

Oracle® Communications Services Gatekeeper

Communication Service Reference

Release 4.1

September 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.

Contents

1. Document Roadmap

Document Scope and Audience	1-1
Guide to this Document	1-2
Terminology	1-3
Related Documentation	1-6

2. Parlay X 3.0 Audio Call Communication Service

An Overview of the Parlay X 3.0 Audio Call Communication Service	2-1
How It Works	2-2
Supported Network Protocols	2-2
Parlay 3.3 Multi-Party Call Control and Call User Interaction	2-2
Configuration Specifics for the Parlay X 3.0 Audio Call Communication Service	2-3
Charging Data Records	2-3
Event Data Records	2-4
Statistics	2-4
Supported Address Schemes	2-5

3. Parlay X 2.1 Third Party Call Communication Services

An Overview of the Parlay X 2.1 Third Party Call Communication Services	3-1
How It Works	3-2
Call Setup	3-2
Call Duration	3-2
Supported Network Protocols	3-3

SIP	3-3
INAP/SS7	3-3
Configuration Specifics for the Parlay X 2.1 Third Party Call Communication Services . .	3-3
Charging Data Records	3-4
Generation of CDRs	3-4
Event Data Records	3-4
Statistics	3-5
Supported Address Schemes	3-5

4. Parlay X 3.0 Third Party Call Communication Service

An Overview of the Parlay X 3.0 Third Party Call Communication Service.	4-1
How It Works.	4-2
Supported Network Protocols	4-3
Parlay 3.3 Multi-Party Call Control	4-3
Configuration Specifics for the Parlay X 3. 0 Third Party Call Communication Service . .	4-3
Charging Data Records	4-4
Generation of CDRs	4-4
Event Data Records	4-4
Statistics	4-5
Supported Address Schemes	4-5

5. Parlay X 2.1 Call Notification Communication Service

An Overview of the Parlay X 2.1 Call Notification Communication Service	5-1
How It Works.	5-2
Simple monitoring	5-2
Monitoring and rerouting	5-3
Supported Network Protocols	5-3

SIP	5-4
Configuration Specifics for the Parlay X 2.1 Call Notification Communication Services	5-4
Charging Data Records	5-4
Generation of CDRs	5-4
Event Data Records	5-5
Statistics	5-5
Supported Address Schemes	5-6

6. Parlay X 3.0 Call Notification Communication Service

An Overview of the Parlay X 3.0 Call Notification Communication Service	6-1
How It Works	6-2
Monitoring	6-2
Monitoring and rerouting	6-3
Supported Network Protocols	6-4
Parlay 3.3 Multi-Party Call Control	6-4
Configuration Specifics for the Parlay X 3.0 Call Notification Communication Service	6-4
Charging Data Records	6-4
Generation of CDRs	6-4
Event Data Records	6-5
Statistics	6-5
Supported Address Schemes	6-6

7. Parlay X 2.1 Short Messaging Communication Service

An Overview of the Parlay X 2.1 Short Messaging Communication Service	7-1
Processing Application-initiated Requests	7-2
Send Receipts	7-3
Delivery Receipts	7-3

Processing Network-triggered Requests	7-4
Multiple Connections and Multiple Plug-in Instances	7-5
Supported Network Protocols	7-6
SMPP v3.4	7-6
Short Code Translation	7-6
Configuration Specifics for the Parlay X 2.1 Short Messaging Communication Service . .	7-7
Charging Data Records	7-7
Generation of CDRs	7-7
Event Data	7-8
Statistics	7-8
Supported Address Schemes	7-9

8. Extended Web Services Binary SMS Communication Service

An Overview of the EWS Binary SMS Communication Service	8-1
Send Receipts	8-2
Delivery Receipts	8-2
Supported Network Protocols	8-2
Configuration Specifics for the EWS Binary SMS Communication Service	8-3
Charging Data Records	8-3
Event Data	8-4
Statistics	8-4
Supported Address Schemes	8-5

9. Parlay X 2.1 Multimedia Messaging Communication Service

An Overview of the Parlay X 2.1 Multimedia Messaging Communication Service . . .	9-1
Processing Application-initiated Requests	9-3
Send Receipts	9-3
Delivery Receipts	9-4

Processing Network-triggered Requests	9-5
Supported Network Protocols	9-6
MM7	9-6
Short Code Translation	9-6
Configuration Specifics for the Parlay X 2.1 Multimedia Messaging Communication	
Services	9-7
Charging Data Records	9-7
Generation of CDRs	9-7
Event Data Records	9-7
Statistics	9-8
Supported Address Schemes	9-8

10. Parlay X 3.0 Payment Communication Services

An Overview of the Parlay X 3.0 Payment Communication Service	10-1
Processing Direct Queries/Application-initiated Requests	10-2
Processing Notifications/Network-triggered Requests	10-2
Supported Network Protocols	10-2
Diameter	10-2
Configuration Specifics for the Parlay X 3.0 Payment Communication Services	10-3
Charging Data Records	10-3
Generation of CDRs	10-3
Event Data Records	10-4
Statistics	10-4
Supported Address Schemes	10-5

11. Parlay X 2.1 Terminal Location Communication Services

An Overview of the Parlay X 2.1 Terminal Location Communication Service	11-1
Processing Direct Queries/Application-initiated Requests	11-2
Processing Notifications/Network-triggered Requests	11-3

Supported Network Protocols	11-3
MLP 3.0 and 3.2.	11-3
Configuration Specifics for the Parlay X 2.1 Terminal Location Communication Services	
11-3	
Charging Data Records	11-4
Generation of CDRs	11-4
Event Data Records	11-4
Statistics	11-5
Supported Address Schemes	11-5

12. Parlay X 2.1 Presence Communication Service

An Overview of the Parlay X 2.1 Presence Communication Service	12-1
The Client as Presence Consumer	12-2
The Client as Presence Supplier	12-3
Supported Network Protocols	12-3
Configuration Specifics for the Parlay X 2.1 Presence Communication Service.	12-3
Charging Data Records	12-4
Generation of CDRs	12-4
CDR Details	12-4
Event Data Records	12-4
Statistics	12-5
Supported Address Schemes	12-6

13. EWS Subscriber Profile Communication Service

An Overview of the Extended Web Services Subscriber Profile Communication Service .	
13-1	
How It Works.	13-2
Supported Network Protocols	13-2
LDAPv3	13-2

Configuration Specifics for the Extended Web Services Subscriber Profile	
Communication Service	13-2
Charging Data Records	13-3
Event Data Records	13-3
Statistics	13-3
Supported Address Schemes	13-4

14. Extended Web Services WAP Push Communication Service

An Overview of the Extended Web Services WAP Push Communication Service . . .	14-1
Supported Network Protocols	14-2
Push Access Protocol (PAP) 2.0	14-2
Configuration Specifics for the WAP Push Communication Service	14-3
Charging Data Records	14-4
Generation of CDRs	14-4
Event Data Records	14-4
Statistics	14-4
Supported Address Schemes	14-5

15. Native MM7 Communication Service

An Overview of the Native MM7 Communication Service	15-1
Status Reports	15-2
Delivery Reports	15-2
Read-Reply Report	15-2
Network-triggered Multimedia Messages	15-3
Supported Network Protocols	15-3
MM7	15-3
Configuration Specifics for the Native MM7 Communication Service	15-3
Charging Data Records	15-4
Generation of CDRs	15-4

Event Data Records	15-4
Statistics	15-5
Supported Address Schemes	15-5

16. Native SMPP Communication Service

An Overview of the Native SMPP Communication Service	16-1
Session Handling and Provisioning	16-2
Application-triggered Short Messages	16-2
Network-triggered Short Messages	16-3
Supported Network Protocols	16-3
SMPP v3.4	16-3
Short Code Translation	16-3
Limitations and Recommendations	16-3
Load Balancing, High Availability and Fail-Over.	16-4
Configuration Specifics for the Native SMPP Communication Service	16-5
Charging Data Records	16-5
Generation of CDRs	16-5
Event Data Records	16-5
Statistics	16-7
Supported Address Schemes	16-8

A. Events, Alarms, and Charging

Events	17-1
Event handling in the Access Tier	17-1
Event handling in the Network Tier	17-2
Alarms	17-4
Management integration	17-5
OSS	17-5

SNMP	17-5
Charging Data Records	17-5

Document Roadmap

This chapter describes the audience for and the organization of this document. It includes:

- [Document Scope and Audience](#)
- [Guide to this Document](#)
- [Terminology](#)
- [Related Documentation](#)

Document Scope and Audience

This document provides a detailed reference for information that is specific to individual communication services. It includes:

- An overview of each communication service's functioning
- The network protocols each application facing interface supports
- Configuration specifics, including communication service specific:
 - Charging Data Records
 - Event Data Records

This document will be of use to system administrators charged with installing and maintaining Oracle Communications Services Gatekeeper, as well as managers, support engineers, and sales and marketing people. For an overview of the characteristics that all communication services

have in common, please see the [“Introducing Communication Services”](#) chapter in *Concepts and Architectural Overview*, another document in this set.

Guide to this Document

This document contains the following chapters:

[Chapter 1, “Document Roadmap.”](#) This chapter

[Chapter 2, “Parlay X 3.0 Audio Call Communication Service.”](#) Information on the communication service that supports the Parlay X 3.0 Audio Call interface

[Chapter 3, “Parlay X 2.1 Third Party Call Communication Services.”](#) Information on the communication services that support the Parlay X 2.1 Third Party Call interface

[Chapter 4, “Parlay X 3.0 Third Party Call Communication Service.”](#) Information on the communication service that supports the Parlay X 3.0 Third Party Call interface

[Chapter 5, “Parlay X 2.1 Call Notification Communication Service.”](#) Information on the communication service that supports the Parlay X 2.1 Call Notification interface

[Chapter 6, “Parlay X 3.0 Call Notification Communication Service.”](#) Information on the communication service that supports the Parlay X 3.0 Call Notification interface

[Chapter 7, “Parlay X 2.1 Short Messaging Communication Service.”](#) Information on the communication service that support the Parlay X 2.1 Short Messaging interface

[Chapter 8, “Extended Web Services Binary SMS Communication Service.”](#) Information on the communication service that supports Extended Web Service Binary SMS interface

[Chapter 9, “Parlay X 2.1 Multimedia Messaging Communication Service.”](#) Information on the communication services that support the Parlay X 2.1 Multimedia Messaging interface

[Chapter 10, “Parlay X 3.0 Payment Communication Services.”](#) Information on the communication services that support the Parlay X 3.0 Payment interface

[Chapter 11, “Parlay X 2.1 Terminal Location Communication Services.”](#)Information on the communication services that support the Parlay X 2.1 Terminal Location interface

[Chapter 12, “Parlay X 2.1 Presence Communication Service.”](#) Information on the communication service that supports the Parlay X 2.1 Presence interface

[Chapter 13, “EWS Subscriber Profile Communication Service.”](#) Information on the communication service that supports the Extended Web Services Subscriber Profile interface.

[Chapter 14, “Extended Web Services WAP Push Communication Service.”](#) Information on the communication service that supports the Extended Web Services WAP Push Message interface

[Chapter 15, “Native MM7 Communication Service.”](#) Information on the communication service that supports the Native MM7 interface.

[Chapter 16, “Native SMPP Communication Service.”](#) Information on the communication service that supports the Native SMPP interface.

Terminology

The following terms and acronyms may be used in this document:

- **Account**—A registered application or service provider. An account belongs to an account group, which is tied to a common SLA
- **Account group**—Multiple registered service providers or services which share a common SLA
- **Administrative User**—Someone who has privileges on the Oracle Communications Services Gatekeeper management tool. This person has an administrative user name and password
- **Alarm**—The result of an unexpected event in the system, often requiring corrective action
- **API**—Application Programming Interface
- **Application**—A TCP/IP based, telecom-enabled program accessed from either a telephony terminal or a computer
- **Application-facing Interface**—The Application Services Provider facing interface
- **Application Service Provider**—An organization offering application services to end users through a telephony network
- **AS**—Application Server
- **Application Instance**—An Application Service Provider from the perspective of internal Oracle Communications Services Gatekeeper administration. An Application Instance has a user name and password
- **CBC**—Content Based Charging
- **End User**—The ultimate consumer of the services that an application provides. An end user can be the same as the network subscriber, as in the case of a prepaid service or they

can be a non-subscriber, as in the case of an automated mail-ordering application where the subscriber is the mail-order company and the end user is a customer to this company

- Enterprise Operator —See Service Provider
- Event—A trackable, expected occurrence in the system, of interest to the operator
- Communication Service—A mechanism by which a particular telecom network capability is made available to Internet based applications. It consists of an application-facing interface (north), a generic capability, and a network-facing interface (south).
- HA —High Availability
- HTML—Hypertext Markup Language
- IP—Internet Protocol
- JDBC—Java Database Connectivity, the Java API for database access
- Location Uncertainty Shape—A geometric shape surrounding a base point specified in terms of latitude and longitude. It is used in terminal location
- MAP—Mobile Application Part
- Mated Pair—Two physically distributed installations of Oracle Communications Services Gatekeeper nodes sharing a subset of data allowing for high availability between the nodes
- MM7—A multimedia messaging protocol specified by 3GPP
- MPP—Mobile Positioning Protocol
- Network Plug-in—The Oracle Communications Services Gatekeeper module that implements the interface to a network node or OSA/Parlay SCS through a specific protocol
- NS—Network Simulator
- OAM —Operation, Administration, and Maintenance
- Operator—The party that manages the Oracle Communications Services Gatekeeper. Usually the network operator
- OSA—Open Service Access
- PAP—Push Access Protocol
- Plug-in—See Network Plug-in

