

# **Oracle® Communications Services Gatekeeper**

Licensing

Release 4.1

January 2009

**ORACLE®**

Copyright © 2007, 2008, 2009 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Transaction Based Licensing

The usage cost model for Oracle Communications Services Gatekeeper is based on the idea of the *transaction unit*. This book describes how transaction *categories* are monitored and how transaction *units* are calculated.

## Transaction Based Usage Costs

Usage costs for Oracle Communications Services Gatekeeper are based on the rate (measured in *transaction units per second* or TUPS) during a specific time period per 24-hour interval. As Oracle Communications Services Gatekeeper runs, it continuously counts the number of transactions it processes. Every 5 minutes that count is stored. After 24 hours of use, the busiest 60 minute window (the *busy hour*) is identified and the total number of transaction units processed in that period is divided by 3600. This computed busy hour forms the basis of the cost of using Oracle Communications Services Gatekeeper.

**Note:** Transaction units are based on statistics collected by the statistics service. By default, the system simply keeps track of usage for auditing purposes, but if an operator wishes to monitor usage more closely, it is possible to use OAM to set thresholds to raise alarms if certain usage levels are reached. For more information on setting alarms, see [“Managing and Configuring Transaction Licenses”](#) in *System Administrator’s Guide*, another document in this set.

What specifically constitutes a transaction unit is based on the type of functionality being measured. Sending an SMS is not the same as setting up a call between two parties. In general, the Oracle Communications Services Gatekeeper model is arranged in a tiered manner, using two large transaction categories, Base Platform and Oracle Module. Usage monitoring is based on the

TUPS rates associated with these two transaction categories. For more information, see [Transactions - Categories and Units](#).

**Note:** In application-initiated scenarios, transaction units are logged only if Oracle Communications Services Gatekeeper returns an OK to the application. If there is an exception, no transaction unit is recorded. In network-triggered scenarios, transaction units are logged only if Oracle Communications Services Gatekeeper returns an OK to the network. If there is an exception, no transaction unit is recorded.

## Transactions - Categories and Units

Oracle Communications Services Gatekeeper keeps track of transaction units at two basic levels:

- [Base Platform TUPS](#), the larger and more general level, which tracks:
  - [Platform Services](#), which covers services such as subscriber profile or callable policy that can be used by multiple capabilities
  - [Custom Modules](#), which covers custom communication services

And also includes:

- Oracle Module TUPS, which covers individual communication services.
- [Oracle Module TUPS](#), which tracks transaction units in Oracle Communications Services Gatekeeper supplied communication services, which fall into the following groups:
  - Call Control
  - Messaging
  - Mobility
  - Presence
  - Payment

## Base Platform TUPS

The larger and more general rate category is Base Platform TUPS. The Base Platform TUPS rate is the sum of the TUPS limits for Platform Services and Custom Modules plus that for Oracle Modules.

## Platform Services

Platform Services covers general services that can be used by multiple capabilities.

These services include:

- Subscriber Profile (based on Extended Web Services Subscriber Profile Interface)
- Callable Policy

## Application-initiated - Subscriber Profile

A Subscriber Profile transaction unit is recorded in the following situations:

- Getting one or more attributes from a subscriber profile.
- Getting either a whole or a subset of a profile identified by a profile ID.

## Application-initiated - Callable Policy

A Callable Policy transaction unit is recorded in the following situation:

- Evaluating a request using the Interceptor stack.

These services support application-initiated requests only.

## Network-triggered

Network-triggered scenarios are not applicable for subscriber profile or callable policy.

# Custom Modules

Custom Modules covers communication services or custom modules created using Oracle Communications Services Gatekeeper's Platform Development Toolkit.

