are accessed via the Logic Configuration page if you select a logic configuration type of Purge Configuration. The options you select determine how Oracle Transportation Management performs a data purge. For a list of the parameters used to configure a data purge, see the "Logic Configuration - Purge Configuration" topic in the online help.

More information about archive is available in the "Archive Data" section.

Partitioned Tables

Because Oracle Transportation Management contains several integration and logging tables that can become quite large very quickly, these tables have been partitioned to allow for quick purges of older data. By partitioning the tables, a particular partition (segment) can be truncated, instead of records being individually deleted, which is inefficient for large amounts of data.

Range Partitioning

Here is an example of partitioning, or segmenting, a table where a certain year and month drive the partition the data should be placed in. This is an example of partitioning by range:

Table 13-4

Range	Partition
Jan-Mar 2001	sales_q1
Apr-Jun 2001	sales_q2
Jul-Sep 2001	sales_q3
Oct-Dec 2001	sales_q4

```
CREATE TABLE sales
          ( invoice_no NUMBER,
                sale_year  INT NOT NULL,
                sale_month INT NOT NULL)
    PARTITION BY RANGE (sale_year, sale_month)
          ( PARTITION sales_q1 VALUES LESS THAN (2001, 04)
                      TABLESPACE data,
                PARTITION sales_q2 VALUES LESS THAN (2001, 07)
                      TABLESPACE data,
                PARTITION sales_q3 VALUES LESS THAN (2001, 10)
                      TABLESPACE data,
                PARTITION sales_q4 VALUES LESS THAN (2002, 01)
                      TABLESPACE data );
```

Oracle Transportation Management Partitioned Tables

The Oracle Transportation Management database has groups of partitioned tables. Most of them are range partitioned. However, the ranges are not hard-coded. Instead, a partition_key column has been added to each partitioned table, which is populated with a trigger. The value of the calculated key determines into which partition the data should be placed, based upon the INSERT_DATE. This adds to flexibility in how the tables are segmented.

For integration related partitioned tables, each table has been defined as a quarterly, a monthly, weekly, or a daily table. The frequency to which the table has been associated drives how the partitions are populated, and also the frequency of the purge. Each of these tables has four partitions (except for I_transaction and I_transmission). I_TRANSACTION has sixteen partitions, four for inbound transmissions, four for outbound transmissions, seven for mobile messages and one for tender related outbound transmissions. I_TRANSMISSION has fifteen partions, four inbound for inbound transmissions, four for outbound transmission and seven for mobile messages.

Table 13-5

Quarterly	Monthly	Weekly	Daily
	EXPLANATION	I_TRANSACTION (outbound)	I_TRANSACTION (mobile)
	GL_LOGIN_HISTORY	I_TRANSMISSION (outbound)	I_TRANSMISSION(mobile)
	IE_INROUTE_SERVPROV		I_MESSAGE
	IE_MARKS		I_MESSAGE_ACK
	IE_SHIPMENTSTATUS		I_LOG (mobile)
	IE_SHIPMENT_REFNUM		I_LOG_DETAIL (mobile)
	IE_SHIP_GROUP_REFNUM		I_TRANSACTION_ACK (mobile)
	IE_SS_EQUIPMENT		I_TRANSACTION_DETAIL (mobile)
	IE_SS_REMARK		I_TRANSACTION_REFNUM (mobile)
	IE_SS_STOP		I_TRANSMISSION_ACK (mobile)
	IE_SS_S_EQUIPMENT		I_TRANSMISSION_REFNUM (mobile)
	IE_S_SHIP_UNIT		I_TRANSMISSION_REPORT (mobile)
	IE_S_SHIP_UNIT_LINE		
	I_LOG		
	I_LOG_DETAIL		
	I_TRANSACTION_DETAIL		
	PROBLEM		
	I_TRANSACTION (inbound)		
	I_TRANSMISSION (inbound)		

Here is how it works:

If the table is defined as quarterly, then one quarter will go into each partition. In other words,

Table 13-6

Quarter	Partition
Jan-Mar	partition 1
Apr–Jun	partition 2
Jul–Sep	partition 3
Oct–Dec	partition 4

If the table is monthly, then the data is segmented as follows:

Table 13-7

Month	Partition
Jan	partition 1
Feb	partition 2
Mar	partition 3
Apr	partition 4
May	partition 1
June	partition 2
…	

If the table is weekly, then the data is divided as follows:

Table 13-8

Week	Partition
Jan 1 –7	partition 1
Jan 8-14	partition 2
Jan 15- 21	partition 3
Jan 22-28	partition 4
Jan 29-Feb 4	partition 1

Week	Partition
Feb 5–11	partition 2
…	

Every table reuses its partitions, because the intention is that before the end of the cycle, the oldest partition is purged in preparation for the new cycle. In other words, for a monthly table, on April 30th, partition 1 should be purged to remove January's data, which will then be used for May.

Truncate Jobs for Partitioned Tables

In order to make truncating the partitioned tables as seamless as possible, a script is provided that submits an Oracle Database job to automatically purge the oldest partition, on the last day of the cycle. These jobs should be submitted as a final step in any migration, logged in as **GLOGOWNER**.

1. Enter SET SERVEROUTPUT ON to be able to see feedback.
2. Run Exec partit.submit_job('WW');
3. Run Exec partit.submit_job('MM');
4. Run Exec partit.submit_job('Q');
5. Run Exec partit.submit_job('D');

These jobs are set to run at 1 AM on the last day of the cycle by default.

> **Note**: I_TRANSACTION table's OUTBOUND5 partition stores tender data. OUTBOUND5 partition will not be truncated by this job; Tender data follows lifecyle of SHIPMENT data hence it will be deleted by OTM shipment purge process "mark for purge". If Tender data needs to be deleted during the shipment purge process, then change the logic configuration parameter PURGE_TENDER_TRANSACTION to TRUE from the default value of FALSE. It can also manually purged with sql command "`exec domainman.orphan_tender_purge( <total_minutes>,<batch size>);`"

Schedule Jobs at a Different Time

You can specify a different time if you would like, as an optional second parameter. For example, the following command will have the job run at 2:30am. You can resubmit these jobs whenever you need to change the runtime, since it will auto-delete the previous job, and create it again with the new time.

```
SQL> exec partit.submit_job('WW','02:30');
Job 1164 has been submitted.
It will run next on 13-MAY-2002 02:30
```

Remove Jobs

If you need to remove the job, you can execute the partit.remove_job procedure:

```
SQL> EXEC PARTIT.REMOVE_JOB('WW');
PL/SQL procedure successfully completed.
```

Since the job is submitted as a standard Oracle job, you may want to read Oracle Database documentation about Oracle Database jobs, and the DBMS_JOB package that we utilize in our procedures.