- Plug-in Manager—The Oracle Communications Services Gatekeeper module charged with routing an application-initiated request to the appropriate network plug-in
- Policy Engine—The Oracle Communications Services Gatekeeper module charged with evaluating whether a particular request is acceptable under the rules
- Quotas—Access rule based on an aggregated number of invocations. See also Rates
- Rates—Access rule based on allowable invocations per time period. See also Quotas
- Rules—The customizable set of criteria - based on SLAs and operator-desired additions - according to which requests are evaluated
- SCF—Service Capability Function or Service Control Function, in the OSA/Parlay sense.
- SCS—Service Capability Server, in the OSA/Parlay sense. Oracle Communications Services Gatekeeper can interact with these on its network-facing interface
- Service Capability—Support for a specific kind of traffic within Oracle Communications Services Gatekeeper. Defined in terms of communication services
- Service Provider—See Application Service Provider
- SIP—Session Initiation Protocol
- SLA—Service Level Agreement
- SMPP—Short Message Peer-to-Peer Protocol
- SMS—Short Message Service
- SMSC—Short Message Service Centre
- SNMP—Simple Network Management Protocol
- SOAP—Simple Object Access Protocol
- SPA—Service Provider APIs
- SS7—Signalling System 7
- Subscriber—A person or organization that signs up for access to an application. The subscriber is charged for the application service usage. See End User
- SQL—Structured Query Language
- TCP—Transmission Control Protocol

- USSD—Unstructured Supplementary Service Data
- VAS—Value Added Service
- VLAN—Virtual Local Area Network
- VPN—Virtual Private Network
- Oracle Communications Services Gatekeeper Core—The container that holds the Core Utilities
- Oracle Communications Services Gatekeeper Core Utilities—A set of utilities common to all communication services
- WSDL —Web Services Definition Language
- XML—Extended Markup Language

Related Documentation

This communication service reference is a part of the Oracle Communications Services Gatekeeper documentation set. The other documents include:

- [*System Administrator's Guide*](#)
- [*Concepts and Architectural Overview*](#)
- [*Installation Guide*](#)
- [*Integration Guidelines for Partner Relationship Management*](#)
- [*Managing Accounts and SLAs*](#)
- [*Statement of Compliance and Protocol Mapping*](#)
- [*Application Development Guide*](#)
- [*SDK User Guide*](#)
- [*Handling Alarms*](#)
- [*Licensing*](#)
- [*Platform Development Studio - Developer's Guide*](#)
- [*Platform Test Environment*](#)

- *[RESTful Application Development](#)*

Document Roadmap

Parlay X 3.0 Audio Call Communication Service

This chapter describes the Parlay X 3.0 Audio Call communication service in detail:

- [An Overview of the Parlay X 3.0 Audio Call Communication Service](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 3.0 Audio Call Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 3.0 Audio Call Communication Service

The Audio Call communication service implements the Parlay X 3.0 Audio Call set of interfaces. For the exact version of the standard the communication service supports, see “[Appendix A: Standards and Specifications](#)” of *Concepts and Architectural Overview*, another document in this set.

Using the Audio Call communication service, an application can:

- Play audio to one or more call participants in an existing call session set up by the Parlay X 3.0 Third Party Call communication service.
- Find out if the audio is currently being played or has not yet started to play.
- Explicitly end playing of the audio.
- Collect digits from call participants in response to an audio message that has been played to them and, in conjunction with the Parlay X 3.0 Call Notification communication service, return the information to the application.
- Interrupt an ongoing interaction, such as on-hold music.

How It Works

The Audio Call communication service can be used by applications to play audio messages to one or more call participants in an existing call. The existing call is identified by the Call Session Identifier returned to the application at the time the call session is set up using the Parlay X 3.0 Third Party Call communication service. If desired, applications can receive digits collected from those participants in response to the audio message using a notification set up using the Parlay X 3.0 Call Notification communication service.

The audio message content to be played must be defined in a binary format such as WAV stored at a URL available to the network and rendered by an audio player. Oracle Communications Oracle Communications Services Gatekeeper does *not* actually render the message: this is the responsibility of equipment that must be present on the target telecom network, such as Interactive Voice Response (IVR) systems.

Supported Network Protocols

Off the shelf, the Parlay X 3.0 Audio Call communication service supports a combination of the following network protocols:

- [Parlay 3.3 Multi-Party Call Control and Call User Interaction](#)

Parlay 3.3 Multi-Party Call Control and Call User Interaction

When Oracle Communications Services Gatekeeper is configured to use these protocols, it connects to a Parlay 3.3 Multi-Party Call Control SCS and a Parlay 3.3 Call User Interaction SCS provided by an OSA/Parlay Gateway. See “[Appendix A: Standards and Specifications](#)” of *Concepts and Architectural Overview*, another document in this set, for the exact version of the standards Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Parlay X 3.0 Audio Call Communication Service

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Audio Call communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

There are two Audio Call-Parlay 3.3 MPCC/CUI-specific CDRs. They occur when the following criteria are met:

- When `sendInfoRes` is sent from the network to Oracle Communications Services Gatekeeper, indicating that the audio message has completed playing, if this is not the result of an explicit request to stop from the application.
- When `sendInfoAndCollectRes` is sent from the network to Oracle Communications Services Gatekeeper, indicating that the audio message has completed playing and the call participant’s response has been collected in the form of digits.

Event Data Records

[Table 2-1](#) lists the EDR IDs created by the Audio Call-Parlay 3.3 MPCC/CUI communication service. This does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 2-1 Event types emitted in Audio Call communication service

EdrId	Method Called
11100	playAudioMessage
11101	getMessageStatus
11102	endMessage
11103	startPlayAndCollectInteraction
11104	stopMediaInteraction
11105	sendInfoRes
11106	SendInfoErr
11107	sendInfoAndCollectRes
11108	SendInfoAndCollectErr
11109	attachMediaRes
111010	attachMediaErr
111011	detachMediaRes
111012	detachMediaErr
111013	abortActionRes
111014	abortActionErr

Statistics

The table below outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications

Services Gatekeeper for the Parlay X 3.0 Audio Call-Parlay 3.3 MPCC/CUI communication service:

Table 2-2 Transaction types for Parlay X 3.0 Audio Call-Parlay MPCC/CUI communication service

Method	Transaction type
startPlayAndCollectInteractions	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED
playAudioMessage	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED

Supported Address Schemes

The Parlay X 3.0 Audio Call-Parlay 3.3 MPCC/CUI communication service does not use any address schemes directly, but in working with the Parlay X 3.0 Third Party Call and Call Notification communication services, it uses the `tel:` scheme indirectly.

Parlay X 2.1 Third Party Call Communication Services

This chapter describes the Parlay X 2.1 Third Party Call communication services in detail:

- [An Overview of the Parlay X 2.1 Third Party Call Communication Services](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 2.1 Third Party Call Communication Services](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 2.1 Third Party Call Communication Services

The SOAP Service Facade Third Party Call Parlay X 2.1 communication services implement the Parlay X 2.1 Third Party Call interface. For the exact version of the standard these communication services support, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Third Party Call interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using a Third Party Call Parlay X 2.1 communication service, an application can:

- Set up a call between two parties. For example, an application could set up a call between an investor and a broker if a particular stock reaches a predetermined price. Or a computer user could set up a call between himself and someone in the address book with a mouse click.
- Query Oracle Communications Services Gatekeeper for the status of a previously set up call
- Cancel a call it is creating as it is about to be set up
- Terminate an ongoing call it created

How It Works

In the Parlay X 2.1 Third Party Call communication services model, a call has two distinct stages:

- [Call Setup](#)
- [Call Duration](#)

Call Setup

There are two parties involved in Third Party Call calls: the A-party (the caller) and the B-party (the callee). When a call is set up using a Third Party Call communication service, Oracle Communications Services Gatekeeper attempts to set up a call leg to the A-party. When the caller goes off-hook (“answers”), Oracle Communications Services Gatekeeper attempts to set up a call leg to the B-party. When the callee goes off-hook, the two call legs are connected using the underlying telecom network. This ends the call setup-phase.

The application can cancel the call during this phase.

Call Duration

While the call is underway, the audio channel that connects the caller and the callee is completely managed by the underlying telecom network. During this phase of the call, the application can only query as to the status of the call. A call can be terminated in two ways, either using the application-facing interface, or having the caller or callee hang up.

Requests using a Parlay X 2.1 Third Party Call communication service flow only in one direction, from the application to the network. Therefore this communication service supports only application-initiated (or mobile-terminated) functionality.

Note: Third Party Call communication services manage only the signalling, or controlling, aspect of a call. The media, or audio, channel is managed by the underlying telecom network. Only parties residing on the same network can be controlled, unless:

- The network plug-in connects to a media gateway controller
- One of the participants is connected to a signalling gateway so that, from a signalling point of view, all parties reside on the same network

Supported Network Protocols

Off the shelf, Parlay X 2.1 Third Party Call communication services can be configured to support the following network protocols:

- [SIP](#)
- [INAP/SS7](#)

SIP

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to the SIP network. In this case, Oracle Communications Services Gatekeeper acts as a Back-to-Back User Agent. During the call duration phase, the actual call is peer-to-peer.

INAP/SS7

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to the underlying network via an INAP/SS7 interface. See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Parlay X 2.1 Third Party Call Communication Services

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Parlay X 2.1 Third Party Call communication services, including:

- [Charging Data Records](#)
- [Event Data Records](#)

- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

The Parlay X 2.1 Third Party Call using SIP and the Parlay X 2.1 TPC using INAP/SS7 are standard communication services. CDRs are written:

- When Oracle Communications Services Gatekeeper has received an event from the network stating that the second call leg has been connected and the associated phone has started to ring. This CDR is *not* dependent on whether the call is answered.
- When Call Information has been successfully delivered to the application
- When the call is ended by the application
- When the call request is canceled by the application

In addition, Parlay X 21 Third Party Call using INAP/SS7 also writes CDRs:

- When the network notifies Oracle Communications Services Gatekeeper that the call is connected; that is, that the second participant has answered the call
- When the network notifies Oracle Communications Services Gatekeeper that a call participant has disconnected

Event Data Records

Both the Parlay X 2.1 Third Party Call SIP and the TPC INAP/SS7 communication services use the 3.0 mechanism and are documented in [Table 3-1](#) and [Table 3-2](#). This does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 3-1 Event types emitted in the Parlay X 2.1 Third Party Call/SIP communication service

Event data	Description
8022	makeCall
8023	getCallInformation

Table 3-1 Event types emitted in the Parlay X 2.1 Third Party Call/SIP communication service

Event data	Description
8024	endCall
8025	cancelCallRequest

The following events are generated for the INAP/SS7 communication service in addition to those listed in [Table 3-1](#) above:

Table 3-2 Event types emitted in the Parlay X 2.1 Third Party Call/INAP-SS7 communication service

Event data	Description
8026	callConnected
8027	callReleasedNotification

Statistics

[Table 3-3](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for either the Parlay X 2.1 Third Party Call/SIP or INAP/SS7 communication services:

Table 3-3 Transaction types for Parlay X 2.1 Third Party Call communication service

Method	Transaction type
makeCall	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED

Supported Address Schemes

The Parlay X 2.1 Third Party Call using either SIP or INAP/SS7 communication services supports the `tel:` address scheme.

Parlay X 2.1 Third Party Call Communication Services

Parlay X 3.0 Third Party Call Communication Service

This chapter describes the Parlay X 3.0 Third Party Call communication service in detail:

- [An Overview of the Parlay X 3.0 Third Party Call Communication Service](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 3.0 Third Party Call Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 3.0 Third Party Call Communication Service

This communication service implements the Parlay X 3.0 Third Party Call interface. For the exact version of the standard this communication service supports, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Using the Third Party Call Parlay X 3.0 communication service, an application can:

- Setup a uniquely identified call between one or more participants and add or delete further participants, or transfer them to other established calls

- Indicate charging information to be associated with the call session
- Indicate information on any media to be used in association with the call
- Interact with the functionality of other communication services, such as Audio Call to play audio to call participants or Call Notification to respond to previously established notifications
- Query Oracle Communications Services Gatekeeper for the status of an established call or particular call participants
- Terminate an ongoing call it created

How It Works

The Parlay X 3.0 Third Party Call communication service can be used by applications that need to set up calls to one or more participants, as, for example, in establishing a conference call. It can also be used to set up calls that also use the capabilities of other communication services (Audio Call or Call Notification). The application first sets up the call session by using the `makeCallSession` operation, passing in the address of at least one (the A-party) participant.

Note: In the most common case, the address of a second participant, the B-party, is also passed in.

Oracle Communications Services Gatekeeper sends a request to establish the first call leg to the underlying network and returns a unique identifier (`callSessionIdentifier`) to the application synchronously. This identifier allows the application to perform further administrative tasks on the call, and provides any other communication services (Audio Call or Call Notification) access to the call at any point during the call session.

Note: The Call Session Identifier is returned to the application before the A-party goes off hook (answers). To receive information on the ongoing status of the call session, the application polls Oracle Communications Services Gatekeeper, using the identifier and the `getCallSessionInformation` operation.

