Oracle® Communications Services Gatekeeper

Partner Relationship Management Guide

Release 5.0

E16618-02

April 2011

ORACLE®

Oracle Communications Services Gatekeeper Partner Relationship Management Guide, Release 5.0

E16618-02

Copyright © 2007, 2011, 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

Preface ix

Audience ix
Documentation Accessibility ix
Related Documents x

1 Introduction

Integration with External Systems 1-1
Security 1-1
Deployment Example 1-2

2 The Partner Management Model

Account States 2-2
PRM Users in the Context of Services Gatekeeper Administrative Users 2-3

3 The Interfaces

WSDLs 3-2
Service Provider Interfaces 3-2
Service Provider Service Interfaces 3-2
Management User 3-3
Service Provider Accounts 3-3
Application Account 3-3
Application Instances 3-3
Service Provider CDR Utility Interface 3-4
Service Provider Statistics Utility Interface 3-4
Service Provider Login Interface 3-4
Operator Interfaces 3-4
Operator Service Interfaces 3-4
Management User 3-4
Service Provider Account 3-5
Service Provider Group 3-5
Application Account 3-5
Application Account Group 3-6
Application Instance 3-6
Operator Alarm Utility Interface 3-7
Operator CDR Utility Interface 3-7

Operator Statistics Utility Interface 3-7
Operator Login Interface 3-7

4 Common PRM Use Scenarios

Registering a new Service Provider Account 4-1
Registering a new Application Account 4-1
Registering a new Application Instance 4-2
Operator: Creating a Service Provider Group 4-2
Operator: Creating an Application Account Group 4-3
Service Provider Requests an Account Update 4-3
Service Provider Deactivates an Account 4-3
Service Provider Requests an Account Deletion 4-4
Communicating General Information Between Service Provider and Operator 4-4
Retrieving Charging Data Records 4-4
Retrieving Statistics 4-4
Retrieving Alarms 4-4

5 Service Provider Login

Interface: SpLogin 5-1
login 5-1
Input Parameters 5-1
Return Parameters 5-2
Possible Exceptions 5-2
logout 5-2
Return Parameters 5-2
Possible Exceptions 5-2
registerSpAccountReq 5-2
Input Parameters 5-2
Return Parameters 5-3
Possible Exceptions 5-3
Exceptions 5-3
CommonException 5-3
Complex data types 5-3
SpAccount 5-3
Property 5-3

6 Service Provider Service

Interface: SpService 6-1
deleteSpAccountReq 6-1
Input Parameters 6-1
Return Parameters 6-1
Possible Exceptions 6-1
deactivateSpAccount 6-1
activateSpAccount 6-2
getSpAccount 6-2
getSpAccountState 6-2

registerAppAccountReq ... 6-3
deleteAppAccountReq ... 6-3
updateSpAccountReq ... 6-3
updateAppAccountReq ... 6-4
updateAppInstGroupReq ... 6-4
deleteAppInstGroupReq ... 6-5
listAppAccounts ... 6-5
getAppAccount ... 6-6
getAppAccountState ... 6-6
activateAppAccount ... 6-6
deactivateAppAccount ... 6-7
activateAppInstGroup ... 6-7
getAppInstGroupState ... 6-7
deactivateAppInstGroup ... 6-8
registerAppInstGroupReq ... 6-8
deactivateAppInstGroup ... 6-9
listAppInstGroups ... 6-9
getAppInstGroup ... 6-10
getSpAccountSla ... 6-10
getSpAccountSlaByType ... 6-11
getAppAccountSla ... 6-11
getAppAccountSlaByType ... 6-12
setAppInstGroupPassword ... 6-12
changeSpAccountPassword ... 6-13
Exceptions ... 6-13
ACCESS_DENIED ... 6-13
CommonException ... 6-13
INVALID_STATE ... 6-13
Data types ... 6-13
AppAccount ... 6-13
AppInstGroup ... 6-13
Property ... 6-14
SpAccount ... 6-14
State ... 6-14

7 Service Provider CDR Utility

Interface: SpCdrUtil ... 7-1
countCdrs ... 7-1
listCdrs ... 7-2
Exceptions ... 7-3
CommonException ... 7-3
Data types ... 7-3
CdrInfo ... 7-3
CdrCompletionStatus ... 7-4
Property ... 7-4

8 Service Provider Statistics Utility

Interface: SpService 8-1
listStatisticTypes 8-1
getStatistics 8-1
Exceptions 8-2
CommonException 8-2
Data types 8-2
StatisticsInfo 8-2
StatisticTypeDescriptor 8-3

9 Operator Login

Interface: OpLogin 9-1
login (Deprecated. For backwards-compatibility only) 9-1
logout (Deprecated. For backwards-compatibility only) 9-1
Exceptions 9-2
ACCESS_DENIED 9-2
CommonException 9-2
Complex data types 9-2