In general there are two main types of custom module transactions:

- [Application-initiated](#)
- [Network-triggered](#)

## Application-initiated

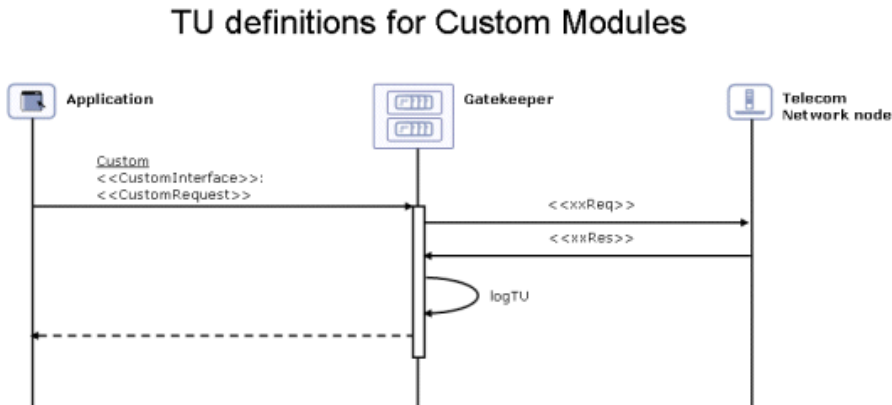
A transaction unit for a custom module application-initiated scenario:

Oracle Communications Services Gatekeeper processes a request that is submitted by the application and passes it to the underlying network node and then Oracle Communications Services Gatekeeper processes the response that is submitted by the underlying network node and passes it to the application.

**Note:** The exact sequence of request and response is defined during the custom module development process.

Figure 1 illustrates when a generic custom module application-initiated transaction unit is logged. The illustrated operations are generic and abstract and would depend on which application-facing interface and/or network protocol the communication service is using.

**Figure 1 TU definition Custom Module application-initiated scenario**



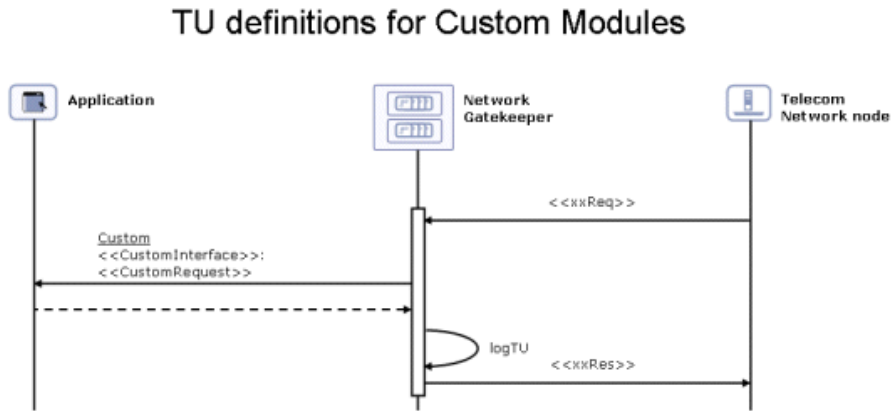
## Network-triggered

A transaction unit for a custom module network-triggered scenario:

Oracle Communications Services Gatekeeper processes a request that is submitted by the telecom network node and passes it on to the application, which processes it and returns a response to Oracle Communications Services Gatekeeper, which passes it on to the telecom network node.

Figure 2 illustrates when a generic custom module network-triggered transaction unit is logged. The illustrated operations are generic and abstract and would depend on which application-facing interface and/or network protocol the communication service is using.

Figure 2 TU definition Custom Module network-triggered scenario



## Oracle Module TUPS

The second rate category is Oracle Module TUPS. The Oracle Module TUPS rate measures the total transaction rate for communication services delivered as a part of Oracle Communications Services Gatekeeper.

### Note:

For definitional purposes, Oracle Module-based communication services are divided into the following transaction groups:

- [Messaging](#)
- [Mobility](#)
- [Call Control](#)
- [Presence](#)
- [Payment](#)

## Messaging

The Messaging transaction group covers transactions for the following Parlay X 2.1 and RESTful communication services:

- Short Messaging
- Multimedia Messaging

It also covers transactions for all Extended Web Services and RESTful WAP Push communication services, the Binary SMS communication service, and the native MM7 communication service.

There are two main types of transaction units in the Messaging group:

- [Application-initiated](#)
- [Network-triggered](#)

### Application-initiated

In an application-initiated scenario, a transaction unit consists of the following sequence:

Oracle Communications Services Gatekeeper receives a request from an application and passes it on to the underlying network node. If notifications have been set up, Oracle Communications Services Gatekeeper also receives the delivery notification from the network and passes it back to the application. Even if the message is distributed to a group of destination addresses, only one transaction unit is logged per message.