While the call is underway, the application can add additional parties, delete one or more parties, or transfer parties to and from other established call sessions, using the identifier returned during the call setup phase. The functionality of the Audio Call (playing media to one or more participants) and Call Notification (for example rerouting a “busy” address to a second predefined one) communication services can also be accessed in the context of the call session using this same identifier. A call can be terminated in two ways, either by using the application-facing interface, or by having the call participants hang up.

Requests using the Third Party Call communication service flow only in one direction, from the application to the network. By itself this communication service supports only application-initiated (or mobile terminated) functionality. Mobile originated scenarios can be supported when using this communication service in concert with Call Notification.

Note: The Parlay X 3.0 Third Party Call communication service manages only the signalling, or controlling, aspect of a call. The call itself takes place in the underlying telecom network. Only parties residing on the same network can be controlled, unless:

- The network plug-in connects to a media gateway controller
- One of the participants is connected to a signalling gateway so that, from a signalling point of view, all parties reside on the same network

Supported Network Protocols

Off-the shelf, the Parlay X 3.0 Third Party Call communication service can be configured to support the following network protocol:

- [Parlay 3.3 Multi-Party Call Control](#)

Parlay 3.3 Multi-Party Call Control

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to a Parlay 3.3 Multi-Party Call Control SCS provided by an OSA/Parlay Gateway. See [“Appendix A: Standards and Specifications”](#) in the *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Parlay X 3.0 Third Party Call Communication Service

Communication services share many common features, covered in the [“Introducing Communication Services”](#) chapter of the *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Parlay X 3.0 Third Party Call communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)

- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

There are seven specific Parlay X 3.0 Third Party Call-Parlay 3.3 MPCC communication service CDRs. They occur when the following criteria are met:

- After Oracle Communications Services Gatekeeper has created the first call leg of a call session. This is not dependent on whether the participant has answered.
- After a call participant has been added to a call session, deleted from a session, or transferred to or from another session.
- When call information or call participant information has been successfully delivered to the application
- When the call is ended by the application

Event Data Records

The EDRs produced by the Parlay X 3.0 Third Party Call/Parlay 3.3 communication service are of the standard type and are documented in [Table 4-1](#). This does not include EDRs created when exceptions are thrown. For more information on the contents of standard EDRs, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 4-1 Event types emitted in the Parlay X 3.0 Third Party Call/Parlay 3.3 communication service

EdrId	Method Called
10000	addCallParticipant
10001	deleteCallParticipant
10002	endCallSession
10003	getCallParticipantInformation
10004	getCallSessionInformation
10005	makeCallSession
10006	transferCallParticipant

Table 4-1 Event types emitted in the Parlay X 3.0 Third Party Call/Parlay 3.3 communication service

EdrId	Method Called
10007	eventReportRes
10008	eventReportErr
10009	callLegEnded
10010	callEnded
10011	createAndRouteCallLegErr
10012	getInfoRes
10013	getInfoErr

Statistics

[Table 4-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 3.0 Third Party Call-Parlay 3.3 MPCC communication service:

Table 4-2 Transaction types for Parlay X 3.0 Third Party Call - Parlay 3.3 MPCC communication service

Method	Transaction type
makeCallSession	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED
transferCallParticipant	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED
addCallParticipant	TRANSACTION_TYPE_CALL_CONTROL_SERVICE_INITIATED

Supported Address Schemes

The Parlay X 3.0 Third Party Call-Parlay 3.3 MPCC communication service supports the `tel:` address scheme.

Parlay X 2.1 Call Notification Communication Service

The following chapter describes the Parlay X 2.1 Call Notification communication service in detail:

- [An Overview of the Parlay X 2.1 Call Notification Communication Service](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 2.1 Call Notification Communication Services](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 2.1 Call Notification Communication Service

The SOAP Services Facade Call Notification communication service implements the Parlay X 2.1 Call Notification set of interfaces. For the exact version of the standard this communication service supports, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Call Notification interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using the Call Notification communication service, an application can:

- Set up and tear down notifications on call events for a given combination of caller and callee
- Receive additional notifications on call events related to the notification in question
- Affect a call during call setup

Note: The operations made available by the Call Notification communication service are concerned only with monitoring (and, in some cases, making certain changes to) calls. This communication service is not used to set up new calls; it is used only to reroute or terminate calls already in progress.

How It Works

For an application to receive notifications about call setup attempts from the network, it must register its interest in these notifications by setting up a subscription in Oracle Communications Oracle Communications Services Gatekeeper. A subscription, or a notification, is defined by a set of addresses and a set of criteria. The criteria define the events in which the application is interested.

Note: The addresses may possibly be translated by some mechanism in the telecom network prior to reaching Oracle Communications Services Gatekeeper.

Two types of notifications exist:

- [Simple monitoring](#)
- [Monitoring and rerouting](#)

Simple monitoring

An application can register to be notified about the following events as the call between the caller and the callee is set up:

- Callee is busy
- Callee is not reachable
- Callee does not answer

- Call is in progress
- Call setup in progress

Monitoring and rerouting

In addition to monitoring the state of call setup, an application can also choose to make certain changes to the call under certain conditions. An application can:

- Intercept a call setup attempt between the caller and the callee and reroute the call (to a C-party) without making an attempt to connect with the callee (B-party). An example might be a general technical support number that is routed to the appropriate call center based on time of day.

In addition, if one of the monitored events occurs (busy, not reachable, does not answer), an application can:

- Let further processing of the call be handled by the network
- End the call
- Reroute the call to another callee (C-party.)

Requests (that is, registration for notifications) using the Call Notification communication service flow only in one direction: from the application to Oracle Communications Services Gatekeeper. Therefore this communication service supports only network-triggered requests.

Note: The Call Notification communication service manages only the signalling, or controlling, aspect of a call. The media, or audio, channel is managed by the underlying telecom network. Only parties residing on the same network can be controlled, unless:

- The network plug-in connects to a media gateway controller
- One of the participants is connected to a signalling gateway so that, from a signalling point of view, all parties reside on the same network

Supported Network Protocols

Off-the shelf, the Parlay X 2.1 Call Notification communication service can be configured to support the following network protocol:

- [SIP](#)

SIP

When Oracle Communications Services Gatekeeper is configured to use this protocol it connects to the SIP network. Oracle Communications Services Gatekeeper acts as a SIP Back-to-Back User Agent. Although from a signalling point of view three parties are involved - the caller, the callee, and Oracle Communications Services Gatekeeper - from the SIP perspective, the media channel (the actual call) that is established is purely peer to peer.

Configuration Specifics for the Parlay X 2.1 Call Notification Communication Services

Communication services share many common features, covered in “[Introducing Communication Services](#)” in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Parlay X 2.1 Call Notification communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

The CDRs generated by the Parlay X 2.1 Call Notification to SIP communication service belong to two basic groups. They occur when the following criteria are met:

- After a `notifyBusy`, `notifyCalledNumber`, `notifyNoAnswer`, `notifyAnswer`, or `notifyNotReachable` is sent from the network.
- After a `handleBusy`, `handleCalledNumber`, `handleNoAnswer`, or `handleNotReachable` is called

Event Data Records

[Table 5-1](#) lists EDR IDs created by the Parlay X 2.1 Call Notification to SIP communication service. This list does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 5-1 Event types emitted in the Call Notification/SIP communication service

Event data	Description
3000	startCallNotification
3001	stopCallNotification
3002	startCallDirectionNotification
3003	stopCallDirectionNotification
8014	notifyBusy
8015	notifyCalledNumber
8016	notifyNoAnswer
8017	notifyNotReachable
8018	handleBusy
8019	handleCalledNumber
8020	handleNoAnswer
8021	handleNotReachable

Statistics

[Table 5-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2.1 Call Notification to SIP communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 5-2 Transaction types for Parlay X 2.1 Call Notification-SIP communication service

Method	Transaction type
notifyBusy	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
notifyNotReachable	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
notifyNoAnswer	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
notifyCalledNumber	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
handleBusy	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
handleNotReachable	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
handleNoAnswer	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED
handleCalledNumber	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I INITIATED

Supported Address Schemes

The Parlay X 2.1 Call Notification-SIP communication service supports the `tel:` address scheme.

Parlay X 3.0 Call Notification Communication Service

This chapter describes the Parlay X 3.0 Call Notification communication service in detail:

- [An Overview of the Parlay X 3.0 Call Notification Communication Service](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 3.0 Call Notification Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 3.0 Call Notification Communication Service

This Call Notification communication service implements the Parlay X 3.0 Call Notification set of interfaces. For the exact version of the standard this communication service supports, see [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview*, another document in this set.

Using a Call Notification communication service, an application can:

- Set up and tear down notifications on call events for a given combination of caller and callee
- Receive notifications on call events related to established notifications
- Interact with the functionality of other communication services, including Audio Call to play audio to call participants and/or collect data from them or Third Party Call to reroute the call or set up additional call legs.
- End the call

Note: The operations made available by the Call Notification communication service are concerned only with monitoring (and, in some cases, making certain changes to) calls during the setup phase. By itself, this communication service is not used to set up new calls, only to reroute or terminate calls already in progress.

How It Works

In order for an application to receive notifications about call setup attempts from the network, it must register its interest in these notifications by setting up a subscription in Oracle Communications Services Gatekeeper. A subscription, or a notification, is defined by a set of addresses and a set of criteria. The criteria define the events in which the application is interested.

Note: The addresses may possibly be translated by some mechanism in the telecom network prior to reaching Oracle Communications Services Gatekeeper.

Two types of notifications exist:

- [Monitoring](#)
- [Monitoring and rerouting](#)

Monitoring

An application can register to be notified about the following events as the call between the caller and the callee is set up:

- The callee is busy
- The callee is not reachable
- The callee does not answer
- The caller is attempting to call the callee
- The callee has answered the call

Note: The above notifications may include a Call Session Identifier identifying the call session in the network, if available, to allow interactions with other Parlay X Web Services, such as Third Party Call and Audio Call. These interactions tend to be asynchronous.

- A call participant has interacted with a play-and-collect-media event. The notification contains the results of the interaction, including the digits collected.
- A call participant has interacted with a play-and-record-media event. The notification contains the results of the interaction, including the location of the recorded information.

Note: Setting up a notification for a play and record media event is supported in Oracle Communications Services Gatekeeper version 4.0, but setting up the play and record interaction is *not* supported in the Parlay X 3.0 Audio Call communication service in this version.

Monitoring and rerouting

In addition to monitoring the state of call setup, an application can also choose to make certain changes to the call under certain conditions, in a synchronous manner. In the case of certain monitored events (busy, not reachable, no answer, call attempt), an application can specify how to handle them, including:

- Continue to let the call be managed by the network in the normal manner, by, for example, playing a busy tone
- End the call
- Intercept the call setup attempt between the caller and the callee and reroute the call to another callee (C-party) without making an attempt to connect with the callee (B-party). An example might be a general technical support number that is routed to the appropriate call center based on time of day.

Note: If the call is rerouted, the media type is always negotiated by the underlying network. The MediaInfo parameter is not currently used by the communication service.

Because the Call Notification communication service does handle traffic in two directions (from the application to the network and from the network to the application) its functionality has some aspects of both the application-initiated and the network-triggered types. The communication service itself manages only the signalling, or controlling, aspect of the call. The call itself, the media, or audio, channel, is completely handled by the underlying telecom network.

Note: Because the communication service manages only the signalling aspect of the call, only parties residing on the same network can be controlled, unless:

- The network plug-in connects to a media gateway controller

- One of the participants is connected to a signalling gateway so that, from a signalling point of view, all parties reside on the same network

Supported Network Protocols

Off-the shelf, the Parlay X 3.0 Call Notification communication service can be configured to support the following network protocol:

- [Parlay 3.3 Multi-Party Call Control](#)

Parlay 3.3 Multi-Party Call Control

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to a Parlay 3.3 Multi-Party Call Control SCS provided by an OSA/Parlay Gateway. See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Parlay X 3.0 Call Notification Communication Service

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Call Notification communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

There are three CDRs generated by the Call Notification Parlay X 3.0/Parlay 3.3 MPCC communication service. They occur when the following criteria are met:

- After a `reportNotification` is sent from the Parlay gateway to Oracle Communications Services Gatekeeper, indicating that a call event defined by the notification has occurred and (in appropriate cases) needs to be handled.
- After a `sendInfoandCollectRes` has been sent from the Parlay gateway to Oracle Communications Services Gatekeeper, indicating that a call participant has interacted with a play-and-collect operation. The response includes the digits collected.

Event Data Records

[Table 6-1](#) lists EDR IDs created by the Parlay X 3.0 Call Notification/Parlay 3.3 MPCC communication service. This does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 6-1 Event types emitted in the Parlay X 3.0 Call Notification/Parlay 3.3 MPCC communication service

Event data	Description
11000	<code>startCallDirectionNotification</code>
11001	<code>stopCallDirectionNotification</code>
11002	<code>startCallNotification</code>
11003	<code>stopCallNotification</code>
11004	<code>startPlayAndCollectNotification</code>
11006	<code>stopMediaInteractionNotification</code>
11007	<code>reportNotification</code>
11008	<code>deleteNotification</code>
11009	<code>createNotification</code>
11011	<code>sendInfoAndCollectRes</code>

Statistics

[Table 6-2](#) below outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications

Services Gatekeeper for the Parlay X 3.0 Call Notification/Parlay 3.3 MPCC communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 6-2 Transaction types for Parlay X 3.0 Call Notification/Parlay 3.3 MPCC communication service

Method	Transaction type
reportNotification (both CallNotification and CallDirection)	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I NITIATED
sendInfoAndCollectRes (callNotification only)	TRANSACTION_TYPE_CALL_CONTROL_NETWORK_I NITIATED

Supported Address Schemes

The Parlay X 3.0 Call Notification/Parlay 3.3 MPCC communication service supports the `tel:` address scheme.