If you need to manually run the purge, you can do so by running purge_partitioned_tables.sql logged in as GLOGOWNER. You will be prompted for the frequency. You may want to run this script if you are not automating the purge, or if it fails and you would like to run it manually. For example, it can look like this:

```
SQL>  @purge_partitioned_tables
```

This routine will truncate the oldest partition for each table that has been identified to be purged at the inputted frequency. The valid values are 'Q' for quarterly, 'MM' for monthly, or 'WW' for weekly, or 'D' for daily. By purging the old data, it can prepare room for the next week, month, or quarter.

```
Please enter a frequency (Q, MM, WW, D): MM

Please enter a frequency (Q, MM, WW, D): MM

Purging partition 2...
    Purged partition 2 for table I_TRANSMISSION_REPORT_T
    Purged partition 2 for table I_TRANSMISSION_REFNUM_T
    Purged partition 2 for table I_TRANSMISSION_ACK_T
    Purged partition 2 for table I_TRANSACTION_REFNUM_T
    Purged partition 2 for table I_TRANSACTION_DETAIL_T
    Purged partition 2 for table I_TRANSACTION_ACK_T
    Purged partition 2 for table I_LOG_T
    Purged partition 2 for table GL_LOGIN_HISTORY_T

   Purged inbound i_transaction_t records
   Purged inbound i_transmission_t records

PL/SQL procedure successfully completed.

SQL> @purge_partitioned_tables

This routine will truncate the oldest partition for each table that has been
identified to be purged
at the inputted frequency. The valid values are 'Q' for quarterly, 'MM' for
monthly,
or 'WW' for weekly, or 'D' for daily.
By purging the old data, it can prepare room for the next week, month, or
quarter.

Please enter a frequency (Q, MM, WW, D): D

Purging partition 3...
    Purged partition 3 for table EXPLANATION
    Purged partition I_LOG_DET_MSG_PART3 for table I_LOG_DETAIL
    Purged partition I_LOG_MSG_PART3 for table I_LOG
    Purged partition I_LOG_DET_MSG_PART3 for table I_LOG_DETAIL
    Purged partition I_MSG_PART3 for table I_MESSAGE
    Purged partition I_MSG_ACK_PART3 for table I_MESSAGE_ACK
    Purged partition I_TRANSACTION_MSG_PART3 for table I_TRANSACTION
    Purged partition I_TRNSCTN_ACK_MSG_PART3 for table I_TRANSACTION_ACK
    Purged partition I_TRNSCTN_DET_MSG_PART3 for table I_TRANSACTION_DETAIL
    Purged partition I_TRNSCTN_RFNM_MSG_PART3 for table I_TRANSACTION_REFNUM
    Purged partition I_TRANSMISSION_MSG_PART3 for table I_TRANSMISSION
    Purged partition I_TRNSMSN_ACK_MSG_PART3 for table I_TRANSMISSION_ACK
    Purged partition I_TRNSMSN_RFNM_MSG_PART3 for table I_TRANSMISSION_REFNUM
    Purged partition I_TRNSMSN_REP_MSG_PART3 for table I_TRANSMISSION_REPORT

PL/SQL procedure successfully completed.
```

Archive Data

The “Business Data Purge” process purges business data; it stores purged data into ARCHIVE_C schema before deleting. Archived data is stored in compressed format to provide disk space saving. All of the ARCHIVE_C tables are partitioned by year (yearly interval partitioning). New partitions will be automatically added as data gets inserted on yearly basis.

> **Note**: ARCHIVE_C schema does not store data purged by any other purge process except “Business Data Purge”.

Here is an example of ARCHIVE_C table which stores data purged from SHIPMENT_STATUS table.

```
  CREATE TABLE "ARCHIVE_C"."SHIPMENT_STATUS"
   (      "SHIPMENT_GID" VARCHAR2(101 CHAR),
   "STATUS_TYPE_GID" VARCHAR2(101 CHAR),
   "STATUS_VALUE_GID" VARCHAR2(101 CHAR),
   "DOMAIN_NAME" VARCHAR2(50 CHAR),
   "INSERT_DATE" DATE,
   "UPDATE_DATE" DATE,
   "INSERT_USER" VARCHAR2(128 CHAR),
   "UPDATE_USER" VARCHAR2(128 CHAR),
   "ARCH_OTM_VER" VARCHAR2(50 BYTE),
   "ARCH_TS" DATE NOT NULL ENABLE
   )
 ROW STORE COMPRESS ADVANCED
  TABLESPACE "ARCH_C_TBS"
  PARTITION BY RANGE ("ARCH_TS") INTERVAL (NUMTOYMINTERVAL(1, 'YEAR'))
 (PARTITION "PART_0"  VALUES LESS THAN (TO_DATE(' 2016-05-02 00:00:00', 'SYYYY-
MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  ROW STORE COMPRESS ADVANCED
 TABLESPACE "ARCH_C_TBS" ) ;
```

Another database user ARCHIVE_C_USER has been provided; it has read only privilege for all ARCHIVE_C and GLOGOWNER tables. ARCHIVE_C_USER should be used for querying archive data or developing any user-defined report or application that access archived data.

Archive Data Purge

Purge data gets accumulated in the ARCHIVE_C schema over time. There may be a time when archived data is no longer necessary and needs to be removed from the database. Oracle Transportation Management provides PL/SQL package to remove archived data from all of the ARCHIVE_C tables on a yearly basis.

```
Schema: ARCHIVE_C
Package:  PKG_ARCHIVE_C_PURGE
Procedure PKG_ARCHIVE_C_PURGE.TRUNCATE_PARTITIONS(P_MAX_ARCHIVE_AGE_IN_YEARS
varchar2)
```

This procedure truncates all ARCHIVE_C table partitions which has data older than P_MAX_ARCHIVE_AGE_IN_YEARS.

Parameter P_MAX_ARCHIVE_AGE_IN_YEARS defaults to 12 years.

Example:

The following command truncates all ARCHIVE_C table partitions which contain data older than 12 years. If this command runs in 2016 then it will truncate all partitions whose data is older than 2004 across all tables.

Log in as ARCHIVE_C

```
exec PKG_ARCHIVE_C_PURGE.TRUNCATE_PARTITIONS;
```

The following command truncates all ARCHIVE_C table partitions which contain data older than 6 years. If this command runs in 2016 then it will truncate all partitions whose data is older than 2010 across all tables.

Log in as ARCHIVE_C

```
exec pkg_archive_c_purge.TRUNCATE_PARTITIONS(6);
```

Database Jobs to purge Archived Data

There are two procedures provided to schedule database job which automatically purges old archive data; both procedures are owned by ARCHIVE_C schema.

1. Procedure name: PKG_ARCHIVE_C_PURGE.INSTALL

This procedure schedules a job named "ARCHIVE_C_PURGE" which will truncate partitions whose data is older than 12 year across all tables. The scheduled job will run on the date and time of submission every year.

Example:

This command schedules a job named "ARCHIVE_C_PURGE" which will truncate partitions whose data is older than 12 year across all tables and runs every year on the day and time of the execution of this command.

Log in as ARCHIVE_C

```
exec PKG_ARCHIVE_C_PURGE.INSTALL;
```

2. Procudure name: `PKG_ARCHIVE_C_PURGE.SCHEDULE_PURGE_JOB ( p_start_date TIMESTAMP default systimestamp, p_max_archive_age_in_years number default 12,  p_interval VARCHAR2 default 'FREQ=YEARLY;', p_job_name VARCHAR2 DEFAULT 'ARCHIVE_C_PURGE')`

This procedure schedules job as specified in the parameters.

Example:

This command schedules archive purge database job named "'CUSTOM_ARCHIVE_PURGE_JOB "which purges archive partitions which contains data older than 7 years, starts on January 1st, 2017 and runs every year after that on January 1st.

Log in as ARCHIVE_C

```
exec pkg_archive_c_purge.schedule_purge_job (to_date('01-01-2017','DD-MM-YYYY'), 7, p_job_name=>'CUSTOM_ARCHIVE_PURGE_JOB');
```

Generating Reports off of Archive Data

Reports can be designed to run off of Archive data. See the Report Designer's Guide for more details.

By default, though, only users with the **DBA.ADMIN** user role have access to archive reports. To allow other users to run archive reports, add the **Archive Reports** Access Control item to the user or user role you wish to grant access.

Pinning Large Object in Memory

This section is for a production database and an experienced DBA.

Oracle Database requires space in the System Global Area (SGA) for stored packages and functions. If SGA space is fragmented, there may not be enough space to load a package or function. Oracle Database recommends pre-allocating space in the SGA shared pool for frequently used packages, procedures and functions. Pinning objects in the shared pool can improve database performance, if it is done correctly. For large packages, when pinned they do not need to be loaded and parsed from the database again and again. Pinning large package also helps to avoid ORA-04031 error (unable to allocate xxx bytes of shared memory).