10 Operator Service

Interface: OpService 10-1
listAppGroups 10-1
getAppGroup 10-1
createAppGroup 10-2
createAppGroupByType 10-2
deleteAppGroup 10-3
moveAppAccountToGroup 10-3
getAppGroupId 10-3
updateAppGroup 10-4
updateAppGroupByType 10-4
listSpGroups 10-5
getSpGroup 10-5
createSpGroup 10-6
createSpGroupByType 10-6
deleteSpGroup 10-7
moveSpToGroup 10-7
getSpGroupId 10-8
updateSpGroup 10-8
updateSpGroupByType 10-9
listAppInstGroups 10-10
getAppInstGroup 10-10
getAppInstGroupState 10-11
registerAppInstGroupReq 10-11
deleteAppInstGroupReq 10-12
deleteAppInstGroupRes 10-12
updateAppInstGroup 10-13

updateAppInstGroupRes ... 10-14
getUpdatePendingAppInstGroup ... 10-14
setAppInstGroupPassword ... 10-15
unlockAppInstGroup ... 10-15
activateSpAccount ... 10-16
deactivateSpAccount ... 10-16
getSpAccount ... 10-17
getSpAccountState ... 10-17
registerAppAccountReq ... 10-17
registerAppAccountRes ... 10-18
updateAppAccountRes ... 10-19
getUpdatePendingAppAccount ... 10-19
deleteAppAccountReq ... 10-20
deleteAppAccountRes ... 10-20
updateAppAccount ... 10-21
getAppAccount ... 10-21
getAppAccountState ... 10-22
activateAppAccount ... 10-22
deactivateAppAccount ... 10-23
registerAppInstGroupRes ... 10-23
activateAppInstGroup ... 10-24
deactivateAppInstGroup ... 10-24
registerSpAccountReq ... 10-25
listAppAccounts ... 10-26
listSpAccounts ... 10-26
registerSpAccountRes ... 10-26
deleteSpAccountReq ... 10-27
deleteSpAccountRes ... 10-27
updateSpAccount ... 10-28
updateSpAccountRes ... 10-28
getUpdatePendingSpAccount ... 10-29
setSpAccountPassword ... 10-29
changeOpAccountPassword ... 10-30
getUserLevel ... 10-30
Exceptions ... 10-30
- ACCESS_DENIED ... 10-31
- CommonException ... 10-31

Data types ... 10-31
- AppAccount ... 10-31
- AppAccountRef ... 10-31
- AppInstGroupRef ... 10-31
- AppGroup ... 10-31
- AppInstGroup ... 10-32
- Property ... 10-32
- RequestResponse ... 10-32
- SpAccount ... 10-32
- SpGroup ... 10-33

UserLevel 10-33
State 10-33

11 Operator CDR Utility

Interface: OpCdrUtil 11-1
countCdrs 11-1
listCdrs 11-2
Exceptions 11-3
CommonException 11-3
Data types 11-3
CdrInfo 11-3
CdrCompletionStatus 11-4
Property 11-4

12 Operator Statistics Utility

Interface: OpStatisticsUtil 12-1
listStatisticTypes 12-1
getStatistics 12-1
Exceptions 12-2
CommonException 12-2
Data types 12-2
StatisticsInfo 12-2
StatisticTypeDescriptor 12-3

13 Operator Alarm Utility

Interface: OpAlarmUtil 13-1
countAlarms 13-1
listAlarms 13-1
Exceptions 13-2
CommonException 13-2
Data types 13-2
AlarmInfo 13-2
AlarmSeverity 13-3

Preface

This document describes the Oracle Communications Service Gatekeeper's Partner Relationship Management module. It includes a detailed description of the Partner Relationship Management interfaces exposed for integrating Oracle Communications Services Gatekeeper service provider account management with operator PRM systems, both internal and customer-facing.

Audience

This book is intended mainly for application developers who are interested in building CRM/PRM applications to manage Service Providers and Applications that are or will use the traffic interfaces exposed by the Oracle Communications Services Gatekeeper. It assumes a familiarity with Oracle Communications Services Gatekeeper.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit

http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Communications Services Gatekeeper set:

- *Accounts and SLAs Guide*
- *Alarm Handling Guide*
- *Application Developer's Guide*
- *Concepts Guide*
- *Communication Service Guide*
- *Installation Guide*
- *Licensing Guide*
- *Platform Development Studio Developer's Guide*
- *Platform Test Environment Guide*
- *RESTful Application Development Guide*
- *SDK User's Guide*
- *Statement of Compliance*
- *System Administrator's Guide*
- *System Backup and Restore Guide*

1 Introduction

Oracle Communications Services Gatekeeper's Partner Relationship Management module gives operators a set of Web Services interfaces that can be used to incorporate the Service Provider and Application management parts of the Oracle Communications Services Gatekeeper into their CRM/PRM systems, intranets and extranets.