Parlay X 2.1 Short Messaging Communication Service

This chapter describes the Parlay X 2.1 Short Messaging communication service in detail:

- [An Overview of the Parlay X 2.1 Short Messaging Communication Service](#)
 - [Processing Application-initiated Requests](#)
 - [Processing Network-triggered Requests](#)
 - [Supported Network Protocols](#)
 - [Short Code Translation](#)
- [Configuration Specifics for the Parlay X 2.1 Short Messaging Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 2.1 Short Messaging Communication Service

The SOAP Service Facade Short Messaging communication service implements the Parlay X 2.1 Short Messaging set of interfaces. For the exact version of the standard this communication service supports, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Short Messaging interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using a Short Messaging communication service, an application can:

- Send short messages to one or many destination addresses. The payload in these short messages can be text, logos, or ringtones

Note: Logos must be in either SmartMessaging or EMS format. The image is not scaled. Ringtones must be in either SmartMessaging or EMS (iMelody) format.

- Ask to be notified that delivery receipts for sent short messages have been received from the network
- Receive delivery receipts on sent short messages that have arrived from the network
- Explicitly query Oracle Communications Services Gatekeeper for delivery receipts on sent short messages
- Sign up to be notified if specified short messages for the application have been received from the network
- Receive notifications that specified short messages for the application have arrived from the network. These notifications include the short message payload
- Explicitly poll Oracle Communications Services Gatekeeper for short messages sent to the application that have arrived from the network and been stored in Oracle Communications Services Gatekeeper

Requests can flow in two directions using the Parlay X 2.1 Short Messaging communication service: from the application to the network (called application-initiated or mobile terminated) and from the network to the application (called network-triggered or mobile originated). Both of these scenarios are covered below.

Processing Application-initiated Requests

After an application has sent a short message to one or more destination addresses, two different types of responses can be returned:

- [Send Receipts](#)
- [Delivery Receipts](#)

Send Receipts

Send receipts are merely acknowledgements that the network node has received the short message from the application by means of Oracle Communications Services Gatekeeper. Although a single short message may be sent to multiple destination addresses, normally only one send receipt is returned to the application by Oracle Communications Services Gatekeeper. The receipt is returned synchronously in the response message to the `sendSms` operation.

Delivery Receipts

Delivery receipts contain the delivery status of the short message; that is, whether the short message has actually been delivered by the network to the mobile terminal. There is one delivery receipt per destination address, with one of three possible states:

- **Successful.** In the case of concatenated short messages, this is returned only when all the parts have been successfully delivered. (see the Note below)
- **Unsuccessful.** The short message could not be delivered before it expired.
- **Delivery notification for this address is not supported.** This can occur if the originating network supports delivery receipts but is unable to acquire the appropriate information for one or more destination addresses. This status is reported for each address for which this is the case.

Because actual delivery of the short message may take several hours, or even days (if, for example, the mobile terminal is turned off at the time the short message is sent), delivery receipts are returned asynchronously. Applications can either choose to have delivery receipts delivered to them automatically by supplying Oracle Communications Services Gatekeeper with a callback interface or they can choose to poll Oracle Communications Services Gatekeeper.

If the application supplies a callback interface, there are two possible outcomes:

- Oracle Communications Services Gatekeeper sends the delivery receipt and the application receives and acknowledges it
- Oracle Communications Services Gatekeeper sends the delivery receipt but the application does not acknowledge reception. In this case, Oracle Communications Services Gatekeeper stores the delivery receipt in a reliable storage. The application can poll Oracle Communications Services Gatekeeper for these receipts. Each stored delivery receipt is timestamped and, after a configurable time period, is removed.

If the application chooses not to supply a callback interface, Oracle Communications Services Gatekeeper stores the delivery receipt in reliable storage. The application can poll Oracle

Communications Services Gatekeeper for these receipts. Each stored delivery receipt is timestamped and, after a configurable time period, is removed.

To correlate a sent message with a delivery receipt from the network node, information about the message is stored in Gatekeeper for a period of time. This information has a lifespan. If the delivery receipt does not arrive prior to the expiration of the message, a cancel request for the message is sent to the SMSC.

Note: The Short Messaging communication service does not put any limitation on the size of the payload in a short message. If the network protocol used restricts the size of the payload, Oracle Communications Services Gatekeeper splits the short message into appropriately sized parts.

Processing Network-triggered Requests

Two sorts of traffic destined for an application can arrive at Oracle Communications Services Gatekeeper from the network, including:

- Delivery receipts for application-initiated sent short messages (see “[Delivery Receipts](#)”).
- Mobile-originated short messages destined for the application

In order for an application to receive short messages from the network, it must register its interest in these short messages by setting up a subscription in Oracle Communications Services Gatekeeper. A subscription, or notification, is defined by a Service Activation Number, the destination address to which the mobile sender directs the short message. This is usually a *Short Code*.

Additional criteria can be tied to the Service Activation Number, such as the first word of the text in the short message payload. For Oracle Communications Services Gatekeeper to accept a message, both the Service Activation Number and the additional criteria must match the details set up in the subscription. Each registered subscription must be unique, and subscription attempts with overlapping criteria are rejected. The application may choose either to poll Oracle Communications Services Gatekeeper for received short messages or it may include a callback interface in setting up the original subscription.

If a short message that matches a subscription arrives at Oracle Communications Services Gatekeeper from the network and the original subscription includes a call-back interface, there are two possible results:

- Oracle Communications Services Gatekeeper sends the short message to the application, and the application receives and acknowledges it. In this case Oracle Communications

Services Gatekeeper simply acknowledges the reception of the short message to the network.

- Oracle Communications Services Gatekeeper sends the short message on to the application, but the application does not acknowledge reception. In this case, Oracle Communications Services Gatekeeper can store the short message in reliable storage, if offline provisioning has occurred and a `registrationIdentifier` has been established. In such a case, Oracle Communications Services Gatekeeper acknowledges the reception of the short message to the network. If no provisioning has been done, Oracle Communications Services Gatekeeper returns an error to the network. The application can poll Oracle Communications Services Gatekeeper for any stored short messages. Messages are removed after the polling application has had the chance to consume them. Each stored short message is timestamped and, after a configurable time period, is removed from the storage.

If a short message that matches a subscription arrives at Oracle Communications Services Gatekeeper, and the original subscription does not include a callback interface, the short message is stored in reliable storage and Oracle Communications Services Gatekeeper acknowledges the reception of the short message to the network. The application can poll Oracle Communications Services Gatekeeper for any such short messages. Each stored short message is timestamped and, after a configurable time period, is removed.

If a short message arrives at Oracle Communications Services Gatekeeper and no matching subscription is found, Oracle Communications Services Gatekeeper does not acknowledge reception to the network. It is the responsibility of the network node to handle any further processing of the short message.

Note: The Short Messaging communication services do not put any limitations on the size of the payload in a message. If the network protocol used restricts the size of the payload, Oracle Communications Services Gatekeeper can concatenate short message segments into one single short message before delivering it to an application. If configured to concatenate segments, it is also configurable whether to deliver parts of the message to the application regardless if all segments have arrived to Oracle Communications Services Gatekeeper or not, and after which time period the segments are considered as lost and the message is propagated to the application.

Multiple Connections and Multiple Plug-in Instances

The SMPP plug-in for Parlay X 2.1 Short Messaging can bind as an ESME transmitter/receiver or transceiver towards an SMSC. If more than one account in the SMPP SMSC is used, create one plug-in instance for each account. If more than one SMSC is used, create a plug-in instance for each account in each of the SMSCs.

Each plug-in instance can establish a set of connections (binds) towards an SMSC. If the SMSC has a throughput limit per connection, configure the number of connections to use to meet the required overall throughput requirements. On plug-in instance level, it is possible to specify the maximum allowed number of unacknowledged SMPP operations per connection. This setting can be used to protect the SMSC from overload.

Each plug-in instance executes on all Network Tier servers and a shared storage is used, so network-triggered messages and delivery notifications can be accepted by all Network Tier servers and match them with all application subscriptions and thus create a robust configuration with high availability.

Supported Network Protocols

Off the shelf, the Short Messaging communication service can be configured to support the following network protocol:

- [SMPP v3.4](#)

SMPP v3.4

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to an SMSC using SMPP v3.4. See “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports. See [System Administrator’s Guide](#) for a description of how the plug-in connects to an SMPP v.3.4 compliant SMSC.

Note: The SMPP protocol expects the sender name value to be in ASCII characters. The use of non-ASCII characters may cause the request to become garbled or even to be removed at the SMSC.

Short Code Translation

A common feature of messaging-capable networks is the use of short codes and message prefixes to help route traffic and to make access to certain features easier for the end user. Instead of having to use the entire address, users can enter shorter sequences when they dial, which are then mapped to the full address in the network. The Parlay X 2.1 Short Messaging to SMPP communication service supports short codes and message prefixes, which allow the same short code to be mapped to multiple addresses, based on what is prepended to the enclosed message.

Examples include:

Short message sent to 1243 with the first word in the message being WEATHER routes the message to an application that provides a weather service.

Short message sent to 4567 with the first word in the message being NEWS routes the message to an application that provides a news service.

Configuration Specifics for the Parlay X 2.1 Short Messaging Communication Service

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Parlay X 2.1 Short Messaging communication service, including:

- [Charging Data Records](#)
- [Event Data](#)
- [Statistics](#)

Charging Data Records

Generation of CDRs

There are three Short Messaging/SMPP-specific CDRs. They occur when the following criteria are met:

- After a `sendSms` is sent from Oracle Communications Services Gatekeeper to the network, using either plug-in
- After a `reportNotification` is sent from the network to Oracle Communications Services Gatekeeper, indicating that a delivery receipt has been returned for the application, using either plug-in
- When a mobile-originated message has been successfully delivered to the application

Event Data

[Table 7-1](#) lists EDR IDs created by the Short Messaging/SMPP communication service. This list does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 7-1 Event types emitted in the Short Messaging/SMPP communication service

EdrId	Method Called
6000	notifySmsDeliveryReceipt
6001	notifySmsReception
7000	sendSms
7001	sendSmsLogo
7002	sendSmsRingtone
7003	startSmsNotification
7004	stopSmsNotification
7005	sendSubmit
7006	receivedSMSDeliveryReport
7007	receivedMobileOriginatedSMS
7008	sendDeliverSMResp
7009	sendSubmitMulti
7010	sendCancel

Statistics

[Table 7-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2.1 Short Messaging/SMPP communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 7-2 Transaction types for Parlay X 2.1 Short Messaging communication service

Method	Transaction type
sendSms	TRANSACTION_TYPE_MESSAGING_SEND
sendSmsLogo	TRANSACTION_TYPE_MESSAGING_SEND
sendSmsRingtone	TRANSACTION_TYPE_MESSAGING_SEND
receivedMobileOriginatedSMS	TRANSACTION_TYPE_MESSAGING_RECEIVE

Supported Address Schemes

The Parlay X 2.1 Short Messaging/SMPP communication service supports the `tel:` address scheme.

Parlay X 2.1 Short Messaging Communication Service

Extended Web Services Binary SMS Communication Service

This chapter describes the Extended Web Services Binary SMS Communication Service in detail.

- [An Overview of the EWS Binary SMS Communication Service](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the EWS Binary SMS Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the EWS Binary SMS Communication Service

The Extended Web Services Binary SMS communication service allows applications to use Short Messaging to send and receive generic binary object attachments, such as vCards. An application can:

- Send short messages with binary attachments to one or more destination addresses
- Subscribe and unsubscribe for network-triggered binary short messages with binary attachments
- Receive network-triggered short messages with binary attachments

The actual message element is made up of an array of UDH and message parts, encoded in Base64. See “3rd Generation Partnership Project; Technical Specification Group Terminals; Technical realization of the short message Service (SMS); (Release 6) 3GPP 23.040 Version 6.5.0”, <http://www.3gpp.org/ftp/Specs/html-info/23040.htm>.

The send message operation gives an application the flexibility to manipulate the SMPP UDH and message data. Both the UDH and message data elements are optional, but the overall element, `binaryMessage`, is required. The contents of the UDH and the message can be of any binary data, although any byte array should be less than 140 bytes due to SMPP limitations, and the number of `BinaryMessage` arrays should be less than the `SegmentsLimit` specified in OAM. The default value is 1024.

The notification operation gives the application access to an array of SMPP UDHs, the SMPP DCS, the protocol identifier according to 3GPP 23.040 Version 6.5.0, and other data such as sender address, destination address and timestamp of the message.

Send Receipts

Send receipts are returned to the application synchronously. Send receipts are acknowledgements that the network node has received the short message from the application by means of Oracle Communications Services Gatekeeper. Although a single short message may be sent to multiple destination addresses, normally only one send receipt is returned to the application by Oracle Communications Services Gatekeeper. The receipt is returned synchronously in the response message to the `sendBinarySms` operation.

Delivery Receipts

Delivery receipt notifications can be set up using this operation, but the actual asynchronous delivery of receipts is accomplished using the Parlay X 2.1 Short Messaging interface. See [Delivery Receipts](#) in [Chapter 7, “Parlay X 2.1 Short Messaging Communication Service”](#) for more information on receiving delivery receipts.