It is recommended that, in an Oracle Transportation Management production database, following objects be pinned in the SGA:

```
Package:
   VPD
   PARTIT
   PKG_PURGE
   RRL
   USER_EXIT_HELPER
   RPT_GENERAL
```

Database Space Monitoring

A DBA should periodically monitor Oracle Transportation Management database(s) space usage. There are several tables containing a column with data type of CLOB. These columns can have great amount of data. Some of the CLOB tables can be loaded quite often. One example is I_TRANSMISSION and I_TRANSACTION table. These tables contain transmission/transaction's XMLs. If there are inbound /outbound transmissions in and out of system frequently, the disk space usage of these two tables can grow very quickly.

Scheduled Jobs

Oracle Transportation Management database uses Oracle Database scheduled job to perform certain activities inside the database. Some of the jobs are set up automatically after the database is installed. The others are optional and can be implemented by a DBA. An Oracle Transportation Management database DBA should monitor these jobs regularly.

Length Semantics

Oracle Transportation Management uses character semantics for new string columns and all of the PL/SQL objects to support multi-byte character encoding scheme. It is highly recommended to use character semantics during configuration of Oracle Transportation Management database schemas. Change NLS_LENGTH_SEMANTICS parameter at session level before adding new column or compiling PL/SQL object during configurations as shown below.

```
alter session set NLS_LENGTH_SEMANTICS=CHAR;
```

For further information about `NLS_LENGTH_SEMANTICS`, please refer to *Oracle Database Globalization Support Guide*.

Optimizing Application-Database Round Trips

When querying or updating a group of records, an Oracle Transportation Management application server can be configured to retrieve or send batches of records at a time. This can significantly increase perceived query performance by reducing the network round trips.

For queries, the following properties controls the SQL fetch size

```
glog.sql.query.fetch.on = true
glog.sql.query.fetch.default = 10
glog.sql.query.fetch.maximum = 250
```

The **on** property specifies whether the application should use the default JDBC fetch size. If set to true, queries will explicitly set their fetch size based on the use case. The **default** property specifies the default fetch size for application queries. Many use cases in Oracle Transportation Management, however, set a fetch size based on criteria such as page size (see the ***User Interface*** section below). In these cases, the fetch size is often controlled by implementation details and can degrade performance. For example, that the Finder Page Size is set to 500. Aside from the delay in retrieving that many records, the overhead of having a fetch size of 500 when the number of records may be much smaller may exceed the savings in round trips. The **maximum** property provides a system-wide constraint on the fetch size.

For updates and inserts, the application can use Oracle Batch updates to minimize round trips and increase modification speed. The following properties control batch updates:

```
glog.sql.update.batch.on = true
glog.sql.update.batch.maximum = 30
```

The **on** property specifies whether the application should use the batch updates. If set to true, inserts and updates will be batched whenever possible. The **maximum** property provides a system-wide constraint on the batch size. Note that batch updating adds overhead when database operations produce errors. Since the batch updating feature cannot provide information as to which record failed the batch, on failure the application retries the update one-by-one to find the errant record.

Integration

Suppressing Persistence XML

There is a capability to suppress storing the full transmission XML in the `i_transmission` table, and the full transaction XML in the `i_transaction` table for the RemoteQuery interface. Each of these should reduce the space used by the raw XML in the database and help performance by reducing the time to persist a transmission.

The full transmission XML is suppressed by setting the following property:

```
glog.integration.transmission.suppressPersistFullXml = true
```

> **Note**: The TransmissionHeader will be stored in the `i_transmission` table even when this property is enabled.

The Remote Query Transaction XML is suppressed by setting the following property:

```
glog.integration.transaction.suppressPersist.RemoteQuery = true
```

As of Oracle Transportation Management version 6.0, both of these properties are enabled by default.

Synchronization of Master Data

Repetitive updates of embedded elements like locations and items should be avoided since they produce unnecessary processing. Location and item masters should be synchronized independently of orders instead of embedding within the order. This practice will reduce inserts into I_TRANSACTION and I_LOG tables, thus increasing performance by reducing storage needs.

OutXMLProfile for Outbound Integration and Notification

Out XML Profiles allow you to generate outbound XML and exclude portions of outbound XML with a high degree of control. This is done by specifying what XML builder class files should be excluded when generating XML documents or by selecting an XML template on which you can base the outbound XML. Outbound XML includes integration and notification. Unnecessarily large outbound integration is a common cause of performance issues. Reducing the size of the XML will greatly reduce the load on the database and application server.

Inbound Integration Options

Oracle Advanced Queuing

> **Note**: Oracle Advanced Queueing (OAQ) is no longer supported and will not function.

Web Services

You can send data to Oracle Transportation Management via a web service call to the Oracle Transportation Management application server. The performance advantage of this approach is the elimination of the web tier from the process of persisting the XML. This should increase UI performance by reducing the overall load on the web server. Refer to the Integration Guide for more information on this subject.

User Interface

Finder Page Size

A Finder Page is the Oracle Transportation Management terminology for the standard result page which is returned when running an Oracle Transportation Management finder query. The number of records returned per page is configurable via the “Finder Page Size” user preference. Increasing this from the default value of 25 has a direct impact on the performance of loading this page due to the increased amount of data that needs to be processed and displayed. Values greater than 100 are likely to make the performance unacceptable.

User Favorites

Favorites allow you to limit initial `Find ()` results so that your favorite results are displayed first. For example, you can create a favorite for locations. In that favorite you specify the top 10 locations which you use on a regular basis. Then whenever you use the `Find ()` button the system first displays those 10 locations. If you want to find a different location you are given the option to search for more locations.

Adding Search Fields to Finders (Grid-Flattening)

Grid flattening enables administrators to present "pseudo fields" for search, result, and other manager pages that are configured to the needs of their users. Pseudo fields "flatten" a grid by presenting field labels that are more specific than the default fields. For example, a PO Number field could be added to the order base search page, so users can search for purchase orders directly by their numbers. Grid flattening is used to make it appear as if data from a child database table is on a parent database table. Using grid flattening on search and result pages should be done with caution since the resulting query is more complex and therefore susceptible to slower performance.

Active Table

Oracle Transportation Management implements data security via the Oracle Database feature Virtual Private Database (VPD). VPD provides an ability to dynamically append where conditions to SQL statements at runtime. Oracle Transportation Management uses this capability for its external predicate functionality and for cross-domain visibility. By default, a "`where`" condition is appended to a SQL statement for each table in the SQL. These additional where conditions can have an adverse effect on performance.

Depending upon a customer's particular setup in Oracle Transportation Management, these where conditions may be unnecessary. A capability has been implemented to control which tables have where conditions appended when running an Oracle Transportation Management finder query. The behavior for finders is controlled by the following property:

```
glog.query.vpdActiveTables.default=value
```

The possible values are:

- none = Do not apply VPD active table.
- primary = Apply VPD active table to the primary table of the query.
- comma-delimited tables = Apply VPD active table(s) for the supplied list.

Optionally, this behavior can be overridden for a specific Oracle Transportation Management query class with a property like:

```
glog.query.vpdActiveTables.QueryClassName=value
```

For example:

```
glog.query.vpdActiveTables.glog.server.query.invoice.InvoiceQuery=primary
```

The default behavior is to use primary table as active table for all finder set queries except following queries:

```
glog.server.query.preference.QueryTableQuery
glog.server.query.notify.StylesheetProfileQuery
```

Prior to version 6.0 the active table behavior was controlled by a property like:

```
glog.query.vpdOptimization.glog.server.query.invoice.InvoiceQuery=active
```

Status Queries

Queries based on status can be complex and often suffer from poor performance. Some of this complexity is due to the implementation of domains such that status types are created within each domain. Based on a customer's particular domain setup it may be possible to optimize these status queries.