**Note:** If the application is not registered for delivery notifications, the transaction unit is not affected.

### Network-triggered

In a network-triggered scenario, a transaction unit consists of one of the following two sequences:

- Oracle Communications Services Gatekeeper receives a message from the network node and passes it on to the application.
- Oracle Communications Services Gatekeeper receives a message from the network node and consumes the message itself. In this case, Oracle Communications Services Gatekeeper itself acts as the application.



## Mobility

The Mobility transaction group covers transactions for the following Parlay X 2.1 and RESTful communication services:

- Terminal Location

There are two main types of transaction units in the Mobility group:

- [Application-initiated](#)
- [Network-triggered](#)

### Application-initiated

In an application-initiated scenario, a transaction unit consists of one of the two following sequences:

- Oracle Communications Services Gatekeeper receives a polling request for location from an application and passes it to the underlying network node. Oracle Communications Services Gatekeeper then receives the immediate response that the underlying network node returns and passes it back to the application. Group requests are treated the same as single requests.
- Network receives a notification setup request from an application that *also* requests an immediate location check.

**Note:** An immediate check followed by a status notification is considered two transaction units.

A transaction unit is *not* recorded in the following situation:

- Oracle Communications Services Gatekeeper receives a notification setup request that does *not* include an immediate status or location check

### Network-triggered

In a network-triggered scenario, a transaction unit consists of the following sequence:

Oracle Communications Services Gatekeeper receives a notification request that is submitted by the network node and then passes it on to the application.

**Note:** For the purposes of logging transaction units, all notification triggers are equivalent: periodic, geographical, or state-change based are logged in the same manner.

## Call Control

The Call Control transaction group includes transactions for the following Parlay X 2.1/3.0 and RESTful communication services:

- Third Party Call
- Call Notification
- Audio Call

There are two main types of transaction units in the Call Control group:

- [Application-initiated](#)
- [Network-triggered](#)

### Application-initiated

In an application-initiated scenario, a transaction unit consists of one of the following:

- An application requests that Oracle Communications Services Gatekeeper setup a third party call. For the purposes of logging transaction units, no distinction is made based on who eventually terminates the call: the application, the caller or the callee.
- An application requests that Oracle Communications Services Gatekeeper add or transfer a participant in a call (PX 3.0 only).
- An application requests that Oracle Communications Services Gatekeeper either play an Audio Call message of any type or play a message and collect digits (PX 3.0 only).

### Network-triggered

In a network-triggered scenario, a transaction unit consists of one of the two following sequences:

- Oracle Communications Services Gatekeeper receives a notification request from the network node and passes it on to the application. The request contains status information about the call, such as the callee is busy or not answering. No distinction is made between a notification that arrives in notify mode (for information only) and one that arrives in interrupt mode (where the call can be manipulated in the return message from the application).
- Oracle Communications Services Gatekeeper receives a notification request from the network node and processes the call itself, based on rules that have been provisioned by the application at an earlier time.

## Presence

The Presence transaction group covers transactions for the following Parlay X 2.1 and RESTful communication services:

- Presence

There are three main types of transaction units in the Presence group:

- [Application-initiated - Watcher](#)
- [Network-triggered - Watcher](#)
- [Application-initiated - Presentity](#)

### Application-initiated - Watcher

In an application-initiated scenario, a transaction unit consists of the following:

Oracle Communications Services Gatekeeper receives a polling request for user presence information from an application and passes it to the underlying network node. Oracle Communications Services Gatekeeper then receives the immediate response that the underlying network node returns and passes it back to the application.

### Network-triggered - Watcher

In a network-triggered scenario, a transaction unit consists of the following:

Oracle Communications Services Gatekeeper receives a status change notification request from the network node and passes it on to the application.

### Application-initiated - Presentity

In an application-initiated scenario, a transaction unit consists of the following:

A presentity publishes its presence information to Oracle Communications Services Gatekeeper, which passes it on to the Presence Server.

## Payment

The Payment transaction group covers transactions for the following Parlay X 3.0 and RESTful communication services:

- Payment

The Payment transaction group includes only application-initiated requests.

## **Application-initiated**

A Payment transaction unit is recorded in the following situations:

- Charging or refunding an amount
- Charging a split amount
- Reserving an initial amount and releasing it
- Reserving an initial amount and charging it
- Reserving an additional amount

A payment transaction unit is *not* recorded in the following situation:

- Releasing a previously held amount