Supported Network Protocols

The Extended Web Services Binary SMS communication service supports the SMPP v3.4 network protocol. When using this protocol, Oracle Communications Services Gatekeeper connects to an SMSC using SMPP v3.4. See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports. See [Oracle Communications Services](#)

[Gatekeeper System Administrator's Guide](#) for a description of how the plug-in connects to an SMPP v.3.4-compliant SMSC.

Note: SMPP expects the sender name value to be in ASCII characters. The use of non-ASCII characters may cause the request to become garbled or even be removed at the SMSC.

Configuration Specifics for the EWS Binary SMS Communication Service

Communication services share many common features, covered in “[Introducing Communication Services](#)” in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Extended Web Service Binary SMS communication service, including:

- [Charging Data Records](#)
- [Event Data](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

There are two Binary SMS specific CDRs. They occur when the following criteria are met:

- After a `sendBinarySms` is sent from Oracle Communications Services Gatekeeper to the network
- When a network-triggered message has been successfully delivered to the application.

Event Data

[Table 8-1](#) lists EDR IDs created by the EWS Binary SMS/SMPP communication service. This list does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 8-1 Event types emitted in the Binary SMS /SMPP communication service

EdrId	Method Called
7101	sendBinarySms
7201	startBinarySmsNotification
7202	stopBinarySmsNotification
7203	receivedMobileOriginatedBinarySMS
7205	notifyBinarySmsReception

[Table 8-2](#) lists EDR IDs used by the EWS Binary SMS/SMPP communication service that are inherited from the Parlay X 2.1/SMPP communication service:

Table 8-2 Inherited event types emitted in the Binary SMS Binary SMS /SMPP communication service

EdrId	Method Called
7005	sendSubmit
7009	sendSubmitMulti
7010	sendCancel

Statistics

[Table 8-3](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the EWS Binary SMS communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 8-3 Transaction types for EWS Binary SMS communication service

Method	Transaction type
sendBinarySMS	TRANSACTION_TYPE_MESSAGING_SEND
receivedMobileOriginatedBinarySMS	TRANSACTION_TYPE_MESSAGING_RECEIVE

Supported Address Schemes

The EWS Binary SMS communication service supports the `tel:` address scheme.

Parlay X 2.1 Multimedia Messaging Communication Service

This chapter describes the Parlay X 2.1 Multimedia Messaging communication service in detail:

- [An Overview of the Parlay X 2.1 Multimedia Messaging Communication Service](#)
 - [Processing Application-initiated Requests](#)
 - [Processing Network-triggered Requests](#)
 - [Supported Network Protocols](#)
 - [Short Code Translation](#)
- [Configuration Specifics for the Parlay X 2.1 Multimedia Messaging Communication Services](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 2.1 Multimedia Messaging Communication Service

The SOAP Service Facade Multimedia Messaging communication service implements the Parlay X 2.1 Multimedia Messaging set of interfaces. For the exact version of the standard this

communication service supports, see [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Multimedia Messaging interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using a Multimedia Messaging communication service, an application can:

- Send multimedia messages to one or many destination addresses. The payload in these multimedia messages can be any type that can be specified using MIME, including multipart messages
- Sign up to be notified that delivery receipts for sent multimedia messages have been received from the network
- Receive delivery receipts on sent multimedia messages that have arrived from the network
- Explicitly query Oracle Communications Services Gatekeeper for delivery receipts on sent multimedia messages
- Sign up to be notified if specified multimedia messages for the application have been received from the network
- Receive notifications that specified multimedia messages for the application have arrived from the network. These notifications do not include the message payload, but they do provide a message ID
- Explicitly poll Oracle Communications Services Gatekeeper for multimedia messages sent to the application that have arrived from the network and been stored in Oracle Communications Services Gatekeeper.

The polling capability must be set up in advance. It is controlled through the combined use of parameters send in the request and OAM attributes See [Table 9-1](#) for more information:.

Table 9-1 Setting up polling functionality

RequestDeliveryReportFlag OAM Attribute	Means
0	No polling functionality is available
1	Polling is available if: <ul style="list-style-type: none">• The application’s initial send request includes a notification endpointor• The application includes a tunneled parameter with the value <code>com.bea.wlcp.wlng.plugin.multimediamessaging.RequestDeliveryReportFlag</code> set to <code>true</code> in the initial send request or such a parameter is added during policy evaluation
2	Polling functionality is always available

Requests can flow in two directions using the Multimedia Messaging communication service: from the application to the network (called application-initiated or mobile terminated) and from the network to the application (called network-triggered or mobile originated). Both of these scenarios are covered below.

Processing Application-initiated Requests

After an application has sent a multimedia message to one or more destination addresses, two different types of response can be returned:

- [Send Receipts](#)
- [Delivery Receipts](#)

Send Receipts

Send receipts are acknowledgements that the network node has received the multimedia message from the application by means of Oracle Communications Services Gatekeeper. Although a single multimedia message may be sent to multiple destination addresses, normally only one send receipt is returned to the application by Oracle Communications Services Gatekeeper. The receipt is returned synchronously in the response message to the `sendMessage` operation.

Delivery Receipts

delivery receipts contain the delivery status of the multimedia message, that is, whether the multimedia message has actually been delivered by the network to the mobile terminal. There is one delivery receipt per destination address, with one of three possible states:

- Successful
- Unsuccessful. The multimedia message could not be delivered before it expired.
- Delivery notification for this address is not supported. This can occur if the originating network supports delivery receipts but is unable to acquire the appropriate information for one or more destination addresses. This status is reported for each address for which this is the case.

Because actual delivery of the multimedia message may take several hours, or even days (if, for example, the mobile terminal is turned off at the time the multimedia message is sent), delivery receipts are returned asynchronously. Applications can either choose to have delivery receipts delivered to them automatically by supplying Oracle Communications Services Gatekeeper with a callback interface or they can choose to poll Oracle Communications Services Gatekeeper.

If the application supplies a callback interface, there are two possible outcomes:

- Oracle Communications Services Gatekeeper sends the delivery receipt and the application receives and acknowledges it
- Oracle Communications Services Gatekeeper sends the delivery receipt but the application does not acknowledge reception. In this case, Oracle Communications Services Gatekeeper stores the delivery receipt in temporary in-memory storage. The application can poll Oracle Communications Services Gatekeeper for these Receipts. Each stored delivery receipt is timestamped and, after a configurable time period, is removed.

If the application chooses not to supply a callback interface, Oracle Communications Services Gatekeeper stores the delivery receipt in temporary in-memory storage. The application can poll Oracle Communications Services Gatekeeper for these Receipts. Each stored delivery receipt is timestamped and, after a configurable time period, is removed.

Note: The Multimedia Messaging communication service does not put any limitation on the size of the payload in a multimedia message. If the network restricts the size of the payload, Oracle Communications Services Gatekeeper does not split the multimedia message.

Processing Network-triggered Requests

Two sorts of traffic destined for an application can arrive at Oracle Communications Services Gatekeeper from the network, including:

- Delivery receipts for application-initiated sent multimedia messages (see “[Delivery Receipts](#)”).
- Mobile-originated multimedia messages destined for the application

For an application to receive multimedia messages from the network, it must register its interest in these multimedia messages by setting up a subscription in Oracle Communications Services Gatekeeper. A subscription, or notification, is defined by a Service Activation Number, the destination address of the multimedia message.

Note: The Service Activation Number may possibly be translated by some mechanism, such as short codes, in the telecom network.

Additional criteria can be tied to the Service Activation Number, such as the start of the first plain/text part in the multimedia message payload or the subject of the multimedia message. For the message to be accepted by Oracle Communications Services Gatekeeper, both the Service Activation Number and any additional criteria must match the subscription. Each registered subscription must be unique, and subscription attempts with overlapping criteria are rejected. The application may choose either to poll Oracle Communications Services Gatekeeper for received multimedia messages (if polling is enabled) or it may include a callback interface when it sets up the original subscription.

If a multimedia message that matches a subscription arrives at Oracle Communications Services Gatekeeper from the network and the original subscription includes a callback interface, there are two possible results:

- Oracle Communications Services Gatekeeper sends the notification that the multimedia message has arrived on to the application, and the application receives and acknowledges it. In this case, Oracle Communications Services Gatekeeper stores the multimedia message in temporary in-memory storage and acknowledges the reception of the multimedia message to the network. Each stored multimedia message is timestamped and, after a configurable time period, is removed. The application can poll Oracle Communications Services Gatekeeper for any stored multimedia messages.
- Oracle Communications Services Gatekeeper sends the notification that the multimedia message has arrived on to the application, but the application does not acknowledge reception. Oracle Communications Services Gatekeeper does not acknowledge reception to

the network. In this case, it is the responsibility of the network node to handle any further processing of the multimedia message.

If a multimedia message that matches a subscription arrives at Oracle Communications Services Gatekeeper and the original subscription does not include a callback interface, but polling is available, the multimedia message is stored in temporary in-memory storage and Oracle Communications Services Gatekeeper acknowledges the reception of the multimedia message to the network. The application can poll Oracle Communications Services Gatekeeper for any such multimedia messages. Each stored multimedia message is timestamped and, after a configurable time period, is removed.

If a multimedia message arrives at Oracle Communications Services Gatekeeper and no matching subscription is found and polling is not otherwise enabled, Oracle Communications Services Gatekeeper does not acknowledge reception to the network. It is the responsibility of the network node to handle any further processing of the multimedia message.

Note: Multimedia Messaging communication services do not put any limitations on the size of the payload in a multimedia message. If the network protocol used restricts the size of the payload, Oracle Communications Services Gatekeeper does not concatenate multimedia message segments into one single multimedia message before delivering it to an application. Oracle Communications Services Gatekeeper regards these as independent multimedia messages.

Supported Network Protocols

Off the shelf, the Multimedia Messaging communication service is configured to support the following network protocol:

- [MM7](#)

MM7

Oracle Communications Services Gatekeeper connects to an MMSC using MM7. See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Short Code Translation

A common feature of messaging-capable networks is the use of short codes and message prefixes to help route traffic and to make access to certain features easier for the end user. Instead of having to use the entire address, users can enter shorter sequences when they dial, which are then mapped to the full address in the network. The Parlay X 2.1 Multimedia Messaging to MM7

communication service supports short codes and message prefixes, which allow the same short code to be mapped to multiple addresses, based on what is prepended to the enclosed message.

Configuration Specifics for the Parlay X 2.1 Multimedia Messaging Communication Services

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Multimedia Messaging communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

The Multimedia Messaging/MM7 communication service writes CDRs in the following conditions:

- After a `sendMessage` request has entered the network plug-in from the north
- After a `notifyMessageDeliveryReceipt` has entered the network plug-in from the south
- After a `notifyMessageReception` has been delivered to the application
- When there is an error

Event Data Records

Multimedia Messaging using MM7 uses the standard EDR mechanism, documented in [Table 9-2](#). For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 9-2 Event types emitted in Multimedia Messaging/MM7 communication service

EdrId	Meaning
8100	An MO message has arrived from the network
8101	An MO delivery receipt has arrived from the network.
8102	The application has requested that a notification be started
8103	The application has requested that a notification be stopped
8104	The application has polled for a list of received messages
8106	The application has polled for actual messages, returned as attachments.

Statistics

[Table 9-3](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2. Multimedia Messaging/MM7 communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 9-3 Transaction types for Parlay X 2.1 Multimedia Messaging/MM7 communication service

Method	Transaction type
sendMessage	TRANSACTION_TYPE_MESSAGING_MMS_SEND
deliver	TRANSACTION_TYPE_MESSAGING_MMS_RECEIVE
deliveryReport	TRANSACTION_TYPE_MESSAGING_MMS_RECEIVE

Supported Address Schemes

The Parlay X 2.1 Multimedia Messaging/MM7 communication service supports the `tel:` and `mailto:` address schemes.

Parlay X 3.0 Payment Communication Services

This chapter describes the Parlay X 3.0 Payment communication service in detail:

- [An Overview of the Parlay X 3.0 Payment Communication Service](#)
 - [Processing Direct Queries/Application-initiated Requests](#)
 - [Processing Notifications/Network-triggered Requests](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 3.0 Payment Communication Services](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 3.0 Payment Communication Service

The SOAP Service Facade Payment communication service implements the Parlay X 3.0 Payment set of interfaces. For the exact version of the standard these communication services support, see [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Payment interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using a Payment communication service, an application can:

- Charge and refund accounts directly
- Operate on reservations:
 - Make reservations
 - Charge reservation
 - Release reservations
- Charge multiple accounts concurrently

All charging is done on accounts. The unit of charge is specified as a given currency or a charging code.

Processing Direct Queries/Application-initiated Requests

If an application makes a request to interact directly with an account, Oracle Communications Oracle Communications Services Gatekeeper sends the request to the network node capable of handling the request. The request does not return until the targeted account has been updated.

Processing Notifications/Network-triggered Requests

There are no notifications are network-triggered requests for this communications service.

Supported Network Protocols

Off the shelf, the Payment communication service can be configured to support the following network protocols:

- Diameter

Diameter

The Payment communication service acts as an on-line charging application using the Diameter Ro interface. See “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Oracle Communications Services Gatekeeper connects to the Diameter server, and acts as a single on-line charging application.

A reservation expires after a given time. The expiry mechanism provided by the Storage Service is used. If the store entry expires, the reservation is cancelled.