The following glog.property can be used to control the domains from which status types will be queried:

```
glog.query.statusOptimization=value
```

The valid values are:

- none – No optimization so all domains are used.
- currentDomain – Status types from the current domain only are used.
- childDomains – Status types from the current and child domains are used.
- grantedDomains – Status types from all granted domains are used.

The following property controls how many domains will be included before the logic reverts to the same behavior as "none":

```
glog.query.statusOptimizationDomainMax=value
```

Manager Layout Producer Configuration

The XML document for an object is created by a series of "producers". These producers are associated with objects within Oracle Transportation Management. Each manager in Oracle Transportation Management has an XML document associated with it. When you create a user-defined manager layout you can create or remove fields from that manager. The system creates an XML document containing relevant data that is displayed in that manager. However the XML document may contain data that is no longer needed in a user-defined manager that has had a lot of fields removed. By deleting the associated producers from the user-defined manager the XML document will be smaller, allowing the page to load more quickly. Refer to the "Manager Layout: Producers" topic in online help for more information.

Parameterized Saved Queries

The UI finder mechanism in Oracle Transportation Management is a generic engine which enables flexible querying of any business object in Oracle Transportation Management. A SQL query is generated dynamically based on the user's input in the finder query page. Due to the potential complexity of the queries and the generic nature of the engine, it is not possible to always generate SQL which will perform optimally when executed. Saved queries provided a mechanism for an end-user to create a query using user-defined SQL. Users can create a saved query using SQL that is formatted in a manner which performs more efficiently. This also provides a mechanism to specify a hint in the SQL which will get passed to the Oracle SQL optimizer.

In addition to a standard saved query, there is an ability to create a saved query with parameters that can be fulfilled at runtime. For more information on this topic refer to "How To Enable the Use of Parameterized Saved Queries" in the online help.

Limiting Child Records in a Manager

The Oracle Transportation Management UI architecture is based on an XML document that is created by querying a record and all of its related child records. In some scenarios the number of child records can cause the XML document to be extremely large and therefore cause high memory usage and slow performance. The number of child records returned is configurable via the glog.property:

```
glog.query.limit.glog.server.query.orderbase.QueryName=limit
```

The QueryName is the name of the corresponding query and the limit indicates the maximum number of child records to return. If the number of child records exceeds this value, the manager will not display any child records of this type.

For example:

```
glog.query.limit.glog.server.query.order.ShipUnitLineQuery=100
```

As of version 6.0 many, managers have these properties set by default.

Functional Security CheckAccess Cache Sizes

Oracle Transportation Management functional security is cached by user. If your installation is going to have a large number of simultaneous users it may be necessary to increase the size of the functional security CheckAccess cache to be equal to the peak number of users expected in order to obtain optimal performance. In the glog.properties file on the Oracle Transportation Management application server, or the `CUSTOM` property set, you should configure the following property. The default value is 100.

For example:

```
glog.cache.CheckAccess.capacity=nnn
```

The following property controls the number of items caches per user. It can also be adjusted as necessary. For example:

```
glog.cache.CheckAccessItem.capacity=200
```

Client Session Management

You can open multiple pages simultaneously. By default, these client sessions are managed in a "least recently used" queue, but this can be configured to have a timeout. See the About Client Session Management topic in Online Help for more information on the available configuration options.

Scalability

Oracle Transportation Management scalability can be used to solve some performance problems. However, scalability should only be used after the single application server has been tuned properly. Adding more application servers to an installation that is performing poorly may actually make the performance worse. For instance, if the network or database is the performance bottle-neck, adding more application servers will only exaggerate the issue.

One reason to add another scalability node would be that a single application server is running out of memory or performing excessive garbage collection. Since the memory that can be allocated to a single JVM is limited, adding a scalability node will expand the total available memory. In this scenario, the additional scalability node may even be on the same physical hardware (assuming it has sufficient memory and processing power). If the processing power (CPU) is being maxed out on a single application server, adding another scalability node on another physical server will increase the total processing power.

See the Oracle Transportation Management Scalability Guide for more information.

Miscellaneous

Management of Report Files

During report generation, a number of files may be stored in the server’s file system. These include embedded images for HTML and the final reports. Oracle Transportation Management must have full create/read/write/delete privileges to these files and their directories to properly stage reports.

The following properties control the location of reporting files on each tier:

Table 13-9

Property	Holds	Default
glog.bipreports.report.path	final reports	temp

Final report files are managed with a Least Recently Used (LRU) algorithm. The default number of reports maintained by a running server is 200. Use the following property to modify this value:

```
glog.cache.Reports.capacity=<report capacity>
```

If a server generating reports fails or is cycled, existing report files will no longer be managed by the LRU algorithm. The following set of properties provides a backup to clean out old report and image files.

Table 13-10

Property	Holds	Default
glog.bipreports.cleanup.<type>.threshold	# of days to hold files of this type	Reports: 30 Images: 30
glog.bipreports.cleanup.<type>.frequency	# of minutes between file scans	Reports: 360 Images: 360
glog.bipreports.cleanup.<type>.files	comma-delimited list of file wildcards to remove	Reports: report-*, *.html, *.rtf, *.pdf, *.xls Images: xdo*

Valid types include: `report`, `webImages` and `appImages`.

14. Troubleshooting Known Issues

503 (Service Unavailable) Errors

While the application server is starting up, the web server will respond to all requests with a”503 (Service Unavailable)” error message. This will go away once the application server has fully started up. See **Starting Oracle Transportation Management Servers** for more information.

Browsers

Symptoms: Your browser displays stale or incorrect data while using Oracle Transportation Management.

Resolution: Configure your browser to "Check for newer versions of stored pages: Every visit to the page”. This will ensure that your browser displays the correct data and not cached pages. Depending on your browser, this option is usually under the "Cache” or "Temporary Internet files” section of the "Options” or "Preferences” window.

15. Appendix A: List of External Map Engine Properties

Table 15-1: List of External Map Engine Properties

Property	Description	Default
alk.map_url	Specify the map Java script URL provided by ALK. For use with the Workbench. Do not change this property.	
elocation.mapviewer_js_url	This property is used to specify the MapViewer Java script URL provided by Oracle Map Cloud Services. Do not change this property.	
elocation.elocation_js_url	This property is used to specify the eLocation Java script URL provided by Oracle Map Cloud Services. Do not change this property.	

All of these properties were added in 6.4.1.

16. Appendix B: General Properties

See the "Advanced Configuration: User-defined Properties" chapter for instructions on how to manage reserved properties.

Table 16-1: General Properties

Property	Reserved	Description
glog.advancedlayout.jspLocation	Yes	Location on the file system where the advanced layout JSP pages are stored.
glog.advancedlayout.schemaLocation	Yes	Location on the file system where the schema definitions are stored.
glog.advancedlayout.templatesLocation	Yes	Location on the file system where the templates for advanced layout are stored.
glog.advancedlayout.definitionsLocation	Yes	Location on the file system where the definitions for map metadata are stored.
glog.audit.asynchronousRemoval	Yes	If this property is set to False, when you delete records the audit trail records are deleted at the same time. If you set the property to True, then the audit trail records will be deleted by a new thread at a later time. Default: False
glog.audit.synchronizedWrite	Yes	This property prevents multiple users from overwriting audit trail records by enacting a lock on the database to enable only a single user at a time to make alterations. This property should be set to false if you are a heavy user of the audit trail feature. Default: True
glog.bean.retries.<use case> =<delay in msec>[,<# of retries (1)>]	Yes	This property controls retries. The use case is readFailure for load problems and writeFailure for store problems. When a failure occurs and you have not reached the maximum number of retries, it logs the error and that you are retrying the operation. If you hit the maximum number of retries, it logs the error and its full stack trace and throws the error to rollback the transaction. The default for both readFailure and writeFailure is one retry with a 100msec delay.