Integration with External Systems

Administering service provider and application accounts for Oracle Communications Services Gatekeeper can be a work-intensive task for operators. Using CRM/PRM applications that are built using the Partner Relationship module can allow operators to shift some of that work to the service providers themselves, as well as giving those service provides a defined and structured channel both to communicate any changes they desire and to monitor their own usage statistics. The operator's task is reduced to simply approving the pre-entered changes, dramatically reducing administration overhead. Using simple Web Service calls, the integrated PRM application can manage a wide range of Oracle Communications Services Gatekeeper service provider account services.

The PRM interfaces support Oracle Communications Services Gatekeeper's service provider and application administration model, which is described more fully in Chapter 2, "The Partner Management Model."

Note: Users, both operator-based and service-provider-based, who have been given appropriate permissions can also interact with Oracle Communications Services Gatekeeper management systems using JMX-based solutions. For information on the MBeans available for use and the attributes and operations they expose, please see *Accounts and SLAs Guide*.

Security

The PRM Web Services module uses WS-Security to ensure the security of the Web Services based interaction between the PRM and its CRM/PRM application clients. Each request is authenticated using a username token or X.509 certificate that is included in the SOAP header. For more information on how this works, see the "Interacting with Oracle Communications Services Gatekeeper" chapter in the *Application Developer's Guide*, another document in this set. Although the context in that guide is the SOAP headers of traffic requests, the mechanism described is identical to the one used in with the PRM module.

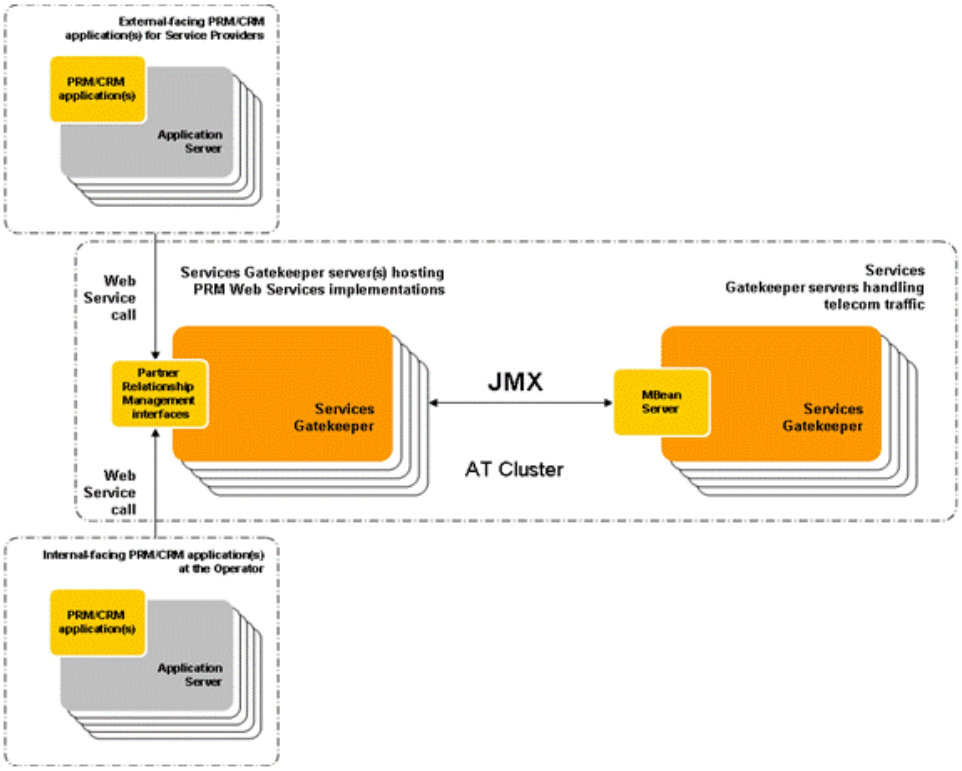
Note: For backwards compatibility purposes, a session ID-based login mode is also supported.

Deployment Example

The Partner Relationship Management module consists of two parts:

- The Web Service interfaces that are used by the CRM/PRM application
- The implementation of these Web Services

Figure 1–1 Deployment example



In the Figure 1–1, "Deployment example", there are two CRM/PRM applications, one supporting the Service Provider set of interfaces and the other supporting the more comprehensive Operator set of interfaces. Each of these applications uses Web Service calls to communicate with the host(s) running the PRM Web Services. The PRM Web Services module is deployed by default on the AT Tier cluster. The PRM server(s) in turn use JMX to communicate with the Oracle Communications Services Gatekeeper servers that actually handles the telecom traffic.