Some Diameter servers, for example Oracle Billing and Revenue Management, mandates that a refund operation is correlated with previous charge operation. The Parlay X 2.1 Payment interface does not provide any correlation between charge operations and refund operations. The tunnelled parameter `session-id` has been added in order to correlated these requests. When an application calls `chargeAmount`, the tunnelled parameter `session-id` is returned in the SOAP header. An application should use this `session-id` in subsequent requests to `refundAmount` to correlate the two requests. If the application does not provided the tunneled parameter, it is the responsibility of the Diameter server to either accept to deny the request. If the request is denied, the application receives a `ServiceException`.

Configuration Specifics for the Parlay X 3.0 Payment Communication Services

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Payment communication services, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

In the Payment/Diameter communication service, CDRs are written when the response to a request is successfully delivered to the application. All methods trigger the generation of CDRs:

- `chargeAmount`
- `refundAmount`

- chargeSplitAmount
- reserveAmount
- reserveAdditionalAmount
- chargeReservation
- releaseReservation

Event Data Records

Events for Payment/Diameter are documented in [Table 10-1](#). For more information, see [Appendix A, “Events, Alarms, and Charging.”](#).

Table 10-1 Event types emitted in Payment/Diameter communication service

EdrId	Description
15001	chargeAmount
15002	refundAmount
15003	chargeSplitAmount
15004	reserveAmount
15005	reserveAdditionalAmount
15006	chargeReservation
15007	releaseReservation

Statistics

[Table 10-2](#) outlines the correlation between the methods being invoked from the application and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 3.0 Payment/Diameter communication service:

Table 10-2 Transaction types for Parlay X 3.0 Payment/Diameter communication service

Method	Transaction type
chargeAmount	TRANSACTION_TYPE_CHARGING_DIRECT
chargeSplitAmount	TRANSACTION_TYPE_CHARGING_DIRECT

Table 10-2 Transaction types for Parlay X 3.0 Payment/Diameter communication service

Method	Transaction type
refundAmount	TRANSACTION_TYPE_CHARGING_DIRECT
reserveAmount	TRANSACTION_TYPE_CHARGING_RESERVED_STRING
reserveAdditionalAmount	TRANSACTION_TYPE_CHARGING_RESERVED_STRING
chargeReservation	TRANSACTION_TYPE_CHARGING_RESERVED_STRING

Supported Address Schemes

The Parlay X 3.0 Payment/Diameter communication service supports the `tel:` address scheme.

Parlay X 2.1 Terminal Location Communication Services

This chapter describes the Parlay X 2.1 Terminal Location communication services in detail:

- [An Overview of the Parlay X 2.1 Terminal Location Communication Service](#)
 - [Processing Direct Queries/Application-initiated Requests](#)
 - [Processing Notifications/Network-triggered Requests](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Parlay X 2.1 Terminal Location Communication Services](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Parlay X 2.1 Terminal Location Communication Service

The SOAP Service Facade Terminal Location communication services implement the Parlay X 2.1 Terminal Location set of interfaces. For the exact version of the standard these communication services support, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Terminal Location interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using a Terminal Location communication service, an application can:

- Ask for the location of one or many terminals by polling
- Ask for the distance between a given terminal and a given position
- Sign up to be notified when a terminal enters or leaves a specified geographical area
- Receive notifications when the terminal enters or leaves the specified geographical area
- Sign up to be notified periodically about the location of a terminal
- Receive periodic location notifications about the location of a terminal

The application can specify a number of parameters concerning the nature of the notification. These include:

- Requested accuracy
- Accepted accuracy
- Accepted response time
- Maximum age of location data
- Tolerance, which expresses the priority of response time versus accuracy
- Minimum frequency of notifications
- Duration of notifications
- Maximum number of notifications

Ultimately, the nature of the information that is available to the application depends on the network node and network protocol used. Not all networks or protocols support all operations. The accuracy of the location provided, the response times, and the frequency of notification are all also dependent on the specifics of the protocol and network node used.

Processing Direct Queries/Application-initiated Requests

If an application directly queries Oracle Communications Services Gatekeeper for the location of a terminal or group of terminals, Oracle Communications Services Gatekeeper sends the request

to the network node, and the location information is sent back synchronously in the response to the request.

Processing Notifications/Network-triggered Requests

If an application registers for periodic or geographically-defined notifications, information for the application (which may or may not include the location data for one or more terminals) arrives at Oracle Communications Services Gatekeeper from the network. The notification is passed on to the application. If the application acknowledges the reception of the notification, Oracle Communications Services Gatekeeper acknowledges the reception of the notification to the network. If the application does not acknowledge the reception of the notification, Oracle Communications Services Gatekeeper does not acknowledge the reception of the notification to the network.

Supported Network Protocols

Off the shelf, the Terminal Location communication service can be configured to support the following network protocols:

- [MLP 3.0 and 3.2](#)

MLP 3.0 and 3.2

Oracle Communications Services Gatekeeper acts as an LCS/MLS client. It is possible to use either version of MLP. See “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Oracle Communications Services Gatekeeper can be configured to act in Standard Location or Emergency Location Immediate mode on the node level.

Oracle Communications Services Gatekeeper connects to the location server using HTTP. It always acts as a single LCS/MLS client towards the location server.

Note: When using MLP 3.0, triggered notifications are not supported.

Configuration Specifics for the Parlay X 2.1 Terminal Location Communication Services

Communication services share many common features, covered in “[Introducing Communication Services](#)” in *Concepts and Architectural Overview*, but each one has a few characteristics that are

specific only to that service. This section describes those specific features for the Terminal Location communication services, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

In the Terminal Location/MLP communication services, CDRs are written:

- When the response to a polling request (of whatever type) is successfully delivered to the application
- When a notification is received from the network
- When an error occurs

Event Data Records

Events for Terminal Location using MLP are documented in [Table 11-1](#). For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 11-1 Event types emitted in Terminal Location/MLP communication services

EdrId	Description
9001	getLocation
9002	getTerminalDistance
9003	getLocationForGroup
9004	sendLocationRequest
9011	LocationEnd

Table 11-1 Event types emitted in Terminal Location/MLP communication services

EdrId	Description
9012	LocationError
9013	LocationNotification

Statistics

[Table 11-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2.1 Terminal Location/MLP communication services:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 11-2 Transaction types for Parlay X 2.1 Terminal Location/MLP communication services

Method	Transaction type
getLocation	TRANSACTION_TYPE_USER_LOCATION
getLocationForGroup	TRANSACTION_TYPE_USER_LOCATION
getLocationDistance	TRANSACTION_TYPE_USER_LOCATION
locationNotification	TRANSACTION_TYPE_USER_LOCATION

Supported Address Schemes

The Parlay X 2.1 Terminal Location/MLP communication services support the `tel:` address scheme.

Parlay X 2.1 Terminal Location Communication Services

Parlay X 2.1 Presence Communication Service

This chapter describes the Parlay X 2.1 Presence communication service in detail:

- [An Overview of the Parlay X 2.1 Presence Communication Service](#)
 - The Client as Presence Consumer
 - The Client as Presence Supplier
 - Supported Network Protocols
- Configuration Specifics for the Parlay X 2.1 Presence Communication Service
 - Charging Data Records
 - Event Data Records
 - Statistics
 - Supported Address Schemes

An Overview of the Parlay X 2.1 Presence Communication Service

The SOAP Service Facade Presence communication service implements the both the watcher aspect and the presentity aspect of the Parlay X 2.1 Presence set of interfaces. For the exact version of the standard these communication services support, see [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade Presence interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Presence information is a collection of data on an end user's status, including such things as current activity, environment, available communication means, and contact addressees. Using the Presence functionality, an application can function as a client in two modes: as a watcher or as a presentity. A watcher is a client that is interested in consuming presence information. A presentity is a client that allows its presence information to be delivered to watchers.

The Client as Presence Consumer

An application acting as a watcher can:

- Subscribe to obtain presence data. Each subscription requires authorization by the presentity. The authorization is returned asynchronously via the notification interface.
- Choose to acquire presence information when a subscription has been established using:
 - Direct synchronous polling. This is only effective for a single presentity. Groups are not supported
 - Specific notifications. These can be used for a single presentity. The watcher sets a notification trigger based on certain user presence attribute changes.

Possible attribute types include:

- Activity (User's status: Available, Busy, At Lunch, etc.)
- Place (User's current location: Home, In a Public Place, etc.)
- Privacy (Degree of privacy the user has: Surrounded by Others, Alone and Can Talk Openly, etc.)
- Sphere (User's personal status: In his Work Capacity; In his Personal Capacity)
- Communication means (Type of communication client preferred: Phone, Email, SMS, etc.)
- Other (name-value pair for arbitrary information)

Non-attribute notification parameters can include:

- Maximum frequency of notifications
- Duration of time during which notifications should occur
- Count - the maximum number of notifications

- Whether status should be checked immediately after notification setup
- End notifications. In this case, the subscription to the presentity is retained, but the specific notification is ended
- Receive information that:
 - The initial conditions of the notification setup have been met (count or duration) and this specific setup has been ended
 - The subscription itself has ended

The Client as Presence Supplier

An application acting as a presentity can:

- Publish present information
- Get a list of new watchers who have asked to subscribe to the client's presence information
- Approve new watchers and update the subscriptions of current watchers
- Get a list of currently subscribed watchers
- Block the subscription of a currently subscribed watcher

Note: The Presentity functionality requires a presence server in the underlying network. To approve new watchers and update the subscriptions of current watchers, there must also be a data manipulation server (DMS) in the underlying network. The block functionality is supported in two modes, one using a DMS and one not.

Supported Network Protocols

Off the shelf, the Presence communication service is configured to support the following network protocol:

- SIP

Configuration Specifics for the Parlay X 2.1 Presence Communication Service

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are

specific only to that service. This section describes those specific features for the Presence communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

There are two Presence/SIP-specific CDRs. They are generated when the following criteria are met:

- After the result of a poll for presence is successfully returned to the application
- After a notification for presence information is successfully sent to the application

CDR Details

A Presence CDR contains the standard information with this additional information in the `additional_info` field for a notification call:

Table 12-1 Additional Info in Presence CDR

Column	Description
<code>additional_info</code>	Endpoint (string)

Event Data Records

[Table 12-2](#) lists EDR IDs created by the Presence/SIP communication service. This list does not include EDRs created when exceptions are thrown. For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 12-2 Event types emitted in the Presence/SIP communication service

EdrId	Method Called
2000	notifyReceived
2001	makeNotifySubscriptionCallback (includes Endpoint, string)
2002	makeSubscriptionEndedCallback (includes Endpoint, string)
2003	makeStatusChangedCallback (includes Endpoint, string)
2004	makeStatusEndCallback (includes Endpoint, string)
2005	subscribePresence (processes from application)
2006	getUserPresence
2007	startPresenceNotification
2008	endPresenceNotification
2009	publish
2010	blockSubscription
2011	getMyWatchers
2012	getOpenSubscriptions
2013	updateSubscriptionAuthorization
2015	onRequest (Watcher Info Notify)

Statistics

[Table 12-3](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2.1 Presence/SIP communication service:

Note: Method names for network-initiated requests are specified by the internal Oracle Communications Services Gatekeeper name, which is not necessarily the same as the message from the network.

Table 12-3 Transaction types for Parlay X 2.1 Presence/SIP communication service

Method	Transaction type
getUserPresence	TRANSACTION_TYPE_PRESENCE_SERVICE_INITIATED
makeStatusChangedCallback	TRANSACTION_TYPE_PRESENCE_NETWORK_INITIATED

Supported Address Schemes

The Parlay X 2.1 Presence/SIP communication service supports the `SIP URI` address scheme.

EWS Subscriber Profile Communication Service

This chapter describes the Extended Web Services Subscriber Profile communication service in detail:

- [An Overview of the Extended Web Services Subscriber Profile Communication Service](#)
 - [How It Works](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the Extended Web Services Subscriber Profile Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Extended Web Services Subscriber Profile Communication Service

There is no standard available for this service, although it uses elements from preliminary Parlay X drafts (preliminary as of January 2008) for this functionality.

Note: The RESTful Service Facade Subscriber Profile interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using these interfaces an application can:

- Retrieve the specific value for a particular property belonging to a subscriber profile stored in an LDAP data source.
- Retrieve an entire subscriber profile from an LDAP data source, subject to SLA filtering.

How It Works

The Extended Web Services Subscriber Profile communication service can be used by applications that need to retrieve subscriber profile data from an LDAP server attached to the underlying network.

Supported Network Protocols

Off-the shelf, the Extended Web Services Subscriber Profile communication service can be configured to support the following network protocol:

- [LDAPv3](#)

LDAPv3

When Oracle Communications Services Gatekeeper is configured to use this protocol, it connects to an LDAP server. See “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Extended Web Services Subscriber Profile Communication Service

Communication services share many common features, covered in “[Introducing Communication Services](#)” in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the Extended Web Services Subscriber Profile communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

There are two specific CDRs associated with the Extended Web Services Subscriber Profile/LDAP communication service. They occur when the following criteria are met:

- After Oracle Communications Services Gatekeeper has returned a full or partial subscriber profile to an application based on one or more attributes requested by that application.
- After Oracle Communications Services Gatekeeper has returned a subscriber profile to an application based on the ID of the profile.