Property	**Reserved**	**Description**
glog.cpu.tick	Yes	Length of a CPU tick. i.e., milliseconds between CPU checks. If 0, no CPU checking is performed. Default: 100
glog.cpu.timeout	Yes	Number of milliseconds for CPU timeout. If the CPU poller thread is blocked for more than the timeout, a warning is sent to the console. Note that CPU timeout is tracked only if glog.cpu.useSystemTicks=true
glog.cpu.timerFast	Yes	Number of milliseconds the tick counter can be faster than the system clock before sending a warning to the console. Note that the tick counter is tracked only if glog.cpu.useSystemTicks=false. Default: 10000
glog.cpu.timerSlow	Yes	Number of milliseconds the tick counter can be slower than the system clock before sending a warning to the console. Note that the tick counter is tracked only if glog.cpu.useSystemTicks=false. Default: 10000
glog.cpu.useSystemTicks	Yes	Controls whether the system clock is used for CPU timeout messages. If false, a tick counter is tracked based on thread sleeps. If the tick counter falls behind the system clock, a warning is sent to the console. If true, the system clock is tracked at each tick. If the timeout is exceeded, a warning is sent to the console. Values are true or false.
glog.custscreens.jspLocation	Yes	This property indicates the location on the file system where the manager layout toolkit will write JSP files.
glog.database.dbaOnly.t2client.connectionURL	Yes	This property indicates the connection URL for database servers. Gets all the RAC database server details.

Property	Reserved	Description
glog.datasourceProfile.default.<data function>=<data source>	Yes	Prior to the addition of the Data Source Profile/Data Source tables and manager in OTM 4.5, data source management was specified by properties. The properties: glog.datasourceProfile.default.<data function>=<data source> map Data Functions to Data Sources, acting as a precursor to the DEFAULT DATA_SOURCE_PROFILE record. These assignments are still used by the application server, but only during startup. Certain data functions (e.g. startup, security, direct_admin, direct_privileged) are required to load startup data, including security, data source and connection pool information.
glog.dbupdate.allowDMLRefreshSuppression	Yes	If you are upgrading to 6.3 and have a direct SQL update DML statement with the Refresh Cache unchecked, the current migration rules will revert to a cache refresh for the DML. To avoid this and achieve complete backward compatibility set this property to true. This is potentially dangerous as entity and business caches will not be refreshed, but may be needed for upgrades.
glog.dbupdate.supportDMLReturning	Yes	If your Oracle Thin JDBC driver is incompatible with DML Returning, set this property to false. This forces any action with DML Returning to use Statement Parsing (with a warning), and sets the default refresh option to Statement Parsing for migrated actions.
glog.dbupdate.validateStatementType	Yes	If this property is set to true, Direct SQL Update checks the SQL for an update, insert or delete statement and validates this against the specified statement type. If the statement type is incorrect, it is adjusted and a warning is logged (under DirectSqlUpdate logging).

Property	Reserved	Description
glog.diag.collectors.secure	Yes	If this is true, the system ensures that URLs and other secure information are suppressed from all collectors accessible to users. The following configuration collectors are suppressed from the selection list when in a secure mode: APP TOOL VERSIONS DATABASE SETTINGS DATA CONNECTIONS ENTITY BEAN SETTINGS JVM SETTINGS OTM PROPERTIES OTM VERSION SCALABILITY WEB TOOL VERSIONS any user-defined Configuration Collector The following performance collectors are suppressed from the selection list when in a secure mode: ENTITY BEAN CACHES JVM DUMP WALLETS
glog.diag.trackedConnection.allOpenTraces	Yes	To write out the stack traces for the all open calls on each open connection, set this property to false.
glog.diag.trackedConnection.includeOpenCursors =<true|false>	Yes	Controls whether cursors are displayed in the OPEN CONNECTIONS performance diagnostic. If true, all open cursors for a connection are listed underneath the connection diagnostic. This setting requires glog.trackCursors.on=true. See the glog.trackCursors Properties help topic for details.

Property	Reserved	Description
glog.diag.trackedConnection.includeOpenCursorTraces =<true|false>	Yes	Controls whether cursors are displayed in the OPEN CONNECTIONS performance diagnostic. If true, the stack trace when the cursor is opened is included in the diagnostic. This setting requires glog.trackCursors.on=true and glog.trackCursors.trace=true. See the glog.trackCursors Properties help topic for details.
glog.diag.trackedConnection.includePinned= $glog.vpd.cache.usePinnedConnections$	Yes	Set this property to include pinned connections in the diagnostics.
glog.diag.trackedConnection.includeRegistered	Yes	Set this property to true to Include registered connections in the diagnostics. These are connections known to last longer than a single thread cycle. They are not part of any connection growth.
glog.diag.trackedConnection.openTrace	Yes	To write out the stack traces for the initial open call on each open connection, set this property to false.
glog.diag.trackedCursor.includeOpenTraces =<true|false>	Yes	Controls whether cursors are displayed in the OPEN CONNECTIONS performance diagnostic. If cursors are being tracked and not automatically closed on connection release, orphaned cursors are reporting in the OPEN CONNECTIONS diagnostic. If the property is set to true, this report includes the cursor's open traces.
glog.diagnostics.log.dir	Yes	This is used to specify the location of the diagnostic log directory.
glog.diagnostics.max.logfiles.per.domain	Yes	This defines the max number of log files that can be stored in the application server machine. Valid values are integers. Default: 40
glog.document.root	Yes	The root location on the server where images, .CSS, and .JS files are stored.

Property	Reserved	Description
glog.exception.sequence =[none|trace|all]	Yes	This property is used for sequence numbers on all displayed and logged exceptions. Sequencing can be controlled by this property. If set to "none", sequence numbers will not be collected or displayed with the exception. This is equivalent to previous behavior. The default value is "all". If set to "trace", sequence numbers will be collected and displayed in the logs and the standard Error screen. It will not be displayed with specialized error messages (e.g. the login page). If set to "all", sequence numbers will be collected and displayed in the logs, the standard Error screen and all specialized error messages. Note that an exception sequence number is only unique for a particular run of a server. If a server is bounced, its exception sequence number restarts at 1. In a Scalability environment, it's important to note the sequence number, time and message of any exception shown in the UI to properly match it with an exception in one of the server logs. See the OTM Application Scalability Guide for more details.
glog.explanation.on	Yes	If false, explanations are not supported and any overhead is removed. This property is false by default: explanations are not supported. In order to enable explanations, this property must be true, and the Tracking Mode and Tracking level on the Settings and Actions for the user must be set.
glog.fax.defaultHandler.routingMode	Yes	To See **ERROR! REFERENCE SOURCE NOT FOUND.** for details.

Property	**Reserved**	**Description**
glog.fax.defaultHandler.routingPicture	Yes	/name={lastName},{firstName}/fax={rawPhone}/ Replacements are represented by one of these identifiers surrounded by parenthesis; for example { firstName}. See the glog.workflow properties help topic for details on glog.workflow.notify.advisor.fax. See **ERROR! REFERENCE SOURCE NOT FOUND.** for details.
glog.fax.handler		glog.server.workflow.notify.DefaultFaxHandler See **ERROR! REFERENCE SOURCE NOT FOUND.** for details.
glog.httppost.allowPerSystemTimeouts	Yes	To enable external system overrides, set this to true. See **ERROR! REFERENCE SOURCE NOT FOUND.** for details.
glog.httppost.authorizationEncode	Yes	Determines whether or not passwords should be Base64 encoded when calling web services.
glog.instrumentation.instrumentor	Yes	This property is used to set the instrumentor. An instance of the default instrumentor (derived from glog.util.instrumentation.instrumentor) is added to each instrumentation class unless overridden by setting one of the following values on this property: • glog.util.instrumentation.ObjectCounts - provides current/max/total object instance counts. • glog.util.instrumentation.ObjectDistribution - provides ObjectCounts functionality along with a distribution of current objects over time. This allows identification of memory leaks. • glog.util.instrumentation.ObjectTrace - provides ObjectDistribution functionality along with a distribution of current objects over distinct stack traces.
glog.instrumentation.interval	Yes	Used to set a default interval (in milliseconds) when the instrumentor class is ObjectDistribution. (default = 1000)

