

# Oracle® Communication Services Gatekeeper

Licensing

Release 4.0

June 2008

ORACLE®

---

Oracle Communication Services Gatekeeper Licensing, Release 4.0

Copyright © 2007, 2008, 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 licensing model for WebLogic Network 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 Licensing

Licensing for WebLogic Network Gatekeeper is based on a maximum allowed rate (measured in *transaction units per second* or TUPS) during a specific time period per 24-hour interval. As Network 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.

Every day this computed busy hour TUPS value is compared to the rates specified by the license agreement as represented in the license key file. If the calculated TUPS rates are higher than the rates allowed by the license key, alarms are raised and logs are written.

**Note:** By contract, anyone operating Network Gatekeeper in a production environment must monitor the transaction rate and report any breaches in the licensing agreement to BEA. As well, anyone operating Network Gatekeeper in a production environment must agree to produce historical license logs upon request from BEA, backdating for a time period as defined in the license agreement.

What specifically constitutes a transaction unit is based on the type of functionality being measured. Sending a Short Message is not the same as setting up a call between two parties. In general, Network Gatekeeper licensing is set up in a tiered manner, using two large transaction

categories, Base Platform and BEA Module. License monitoring and enforcement is based on the TUPS rates associated with these two transaction categories. For more information, see [Transactions - Categories and Units](#).

**Note:** TUPS rates measures only usage inside Network Gatekeeper. As a result transaction units can be logged even in the case when end-to-end requests fail, if that failure is due to errors in elements outside Network Gatekeeper. What this means is:

- Transaction units are logged if an application-initiated request fails to be delivered to a network node when that delivery failure is due to a problem in the network node or the communication channel towards the network node
- Transaction units are logged if a network-triggered request fails to be delivered to an application when that delivery failure is due to a problem in the application or the communication channel towards the application.

## Transactions - Categories and Units

WebLogic Network 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:

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

## 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, Custom Modules, and BEA Module TUPS.

## 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

These services support application-initiated requests only.

### Application-initiated - Callable Policy

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

- Evaluating a request using the Interceptor stack.

### Application-initiated - Subscriber Profile

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

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

### Network-triggered

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

## Custom Modules

Custom Modules covers communication services or custom modules created using Network Gatekeeper's Extension Toolkit.

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

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

### Application-initiated

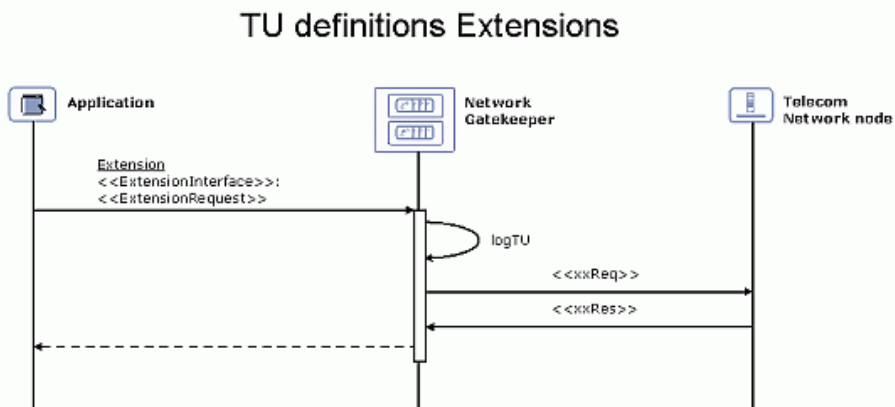
A transaction unit for a custom module application-initiated scenario:

Network Gatekeeper processes a request that is submitted by the application and passes it to the underlying network node and Network 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 extension module development project.

Figure 1 illustrates when a generic extension 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 Extension Module application-initiated scenario**



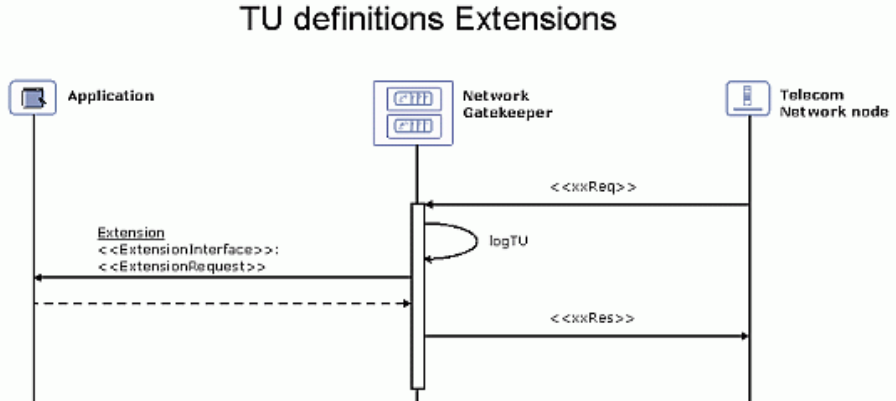
## Network-triggered

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

Network 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 Network 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 Extension Module network-triggered scenario**



## BEA Module TUPS

The second rate category is BEA Module TUPS. The BEA Module TUPS rate is the allowed total transaction rate for communication services delivered as a part of WebLogic Network Gatekeeper. The BEA Module TUPS rate is the absolute limit for non-custom communication service transactions; that is, Base Platform capacity beyond the defined BEA Module capacity cannot be used for any BEA delivered communication service.

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

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

## Messaging

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

- Short Messaging
- Multimedia Messaging

It also covers transactions for all Extended Web Services WAP Push communication services and the Binary SMS 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:

Network Gatekeeper receives a request from an application and passes it on to the underlying network node. If notifications have been set up, Network 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:



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

## Mobility

The Mobility transaction group covers transactions for the following Parlay X 2.1 communication service:

- 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:

- Network Gatekeeper receives a polling request for location from an application and passes it to the underlying network node. Network 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 Gatekeeper 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:

- Network 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:

Network 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 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 Network 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 Network Gatekeeper add or transfer a participant in a call (PX 3.0 only).
- An application requests that Network Gatekeeper either play an Audio Call message of any type or (PX 3.0 only) play a message and collect digits.

### Network-triggered

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

- Network 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).

- Network 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 communication service:

- Presence

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

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

### Application-initiated

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

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

### Network-triggered

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

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

## Transaction Based Licensing