Event Data Records

The EDRs produced by the Extended Web Services Subscriber Profile/LDAPv3 communication service are documented in [Table 13-1](#). This does not include EDRs created when exceptions are thrown. For more information on the contents of standard EDRs, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 13-1 Event types emitted in the EWS Subscriber Profile/LDAP communication service

EdrId	Method Called
13001	get
13002	getProfile

Statistics

[Table 13-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the EWS Subscriber Profile/LDAP communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 13-2 Transaction types for EWS Subscriber Profile/LDAP communication service

Method	Transaction type
get	TRANSACTION_TYPE_SUBSCRIBER_PROFILE
getProfile	TRANSACTION_TYPE_SUBSCRIBER_PROFILE

Supported Address Schemes

The EWS Subscriber Profile/LDAP communication service supports the `tel:`, `id:`, `imsi:` and `ipv4:` address schemes.

Extended Web Services WAP Push Communication Service

This describes the Extended Web Services WAP Push communication service in detail:

- [An Overview of the Extended Web Services WAP Push Communication Service](#)
 - [Supported Network Protocols](#)
- [Configuration Specifics for the WAP Push Communication Service](#)
 - [Charging Data Records](#)
 - [Event Data Records](#)
 - [Statistics](#)
 - [Supported Address Schemes](#)

An Overview of the Extended Web Services WAP Push Communication Service

The SOAP Service Facade WAP Push communication service implements the Oracle Extended Web Services WAP Push interface. Although the specific interface is not standardized, it uses standardized elements. For more information on these elements, see “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview*, another document in this set.

Note: The RESTful Service Facade WAP Push interfaces provide RESTful access to this same functionality. The internal representations are identical, and for the purposes of creating SLAs and reading CDRs, etc., they are the same.

Using the WAP Push communication service, an application can:

- Send a WAP Push message
- Send a replacement WAP Push message
- Ask to be notified asynchronously of the status of WAP Push messages that have been sent. The possible values returned include:
 - Rejected: The message was not accepted
 - Pending: The message is in process
 - Delivered: The message was successfully delivered to the end-user
 - Undeliverable: The message could not be delivered because of a problem
 - Expired: The message reached the maximum age allowed by server policy or could not be delivered by the time specified in the push submission
 - Aborted: The mobile device aborted the message
 - Timeout: The delivery process timed out
 - Cancelled: The message was cancelled through the cancel operation
 - Unknown: The server does not know the state of the message

Note: The result notification message is only sent if the initial push submission was accepted for processing. One result notification message is sent per destination address.

Supported Network Protocols

Off the shelf, the Extended Web Services WAP Push communication service can be configured to support the following network protocol:

- [Push Access Protocol \(PAP\) 2.0](#)

Push Access Protocol (PAP) 2.0

When Oracle Communications Services Gatekeeper is configured to use this protocol, the EWS WAP Push communication service supports a subset of its operations including:

- push-message: Submits a message to be delivered. This operation is also used to send a replacement message
- push-response: The response to the push-message operation. This response includes a code specifying the immediate status of the message submission, of the following general types:
 - 1xxx Success: The action was successfully received, understood, and accepted

- 2xxx Client Error: The request contains bad syntax or cannot be fulfilled
- 3xxx Server Error: The server failed to fulfil an apparently valid request
- 4xxx: Service Failure: The service could not be performed. The operation may be retried
- resultnotification-message: Specifies the final outcome of a specific message for a specific recipient. Sent only if the initial request includes the URL to which this notification is to be delivered. Includes both textual indication of state and a status code including the following general types:
 - 1xxx Success: The action was successfully received, understood, and accepted
 - 2xxx Client Error: The request contains bad syntax or cannot be fulfilled
 - 3xxx Server Error: The telecom network node failed to fulfil an apparently valid request
 - 4xxx: Service Failure: The service could not be performed. The operation may be retried
 - 5xxx: Mobile Device Abort: The mobile device aborted the operation.
- resultnotification-response: The response to the result notification. This response includes a code specifying the status of the notification
 - 1xxx Success: The action was successfully received, understood, and accepted
 - 2xxx Client Error: The request contains bad syntax or cannot be fulfilled
- badmessage-response: A response indicating that request is unrecognizable or is of a protocol version that is not supported. This response contains either a 3002 code (Version not supported) or a 2000 code (Bad Request). In the case of Bad Request, a fragment of the unrecognizable message is included in the response

See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the protocol standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the WAP Push Communication Service

Communication services share many common features, covered in [“Introducing Communication Services”](#) in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the WAP Push communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

There are two WAP Push-specific CDRs. They are written:

- When the `sendPushMessage` response returns from the network
- When a `sendResultNotificationMessage` response returns from the application

Event Data Records

[Table 14-1](#) lists of EDR IDs created by the EWS WAP Push communication service. For more information see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 14-1 Event types emitted by the EWS WAP Push Message communication service

EdrId	Meaning
14001	sendPushMessage
14002	sendResultNotificationMessage

Statistics

[Table 14-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Oracle Communications Services Gatekeeper for the Parlay X 2. Multimedia Messaging communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 14-2 Transaction types for EWS WAP Push Message communication service

Method	Transaction type
sendPushMessage	TRANSACTION_TYPE_MESSAGE_SENDER_SEND
sendResultNotificationMessage	TRANSACTION_TYPE_MESSAGE_SENDER_NOTIFY

Supported Address Schemes

The EWS WAP Push Message communication service supports the `tel:` and `wapuser:` address schemes.

Native MM7 Communication Service

This chapter describes the native MM7 communication service in detail.

An Overview of the Native MM7 Communication Service

The native MM7 communication service implements the 3GPP MM7 standard. For more information, see [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview*, another document in this set.

Using the native MM7 communication service, an application can:

- Send a multimedia message to one or many destination addresses. The payload in these multimedia messages can be any type that can be specified using MIME, including multipart messages. If a subscription for notifications has been previously set up using Gatekeeper OAM, the request can also specify that a delivery report and/or a read report should be returned later in relation to this message.
- Receive delivery reports on sent multimedia messages that have arrived from the network, if this option has been set up using Gatekeeper OAM.
- Receive read-reply reports on sent multimedia messages that have arrived from the network, if this option has been set up using Gatekeeper OAM.
- Receive multimedia messages from the network, if this option has been set up using Gatekeeper OAM.

Requests can flow in two directions using the native MM7 communication service: from the application to the network (called application-initiated or mobile terminated) and from the network to the application (called network-triggered or mobile originated).

Status Reports

There are two types of status reports that can be returned to the application from the network via Gatekeeper. Both are returned asynchronously, using callback information provided when the notification is set up using OAM. If the network sends a report but no notification has been set up, Gatekeeper sends the network an error code indicating permanent failure.

- [Delivery Reports](#)
- [Read-Reply Report](#)

Delivery Reports

Delivery reports are acknowledgements that the network node has in some way handled the message from the application that was submitted by means of Gatekeeper. The report indicates the status of the message: for example, Forwarded, Expired, or Rejected. There is one delivery report per destination address. If a connection error occurs within Gatekeeper or between Gatekeeper and the application, an error code is returned to the network, which re-sends the message.

Read-Reply Report

Read-reply reports contain the final delivery status of the multimedia message; that is, whether the message has actually been delivered by the network to the mobile terminal. It also includes the status of the message at that terminal: for example, Read or Deleted without being read. Because a recipient can request that read-reply reports not be generated, lack of a read-reply report does not necessarily mean that the message has not been rendered on the recipient's terminal. There is one read-reply report per destination address. If a connection error occurs within Gatekeeper or between Gatekeeper and the application, an error code is returned to the network, which re-sends the message.

Note: The Native MM7 communication service does not put any limitation on the size of the payload in a multimedia message. If the network restricts the size of the payload, Gatekeeper does not split the multimedia message.

Network-triggered Multimedia Messages

In order for an application to receive multimedia messages from the network, it must register its interest in these messages by setting up a subscription using Gatekeeper OAM. A subscription, or notification, is defined by a destination address. For the message to be accepted by Oracle Communications Services Gatekeeper, the destination address must match the subscription. Each registered subscription must be unique, and subscription attempts with overlapping criteria are rejected. If a message with several destination addresses arrives, Gatekeeper will iterate through the list until it reaches a match or the list is exhausted.

Note: The Native MM7 communication service does not put any limitations on the size of the payload in a multimedia message. If the network protocol used restricts the size of the payload, Gatekeeper does not concatenate multimedia message segments into one single multimedia message before delivering it to an application. Gatekeeper regards these as independent multimedia messages.

Supported Network Protocols

off the shelf, the native MM7 communication service supports the following network protocol:

- [MM7](#)

MM7

Oracle Communications Services Gatekeeper connects to an MMSC using MM7. See “[Appendix A: Standards and Specifications](#)” in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Configuration Specifics for the Native MM7 Communication Service

Communication services share many common features, covered in “[Introducing Communication Services](#)” in *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the native MM7 communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)

- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

The native MM7 communication service writes CDRs in the following conditions:

- After an MMS has been successfully sent from the application to the network
- After an MMS has been successfully sent from the network to the application
- After a delivery report has been successfully delivered to the application
- After a read-reply report has been successfully delivered to the application

Event Data Records

Native MM7 uses the standard EDR mechanism, documented in [Table 15-1](#). For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 15-1 Event types emitted in native MM7 communication service

EdrId	Meaning
401000	An application-initiated message has entered the plugin.
401001	An application-initiated message has exited the plugin.
401002	A network-triggered message sent via v.1.0 has entered the plugin.
401003	A network-triggered message has exited the plugin. It is formatted according to v. 1.2.
401004	A delivery report using v. 1.0 has entered the plugin.
401005	A delivery report has exited the plugin. It is formatted according to v 1.2
401006	A read-reply report using v. 1.0 has entered the plugin.
401007	A read-reply report has exited the plugin. It is formatted according to v. 1.2

Statistics

[Table 15-2](#) outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Gatekeeper for the native MM7 communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 15-2 Transaction types for Parlay X 2.1 Multimedia Messaging/MM7 communication service

Method	Transaction type
submit	TRANSACTION_TYPE_MESSAGING_MMS_SEND
deliver	TRANSACTION_TYPE_MESSAGING_MMS_RECEIVE

Supported Address Schemes

The native MM7 communication service supports the `tel:` and `mailto:` address schemes, and supports short codes.

Native MM7 Communication Service

Native SMPP Communication Service

This chapter describes the native SMPP communication service in detail:

An Overview of the Native SMPP Communication Service

The native SMPP communication service implements the SMPP v. 3.4 standard. For more information, see [“Appendix A: Standards and Specifications”](#) in the *Concepts and Architectural Overview*, another document in this set.

Using the native SMPP communication service, an application can:

- Send a short message to one or many destination addresses.
- Cancel a previously sent message that has not yet been delivered.
- Replace a previously sent message that has not yet been delivered.
- Query the delivery status of a previously sent message.
- Receive short messages arrived from the network.
- Receive the delivery status of a previously sent message.

Requests can flow in two directions using the native SMPP communication service: from the application to the network (called application-initiated or mobile terminated) and from the network to the application (called network-triggered or mobile originated).

Applications act as SMPP ESME clients and Oracle Communications Services Gatekeeper acts as an SMPP server.

Session Handling and Provisioning

Applications can bind to Oracle Communications Services Gatekeeper as a transmitter, a receiver, or a transceiver. Oracle Communications Services Gatekeeper does not initialize binds (SMPP operation outbind).

An application can establish several parallel sessions by issuing multiple bind operations.

Applications use an application instance ID as the ESME system_id and the related password as provisioned in Oracle Communications Services Gatekeeper when binding to Oracle Communications Services Gatekeeper.

As a result of a bind operation from an application, the Native SMMP network protocol plug-in binds to the underlying SMPP server. When the application unbinds, the plug-in unbinds from the SMPP server. The plug-in can bind as a transceiver, transmitter, or receiver.

A plug-in instance can connect to exactly one SMPP server, so at least one plug-in instance must be created per SMPP server to use. The plug-in instance binds to the SMPP server using a configurable ESME system ID.

The Native SMPP Service Facade must be provisioned with additional data about the application instance. This data includes:

- The port number to bind to
- The maximum number of concurrent sessions allowed
- Whether subsequent operations should be allowed to target a previously sent short message
- Whether network-triggered short messages and delivery reports should be forwarded to the application or not.
- The address range that when matched with the destination address of a network-triggered short message forwards the message to the application.

Application-triggered Short Messages

The communication service forwards the SMPP PDU sent from the application transparently to the SMPP server, if not modified by the Service Interceptors, with the exception of the BIND PDU. The BIND PDU is configured in the plug-in.

There is no enforcement on maximum number of unacknowledged SMPP operations, a maximum window size cannot be specified.

Network-triggered Short Messages

The communication service forwards the SMPP PDU sent from the SMPP server to Oracle Communications Services Gatekeeper transparently to the application, if not modified by the Service Interceptors, with the exception of the BIND PDU.

If the application does not acknowledge the reception of a network-triggered message, an error response is sent back to the SMPP SMSC. Both the error message and the time to wait for a response are configurable in the Native SMPP Facade.

Supported Network Protocols

The native SMPP communication service supports the following network protocol:

- [SMPP v3.4](#)

SMPP v3.4

Oracle Communications Services Gatekeeper connects to an SMSC using SMPP v3.4. See [“Appendix A: Standards and Specifications”](#) in *Concepts and Architectural Overview* for the exact version of the standard Oracle Communications Services Gatekeeper supports.

Short Code Translation

The Native SMPP communication service does not offer short code translations.