Property	Reserved	Description
glog.instrumentation.profiler	Yes	This is used to set the profiler. An instance of the default profiler (derived from glog.util.instrumentation.Profiler) is added to each profiling method unless overridden by setting one of the following values on this property. • glog.util.instrumentation.ProfileTimes - provides top-level timing statistics for the method • glog.util.instrumentation.ProfileFrameTimes - provides timing statistics for each frame using the method. This allows both caller coverage statistics and child tree analysis.
glog.instrumentation.profilerOn	Yes	Used to enable (disable) profiling.
glog.integration.http.client.nonProxyHosts	Yes	Allow you to add all your internal server/hosts to the non Proxy Hosts list without affecting the standard glog.integration.http.nonProxyHosts list.
glog.integration.http.nonProxyHosts	Yes	Indicates the hosts which should be connected too directly and not through the proxy server. The value can be a list of hosts, each separated by a |, and in addition a wildcard character (*) can be used for matching. For example: "*.oracle.com|localhost".
glog.integration.http.proxyHost	Yes	Use this property to specify the address of the proxy server behind which Oracle Transportation Management runs.
glog.integration.http.proxyPort	Yes	Specifies the port of the proxy server behind which Oracle Transportation Management runs.
glog.integration.https.urlStreamHandlerClass	Yes	Specifies Sun's HTTPS URL Stream Handler's qualified classname in the current JDK that is used. If you do not define this property, the system defaults to the one used by Sun JDK 1.4 in WebLogic. Change to com.ibm.net.ssl.www.protocol.https.Handler for IBM JDK. Default: sun.net.www.protocol.https.Handler
glog.jdbc.logging.append	Yes	If true, log output from the JDBC driver should be appended to log files from previous runs. Default: True

Property	Reserved	Description
glog.jdbc.logging.count	Yes	The number of log files before cycling. Default: 5
glog.jdbc.logging.enabled	Yes	If JDBC logging is enabled, output is sent to the logging file and STDERR. Valid values: true or false
glog.jdbc.logging.file	Yes	Enter a valid log file path. If the logging file is left blank, output is sent to STDOUT. Otherwise, be sure to specify an absolute file path. E.g. $glog.log.dir$/jdbc.log
glog.jdbc.logging.level	Yes	The logging level controls the level of detail in JDBC logging. Use ALL for full diagnostics. Valid values are:ALL, FINER, FINE, INFO, WARNING, SEVERE, NONE. Default: ALL
glog.jdbc.logging.max	Yes	The maximum number of bytes per log file. Default: 1000000
glog.jdbc.logging.xml	Yes	If true, each log entry is in a canonical XML format; otherwise it is in plain text. Default: False
glog.jdbc.use10gClobs	Yes	This property adds support for 10g Clob updates using direct bind variables. To enable, set this to true. The default is false. Note that bound clobs are only used when OTM passes a String or StringReader to the underlying I/O modules. This includes all bean-based clobs like PROBLEM and FINDER_SET, as well as inbound integrations into I_TRANSMISSION and I_TRANSACTION. These tables should show a 40-50% performance improvement for their clob updates. Certain use cases like outbound integration, though, were already optimized around clob streaming. These are unaffected by the 10g support.
glog.jsp.tags.uom	Yes	Specifies the list of JSP tags that are used for unit of measure (UOM) fields.

Property	Reserved	Description
glog.local.cache.<Cache Name>.capacity=<number>	Yes	Each property controls the caching of data records loaded via Oracle Transportation Management data entities. This should only be modified or set with direction from Oracle Support.
glog.local.cache.mode	Yes	Controls the use of a cache for "local" entity beans. It can be set to "none", "some" or "all". The default is "some", using caches for only those beans with a high hit ratio. The list of which beans are cached under some mode are defined in glog.localEntity.properties. This list is determined by running in "all" mode and monitoring the cache diagnostic servlet for entity caches with a high hit ratio. See the Oracle Transportation Management Scalability Guide for more information.
glog.local.supportEntityLocking	Yes	Enables optional entity bean locking to avoid production synchronization issues due to local entity beans. When locking is on, the initial read or write of an entity bean will grab an object lock whose type is the bean, name is the primary key, and activity is the entity bean method. The object lock will be released when the encompassing transaction is rolled back or committed. Locking is off by default. Default: False
glog.log.max.backups =<# of backups>	Yes	Limits the ability for you to have log files grow in backup count. It sets the maximum number of backups for a log file, regardless of what a user sets for a specific log. The default values for this property is 0, meaning growth is only constrained by the user entries on the LOGFILE record.
glog.log.max.size =<# of chars>	Yes	Limits the ability for you to have log files grow in size. It sets the maximum size of log file, regardless of what a user sets for a specific log. The default values for this property is 0, meaning growth is only constrained by the user entries on the LOGFILE record.
glog.log.mimeType =<MIME type>	Yes	If you want to modify the MIME type of exported system logs, change this property. It defaults to text/plain.

Property	Reserved	Description
glog.log.viewerMaxKBtoDownload =<# of KB>	Yes	Lets you set the default maximum KB to save when saving to file in the System Log view. Default is 10000 (i.e. 10 MB). Maximum is 1000000 (1 GB).
glog.mail.images	Yes	This property is used when attaching a logo to each outbound mail message, either as an image link or an embedded image. It can take one of two forms: [http|https]://<host>/<mail image directory relative to the webserver document root> This will insert the mail logo as a link in the outbound email, where the image resides in a directory underneath the document root. Due to single sign-on restrictions, this option is currently available only to on-premise customers. If does not start with http: or https:, the logo is attached to the email as a standard MIME attachment. This allows any mail browser that is accepting and displaying attachments to read the logo. The attachment is assigned a Content-ID of "logo". This will insert the mail logo, found in the specified path, as an embedded image in the outbound email. Note that not all mail browsers support embedded images. For cloud customers, the property defaults to the location of uploaded mail images and cannot be changed.
glog.mail.mailDocRoot	Yes	Defines a directory to store mail files.
glog.mail.mailFileDir	Yes	If the size of content or an attachment exceeds glog.mail.maxContentSize, a link is embedded in the email (this is to avoid SMTP size limitations). This requires content and attachment files to be stored on an Oracle Transportation Manager server. This property gives the file directory holding these temporary mail files.
glog.mail.maxContentSize=<# of bytes>	Yes	Checks for any attachment larger than the given number of bytes. If found, the attachment is replaced with a link and the attachment is written to a temporary file. A temporary file LRU cache MailFiles is available that stores overflow attachments. The default capacity of this cache is 100.

Property	Reserved	Description
glog.mail.smtp.host	Yes	Default SMTP host for outbound emails.
glog.mail.smtp.port=<SMTP port #>		Set this to the port number to connect to secure SMTP servers. Works in conjunction with: glog.mail.smtp.useSSL=<true|false> glog.mail.smtp.useStartTLS=<true|false> Note that these are global properties and will impact any SMTP host defined in OTM.
glog.mail.smtp.useSSL=<true|false>		Set to connect to secure SMTP servers. Works in conjunction with: glog.mail.smtp.port=<SMTP port #> glog.mail.smtp.useStartTLS=<true|false> Note that these are global properties and will impact any SMTP host defined in OTM.
glog.mail.smtp.useStartTLS=<true|false>		Set to connect to secure SMTP servers. Works in conjunction with: glog.mail.smtp.port=<SMTP port #> glog.mail.smtp.useSSL=<true|false> Note that these are global properties and will impact any SMTP host defined in OTM.
glog.mediator.adhoc.poller.relaxSequencing	Yes	Set to True to suppress sequencing with the Poller. By suppressing the sequencing constraint, the poller can break a mediator deadlock (i.e. deadlock between two mediator groups). This suppression only impacts the poller. Normal mediator and UI activity continues with full sequencing.
glog.mediator.adhoc.poller.relaxSynchronization	Yes	To suppress synchronization with the poller, set to True. By suppressing the synchronization constraint, the poller can ignore an object lock deadlock or an orphaned process holding an object lock. This suppression only impacts the poller. Normal mediator and UI activity continues with full synchronization.