Limitations and Recommendations

Below is a set of limitations to and recommendations for the Native SMPP Communication Service:

- Windowing, as implemented by the Parlay 2.1 Short Messaging/SMPP plug-in, is not supported.
- The following operations are not supported:
 - data_sm
 - alert_notification
 - outbind
- Hitless upgrade is not supported. The old version of the communication service must be undeployed before a new version can be deployed.

- Different applications should not share the same bind towards the SMPP server. See [Load Balancing, High Availability and Fail-Over](#).

Load Balancing, High Availability and Fail-Over

To optimize system utilization, applications should load balance application-triggered requests between all Network Tier servers.

The SMPP server should load-balance network-triggered requests between all Network Tier servers.

Load balancing is only supported between plug-in instances which are located in same Network Tier server and share the same large account. When a request has reached a plug-in instance in a Network Tier server, it uses the Service Facade in the same server to pass the request on to the applications. When a request has reached the Service Facade in a Network Tier server, it uses a plug-in instance in the same Network Tier server to process the request.

High availability and fail-over is supported out-of-the-box between Oracle Communications Services Gatekeeper and the SMPP server. High availability between the application and Oracle Communications Services Gatekeeper must be handled by each application.

A prerequisite for high-availability for the Native SMMP communication service is redundant Network Tier servers, redundant network interface cards in each Network Tier server, and a redundant set of SMPP servers to connect to. By using at least two different plug-in instances per Network Tier server, and having the plug-in instances connect to different SMPP servers high-availability is achieved from Network Gatekeeper to the network.

Between SMPP applications and Network Gatekeeper, the applications need to handle high availability and fail-over for application-initiated requests by binding to two or more Network Tier servers. For network-triggered requests, the same requirement that the applications bind to two or more network tier servers applies.

The Native SMPP communication service can be provisioned so applications share the same large account in the SMPP server, so that they share the same bind. However, when there is only one bind between Oracle Communications Services Gatekeeper and the SMPP server, and more than one application listens to network-triggered messages, the Native SMPP communication service must listen to incoming messages on behalf of all the applications. The bind between the plug-in and the network node is done on all Network Tier servers, so network-triggered messages can be sent to any of these servers. If the network-triggered request ends up in a Network Tier server that the application has not bound to, the message is put into a JMS queue for the server that has the active bind. The latency in this case can have a negative impact on performance. For better performance all applications should bind to all Network Tier servers.

Configuration Specifics for the Native SMPP Communication Service

Communication services share many common features, covered in the [“Introducing Communication Services”](#) chapter of *Concepts and Architectural Overview*, but each one has a few characteristics that are specific only to that service. This section describes those specific features for the native SMPP communication service, including:

- [Charging Data Records](#)
- [Event Data Records](#)
- [Statistics](#)
- [Supported Address Schemes](#)

Charging Data Records

Generation of CDRs

The Native SMPP communication service writes CDRs in the following conditions:

- After a short message has been successfully sent from the application to the network
- After a short message has been successfully sent from the network to the application
- After a delivery report has been successfully delivered to the application

Event Data Records

The Native SMPP communication service uses the standard EDR mechanism, documented in [Table 16-1](#). For more information, see [Appendix A, “Events, Alarms, and Charging.”](#)

Table 16-1 Event types emitted in Native SMPP Communication Service

EdrId	Meaning
400000	An application-initiated bind operation has entered the plug-in.
400001	An application-initiated unbind operation has entered the plug-in.
400002	An application-initiated submit operation has entered the plug-in.

Table 16-1 Event types emitted in Native SMPP Communication Service

EdrId	Meaning
400003	A network-triggered submitResponse operation has been forwarded to the Service Facade.
400004	An application-initiated submitSmMulti operation has entered the plug-in.
400005	A network-triggered submitMultiResponse operation has been forwarded to the Service Facade.
400006	An application-initiated querySm operation has entered the plug-in.
400007	A network-triggered submitMultiResponse operation has been forwarded to the Service Facade.
400008	An application-initiated cancelSm operation has entered the plug-in.
400009	A network-triggered cancelResponse operation has been forwarded to the Service Facade.
400010	An application-initiated replaceSm operation has entered the plug-in.
400011	A network-triggered replaceResponse operation has been forwarded to the Service Facade.
400020	The plug-in initiates an attempt to bind to the SMPP server.
400021	The plug-in unbinds from the SMPP server
400022	An application-initiated submit operation has been forwarded to the SMPP server.
400023	A network-triggered receiveSubmitSmResponse operation has been received from the SMPP server.
400024	An application-initiated sendSubmitSmMulti operation has been forwarded to the SMPP server.
400025	A network-triggered receiveSubmitSmMultiResponse operation has been received from the SMPP server.
400026	An application-initiated sendQuerySm operation has been forwarded to the SMPP server.

Table 16-1 Event types emitted in Native SMPP Communication Service

EdrId	Meaning
400027	A network-triggered receiveQuerySmResponse operation has been received from the SMPP server.
400028	An application-initiated sendCancelSm operation has been forwarded to the SMPP server.
400029	A network-triggered receiveCancelSmResponse operation has been received from the SMPP server.
400030	An application-initiated sendReplaceSm operation has been forwarded to the SMPP server.
400031	A network-triggered receiveReplaceSmResponse operation has been received from the SMPP server.
400100	A network-triggered deliverSm operation has been forwarded to the SMPP server.
400101	An application-initiated deliverSmResponse operation has reached the plug-in.
400102	A network-triggered message that had been stored in the JMS queue has expired and been removed.
400103	A deliverSmResponse operation from the connector has reached the plug-in. The message has been transferred via JMS to a server with an active bind session to the application.
400110	A network-triggered short message has reached the plug-in.
400111	A network-triggered delivery receipt has reached the plug-in.
400112	A response (deliver_sm_resp) to a network-triggered short message or delivery receipt is sent to the SMPP server.

Statistics

The table below outlines the correlation between the methods being invoked from either the application (in application-initiated requests) or the telecom network (in network-initiated requests) and the transaction type collected by the statistics counters in Gatekeeper for the native SMPP communication service:

Note: Method names for network-initiated requests are specified by the internal Gatekeeper name, which is not necessarily the same as the message from the network.

Table 16-2 Transaction types for Native SMPP communication service

Method	Transaction type
submitSm	TRANSACTION_TYPE_MESSAGING_SEND
submitSmMulti	TRANSACTION_TYPE_MESSAGING_SEND
receiveMoReq	TRANSACTION_TYPE_MESSAGING_RECEIV

Supported Address Schemes

The native SMPP communication service supports the `tel:` address scheme.

Events, Alarms, and Charging

This appendix describes the features common to the handling of events, alarms, and charging in Oracle Communications Services Gatekeeper.

- [Events](#)
- [Alarms](#)
- [Charging Data Records](#)

Events

Events are handled differently in the Access Tier and the Network Tier.

Event handling in the Access Tier

The Access Tier runs in the WebLogic Server's Web Services Container, so events or alarms that are raised there can be monitored through standard JMX mechanisms or by using the WebLogic Diagnostics Framework.

See *[Developing Manageable Applications with JMX](#)* and *[Configuring and Using the WebLogic Diagnostics Framework](#)*, part of the Oracle WebLogic Server set of documents, for more information on how this works.

Event handling in the Network Tier

In the Network Tier, much of the functionality comes from the interaction between communication services and Oracle Communications Services Gatekeeper's Container Services. To capture this specialized level of information, and other pertinent information about the status of the tier, Oracle Communications Services Gatekeeper has developed specific mechanisms to record the data.

In standard communication services, all status information generated by the Network Tier - events, alarms, charging data, and usage statistics - begins as an event, which is fired whenever designated methods are called or exceptions are thrown. These events are then sent to the EDR Service. In the EDR Service, events are processed through XML-based filters, which provide the criteria by which the events are classified into types. The filters can also be used to transform the data in the original event, including adding other useful information. When the information has been processed by the filters, it is delivered to type-specific listeners. Out of the box, there are three types of filters that are all found in the file `wlng-edr.xml`. They produce three distinct types of data: *Event Data Records* (EDRs), *Charging Data Records* (CDRs), and *Alarms*. All three of these filters can be customized as desired, using the Administrative Console. These filters can also deliver desired event-based information to external JMS-based listeners. Such listeners are set up as standard JMS topic subscribers and can be anywhere on the network. See [Oracle Communications Services Gatekeeper - System Administrator's Guide](#) for more information on setting up these filters.

Note: For the purposes of backwards compatibility, events, alarms, and charging records can be published and delivered to 2.2 style, as well as standard style, listeners, but this mechanism has been deprecated since version 3.0.

Each EDR always includes the following data:

Element	Represents
ServiceName	The service type (SMS, Call Handling, etc.) that produced the event
ServerName	The name of the WLS host
Timestamp	The time at which the event was triggered (in milliseconds from midnight 1 January 1970)
ContainerTransactionID	The transaction ID from WebLogic Server, if available. This identifies the thread on which the request is executed

Class	The name of the class that logged the event
Method	The name of the method that logged the event
Source	<p>The kind of event. There are two possible values for this field:</p> <ul style="list-style-type: none"> • Method: the event was fired in relation to a method call • Exception: the event was fired in relation to an exception being thrown

In addition, most events include:

Element	Represents
Direction	<p>The direction in which the request is traveling. There are two possible values for this field:</p> <ul style="list-style-type: none"> • South: traveling toward the network node • North: traveling toward the application
Position	<p>The position of the EDR relative to the method that logged the EDR. There are two possible values for this field:</p> <ul style="list-style-type: none"> • Before: the event occurred before the method • After: the event occurred after the method
Interface	<p>The interface at which the EDR is logged. There are three possible values for this field:</p> <ul style="list-style-type: none"> • North: the event was logged at the north plug-in interface • South: the event was logged at the south plug-in interface • Other: the event was logged someplace other than the north or south interfaces
Exception	The name of the exception that triggered the EDR
SessionId	The application's session identifier
ServiceProviderId	The service provider account identifier

ApplicationId	The application account identifier
AppInstanceGroupId	The authentication user name of the Application Account. This is a string that is equivalent to the 2.2 value: Application Instance Group ID
OrigAddress	The originating address with scheme. For example: tel:12123334444
DestAddress	The destination address. If this is a send list, the first address will be listed here. Additional addresses are stored in the AdditionalInfo field.
AdditionalInfo	Variable information depending on the communication service. Stored as “key=value\n” pairs.
PluginID	The unique ID of the plug-in instance.

Alarms

Network Tier alarms are those events that are of immediate interest to the operator. They are EDRs that are defined via filters created in the internal configuration file. While each alarm begins as an EDR, not all the information available in the EDR is stored when the alarm is written to the database (although that information can be retrieved using an external listener). Each alarm entry in the database includes the following information:

Element	Represents
alarm_id	A unique sequential identifier
source	The name of the software module that raised the alarm and the IP address of the server in which the module runs. This is <i>not</i> the same as the Source field in the event
timestamp	The time at which the event was triggered (in milliseconds from midnight 1 January 1970)

severity	The importance of the alarm. There are four possible values for this field: <ul style="list-style-type: none"> • 4 for warning • 3 for minor • 2 for major • 1 for critical
identifier	The alarm type
alarm_info	Information provided by the module that raised the alarm
additional_info	This field includes: <ul style="list-style-type: none"> • Service Provider ID • Application ID • Application Instance ID • Plug-in instance ID • Other information depending on context

Management integration

Oracle Communications Services Gatekeeper supports integration of its alarm and event mechanisms with external management tools.

OSS

An Operation Support System (OSS) can integrate with Oracle Communications Services Gatekeeper alarm and event services through the creation of external JMS listeners. As well, integration can be managed by OAM scripts through the use of JMX-based tools.

SNMP

Oracle Communications Services Gatekeeper also supports the sending of alarms as SNMP traps to SNMP managers. The alarms sent to the SNMP managers can be filtered on alarm severity.

Charging Data Records

CDRs originate as filtered EDRs. While each CDR begins as an EDR, not all the information available in the EDR is stored when the CDR is written to the database (although that information

can be retrieved using an external listener). Each CDR entry in the database includes the following information:

Element	Represents
transaction_id	The Oracle Communications Services Gatekeeper transaction sequence number
service_name	The communication service whose use is being tracked
service_provider	The Service Provider ID
application_id	The Application ID
application_instance_id	The user name of the Application Account. This is a string that is equivalent to the 2.2 value: Application Instance Group ID
container_transaction_id	The transaction ID from WebLogic Server, if available. This identifies the thread on which the request is executed
server_name	The name of the server in which the CDR was generated
timestamp	The time at which the event was triggered (in milliseconds from midnight 1 January 1970)
service_correlation_ID	An identifier that allows the usage of multiple service types to be correlated into a single charging unit
charging_session_id	<p>An ID correlating related transactions within a service capability module that belong to one charging session. For example, a call containing three call legs will produce three separate transactions within the same session</p> <p>Note: In installations where sessions are not used, this field contains only a placeholder value.</p>
start_of_usage	The date and time the request began to use the services of the underlying network
connect_time	The date and time the destination party responded. Used for Call Control traffic only

end_of_usage	The date and time the request stopped using the services of the underlying network
duration_of_usage	The total time the request used the services of the underlying network
amount_of_usage	The used amount. Used when charging is not time dependent, as in, for example, flat rate services
originating_party	The originating party's address
destination_party	The destination party's address. This is the first address in the case of send lists, with all additional addresses placed in the additional_info field.
charging_info	A service code added by the application or by policy service
additional_info	If the communication service supports send lists, all destination addresses other than the first, under the key destinationParty. In addition any other information provided by the communication service

Events, Alarms, and Charging