Property	Reserved	Description
glog.nonProxyHost.dbServer	Yes	A delimited list of the database servers that need to be excluded from using the proxy server.
glog.nonProxyHost.server	Yes	A delimited list of the servers that need to be excluded from using the proxy server, excluding the database servers.
glog.objectLock.cleanupViaProc	Yes	The Object Lock Cleanup recurring process optionally calls the pkg_object_lock_purge package to perform the cleanup - in batches with a timeout. To enable this functionality, set glog.objectLock.cleanupViaProc=true. It is false by default for backward compatibility.
glog.objectLock.delayForScalability	Yes	If you are distributing work across multiple app servers and processes are working on the same object, you may want to set this great than zero. Setting it greater than zero may increase processing time if one machine keeps getting object locks. Default: 0
glog.optimization.allowCallBacks	Yes	If set to false, then the solver cannot be invoked. This is done in order to resolve an issue around a FICO XPRESS OPTIMIZER 170 error along with AIX/WebSphere issue. Default: True
glog.optimization.flushDashLogs	Yes	When on (true), OTM will flush each of the FICO Xpress-Optimizer (Dash) logs immediately. Otherwise, OTM will collect all the logs for the entire solve and print it. Default: False
glog.optimization.invokeGC	Yes	Set this to true to force garbage collection. The lists that hold FICO XPRESS OPTIMIZER objects are cleared in the call to close(). Finally, every 100 calls to FICO XPRESS OPTIMIZER Solver, garbage collection is forced. Default: False

Property	Reserved	Description
glog.optimization.packingsolver.LoadConfigPackingMethod.skipAxleWeights	Yes	An exception can be thrown with a certain setting with axle weights. This property avoids the exception by disabling the axle weight calculation based on a property setting. This property needs to be added and set to true to skip axle weight calculation and thus avoid a potential exception in bulk plan.
glog.optimization.packingsolver.PackingParameters.ColGenBasePenalty	Yes	This is a penalty that is imposed for every order not packed into any container.
glog.optimization.packingsolver.PackingParameters.ColGenPriorityReward	Yes	Setting this property to a high number will influence the selection of higher priority orders over lower priority ones. Default: 100 Setting of 0 will give all of the orders the same priority inside the column generation logic.
glog.optimization.recordDash	Yes	When turned on (true), OTM will start recording processes for the steps within FICO Xpress-Optimizer (Dash). The output is stored in the logs folder and can be searched by file names containing "record" in it. This logging may have unpredictable results whenFICO Xpress-Optimizer (Dash) runs in parallel. For best results, you should turn off multithreading if you need these logs generated. Default: False
glog.optimization.saveDetailedLP	Yes	When turned on, OTM will save a more detailed LP file. The file is stored in the logs folder and has a .svf extension. Default: False
glog.process.allowSynchronousCleanUp	Yes	Specifies whether ProcessEvents must be cleaned up on the single process sweeper thread ("true") or on the threads associated with the individual events ("false"). Default: false.

Property	Reserved	Description
glog.process.defaultCleanUp	Yes	Processes within OTM represent a hierarchical tree of activity where parent processes kick off children processes. These children may run in the same thread as their parent or be queued up to run in some other thread. Generally, resources grabbed for a process are released as soon as a process is completed (i.e. when it and all its descendant processes have completed their activity). Note that process completion can occur long after the thread of execution for the root process has terminated its activity. The default timing of resource release is given by: glog.process.defaultCleanUp=<ByParent|OnCompletion> where OnCompletion releases resources as soon as a process is complete; ByParent relies on the parent process to release resources. The main use for ByParent clean up timing is when a parent process needs information collected by the child processes. This is true for: • transmission processing • formulation of regions • batch rating • batch order releasing Note: This property should not be modified without direction by OTM Support.

Property	**Reserved**	**Description**
glog.process.execAsEnabled.agent	Yes	This property will force planning actions to run as either the logged-in user or the domain of the business object when this property is set to False. • True: will use the domain of the logged-in user when planned through the user interface, but use the domain of the business object when planned through an agent. • False: will use the domain of the logged-in user. The glog.process.execAsEnabled property allows you to also configure how orders are planned, based on domains. For example, if USER.ADMIN is logged in to Oracle Transportation Management, and planning an order in domain A, Oracle Transportation Management will plan the shipment in the USER domain. If you were using the ExecAs property, Oracle Transportation Management would plan a shipment in the domain of the order, which is domain A.
glog.process.execAsEnabled.manualAgentTrigger	Yes	This property determines whether user-defined UI actions that run agents are executed in the context of the user's domain or the business object's domain or the current state of the thread. It accepts the following values: • Login: will always use the domain of the logged-in user, regardless of how the execAsEnabled.agent property is set. • True (or) Business: will always use the domain of the business object, regardless of how the execAsEnabled.agent property is set. Any other value: ignore this property and use the domain as defined by the execAsEnabled.agent property.
glog.process.execAsEnabled.notify	Yes	When the glog.process.execAsEnabled.notify=false, if an event occurs in a subdomain, and glog.lifetimeEventNotify.useDomainName is true, you get two emails in the subdomain, one from the subdomain and one from the parent domain.

Property	Reserved	Description
glog.process.execAsEnabled.planning	Yes	Depending on which of the following three options is defined here, this property will force planning actions to run as either the logged-in user or the domain of the business object. • Login: will always use the domain of the logged-in user, regardless of how the execAsEnabled.agent property is set. • True (or) Business: will always use the domain of the business object, regardless of how the execAsEnabled.agent property is set. False: uses the domain as defined by the execAsEnabled.agent property.
glog.process.statistics.outputFormat	Yes	Specifies the MIME subtype that should be specified for textual statistics reports, where "xml" and "plain" are the supported types. Unrecognized types are treated as "plain" but the specified value is still specified in the MIME subtype. Default: delimited.
glog.processcontrol.nextprocesstime.usecurrenttime	Yes	For recurring processes, the next process time could be set incorrectly if the server is down at the time of processing. This property determines how process time is established. If set to False: The next process time is set in multiples of the cycle time irrespective of the current time. If the next process time happens to be less than the current time, the next possible process time greater than the current time is computed and used. For example, bulk plans are scheduled to run on a staggered schedule every night. The server goes down Friday night and is restarted Sunday night. All bulk plans kicked off immediately Sunday night, but then were rescheduled to run as desired Monday night at their normal staggered times. If set to True: the next process time will not be set correctly if the server is down at the time of processing. Default: False
glog.properties.enabled	Yes	If set to false, all property loading is suppressed. JMX clients of OTM should set this to false.

Property	Reserved	Description
glog.properties.log.email.host	Yes	If you want to send email notification triggered by changes to properties, use this property to set the SMTP host for property change notification emails.
glog.properties.log.email.recipients	Yes	If you want to send email notification triggered by changes to properties, specify them in this property. This capability could be used to notify application administrators of all property changes. Note: This property does not display in the properties servlet by default. It must be added manually.
glog.properties.log.file	Yes	Set the full path for the properties change log using this property.
glog.query.statusOptimizationDomainMax	Yes	The property controls how many domains will be included before the logic reverts to the same behavior as "none".
glog.query.vpdActiveTables	Yes	This controls, which tables have where conditions appended when running an Oracle Transportation Management finder query.
glog.scalability.activatePortOffset	Yes	This provides an option for the application server to listen to a new socket when it is fully ready. The port for this socket is specified by this property. The value is the port offset from the RMI port for the application server. Default: 100 For example, a WebLogic application server listening to RMI on port 7001 will listen on port 7101 when Oracle Transportation Management is fully ready. A 3rd party can ping 7101 to determine if the server is ready for integrations. All scalability checks for application server availability include a check on the ready port.

Property	**Reserved**	**Description**

glog.server.workflow.mediator.supportGlobalGroup	Yes	Transaction mediation supports blocking across Integration Transmissions using a global process group designation. When set to true, a transmission, which has a ProcessGrouping in the TransmissionHeader, defined with a ProcessGroupOwner and the ProcessGroupOwner is not itself specified as a ProcessGrouping within the Transmission, then the ProcessGroupOwner (or Parent Group) will be setup as a global process group which will block on all transactions in the process group. For example, if the only ProcessGrouping in the transmission is defined as below:
		`<ProcessGrouping>`
		`<ProcessGroup>MYGROUP.48785</ProcessGroup>`
		`<ProcessGroupOwner>48785</ProcessGroupOwner>`
		`<InSequence>Y</InSequence>`
		`</ProcessGrouping>`
		then all transactions across transmissions that are designated in MYGROUP.48785 ProcessGroup will be sequenced globally.
		Set to true to enable the functionality.

glog.server.workflow.mediator.synchronization	Yes	The default for this property is "tight" to prevent any impact on 6.1 behavior and performance. Also you can set it to "twoPhase." The two-phased mediator synchronizes access to mediator data structures when an item is added to the mediator. This property requires that each server reboot.
glog.servlet.relax	Yes	Do not change without instruction from Oracle.

Property	Reserved	Description

glog.specialService.default.activityType	Yes	These properties are the default special services contained in Oracle Transportation Management. They include
		glog.specialService.default.activityType.TRANSPORT=TRANSPORT
		glog.specialService.default.activityType.BOBTAIL=BOBTAIL
		glog.specialService.default.activityType.DEADHEAD=DEADHEAD
		glog.specialService.default.activityType.LOAD=LOAD
		glog.specialService.default.activityType.PICKUPEMPTY=PICKUPEMPTY
		glog.specialService.default.activityType.PICKLOADED=PICKLOADED
		glog.specialService.default.activityType.UNLOAD=UNLOAD
		glog.specialService.default.activityType.DROPOFFEMPTY=DROPOFFEMPTY
		glog.specialService.default.activityType.DROPLOADED=DROPLOADED
		glog.specialService.default.activityType.WAIT=WAIT
		glog.specialService.default.activityType.REST=REST
		glog.specialService.default.activityType.RELAY=RELAY
		glog.specialService.default.activityType.BORDERCROSS=BORDERCROSS
		glog.specialService.default.activityType.DEPOT=DEPOT
		glog.specialService.default.activityType.HANDLING=HANDLING

Property	Reserved	Description
glog.sqlServlet.maxChars	Yes	If you need to export huge datasets, you should review whether 10 MB characters is enough. If not, they need to increase this property, while recognizing the potential for large memory consumption.
glog.translations.reloadNonexistentBundles	Yes	If set to true, bundle loading will reload non-existent bundles if sufficient time has passed since the last load request. Note that this mode suppresses the US fallback locale for foreign languages. This may impact existing customers if they are somehow requesting a language locale that does not exist in the TRANSLATION tables. If set to false, bundle loading will behave as it did before the patch. Default: true
glog.translations.reloadNonexistentBundlesFrequency	Yes	If glog.translations.reloadNonexistentBundles=true, this property controls how long to wait before the system tries to reload nonexistent bundles. Assume you login as a user that cannot load translations and he's the first to load his language. If the system tried to reload translations on every retrieval call for the bundle, the system would try to retrieve translations 1000's of times (one for each translation requested). Instead, set a timeout for the non-existent language cache. After the given # of milliseconds, the next request for the bundle will try to physically read it from the database again. As an example, Assume User U1 and U2 are both French. U1 has rights to load translations; U2 does not. After system restart, U2 logs in first and fails to load French translations. U2 sees labels instead of translations. If U1 tries to login within the frequency range, he will see labels as well. But once the cache is older than the frequency, U2 will load French on his next translation label retrieval and will see properly translated values. At that point, French is properly caches in both the ResourceBundle cache and the OTM cache. Both users should see French labels. Default: 30000

Property	Reserved	Description
glog.util.datasource.ConnectionPool.debug	Yes	Set this property to ON to debug connection pool errors.
glog.util.entityMapFilter.MIGRATION	Yes	Specifies a "filter" for Entity Names which should not be available for DBXML Export/Import and Migration Project Export/Import. For Migration Project there is an additional control which limits which types of Object can be selected as a Migration Object Group though this should always be a subset of the Migration Entity list. To filter an entity name from the Migration Entity list:- glog.util.entityMapFilter.MIGRATION=<regular expression> For example, glog.util.entityMapFilter.MIGRATION=Shipment.* will prevent any Entity name starting with 'Shipment' from appearing in the list.
glog.util.log.threadNameMaxLength	Yes	Logging allows the maximum thread length to be set by this property. Default: 200
glog.web.xml.diag	Yes	Set this property to TRUE to collect producer diagnostics data.
glog.webserver.context	Yes	The name of the web application. Should not be changed from GC3.
glog.webserver.contextADF	Yes	The name of the web application for ADF-based screens. Should not be changed from OTM-ADF.
glog.webserver.finder.body	Yes	XSL file to be used to render the criteria page for a finder.

Property	Reserved	Description
glog.webserver.finder.notMassUpdatableQueryName	Yes	Use it to append propertied defined query names to system defaults for Non-mass-updatable query names. System default non-mass-updatable query names are: glog.server.query.powerdata.GlUserQuery glog.server.query.gluser.GlUserQuery glog.server.query.userrole.UserRoleRestrictQuery glog.server.query.userrole.UserRoleQuery glog.server.query.security.AcrRoleQuery glog.server.query.gluser.GlLoginHistoryQuery glog.server.query.gluser.GlAccountPolicyQuery
glog.webserver.results.body	Yes	XSL file to be used to render the results of a finder search.
glog.webserver.results.buttons	Yes	XSL file to be used to render the frame with buttons in the results of a finder search.
glog.webserver.stylesheetRoot	Yes	This setting defines where on your system the XSL files are stored.
glog.webserver.login.fatal.<use case>=[true|false]	Yes	There are four data load use cases where the webtier loads or reloads data from the application tier: 1) webLog = WEB log information 2) uom = Units of Measure and Currency information 3) webFunction = Web SCA Functions (deprecated) 4) derivedDataQuery = Derived data query information This property controls whether an error in the load is fatal. If not fatal, login will be allowed. The faulty data load will be reattempted at each page hit, logging each failed load to the web log. The default for all use cases is false. I.e., data loads are not fatal.

Property	Reserved	Description
glog.webserver.URL glog.webserver.URL.RIQ	Yes	This setting should point to your Oracle Transportation Management web server. For example, tender emails from Oracle Transportation Management use this URL to link back to Oracle Transportation Management.
glog.webserver.waitForAppServer.delay glog.webserver.waitForAppServer.retries	Yes	Prior to property set loading, OTM allowed servlet activity once a Start message was received from the server. However, the server is not completely ready until it opens up the invokable port. These properties allow for a backoff delay to poll for the invokable port.
glog.workflow.notify.OrderJob	Yes	If set to "job", uses job domain instead of order domain for OrderJobTopic. Default: "order".
glog.workflow.notify.OrderShipment	Yes	If set to "shipment", uses shipment domain instead of order domain for OrderShipmentTopic. Default: "order".