2

The Partner Management Model

The management of Oracle Communications Services Gatekeeper is structured around the concept of a Management Model. The Management Model defines roles for the operator and the application service provider and describes the interaction between them. In the context of Oracle Communications Services Gatekeeper, an operator is the entity which runs the network in which Oracle Communications Services Gatekeeper is installed. The operator has partners who want one or more of their applications to interact with the operator's network. These partners are the application service providers and they can be in-house or external to the operator.

The Oracle Communications Services Gatekeeper Partner Management interfaces allow operators to manage these application service providers at increasingly granular levels of control. An application service provider registers with Oracle Communications Services Gatekeeper and is given a Service Provider Account. To support tiering, service provider accounts are associated together into Service Provider Account Groups. It is these groups that are associated with Service Level Agreements or SLAs.

Within a service provider account there are individual Application Accounts, registered on their respective service provider accounts. As in the case of service provider accounts, these application accounts are grouped together into Application Account Groups. Again, SLAs are associated with applications via the application group.

Finally, the model also includes the idea of the Application Instance, which is tied to a specific instance of the application and is used in the traffic authentication process.

For more information about the Partner Management model, see the *Accounts and SLAs Guide*.

The Partnership Management module allows for management of:

- Service Provider accounts
- Application Accounts
- Application Groups
- Service Provider SLAs:
  - Provisioned and enforced in one cluster
  - Provisioned and enforced across clusters (used for establishing geo-redundancy)
- Service Provider Node SLAs
- Application SLAs
  - Provisioned and enforced in one cluster

- Provisioned and enforced across clusters (used for establishing geo-redundancy)

Account States

All Service Provider Accounts and Application Accounts are in one of the following states:

- Registered

  The service provider has requested that an account be registered, but the operator has not yet approved or disapproved it.

- Active

  The operator has approved the account the service provider registered.

- Inactive

  The account has been deactivated, either temporarily or as a step toward being deleted.

- Update pending

  The service provider has requested an update of the account, and this update has not yet been approved by the operator.

- Delete pending

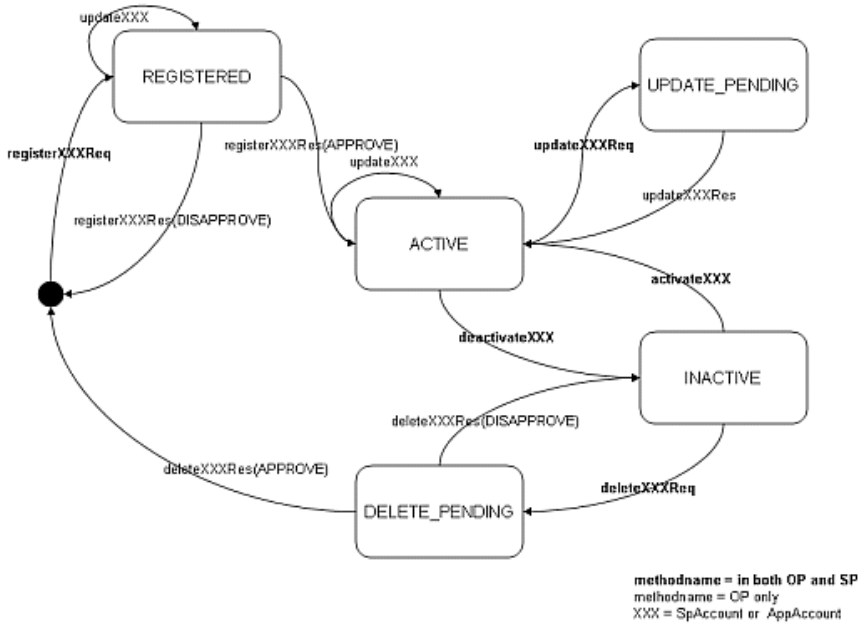
  The service provider or the operator has requested that the account be deleted. This is an intermediate state. The operator can, for example, use this state to process all charging data records for the account before deleting it.

  Note: Charging data records may still be in the Oracle Communications Services Gatekeeper, even when the account information is deleted. Make sure all data has been processed before deleting an account.

  Once an account is deleted, all data about the account is removed from the Oracle Communications Services Gatekeeper.

The possible state transitions are outlined in Figure 2–1.

Figure 2–1 States and state transitions



There are two sets of interfaces in the PRM module. The Service Provider interfaces give application service providers access to information relative to their own accounts and applications. The Operator interfaces allow operators to manage their service providers. These include access to a much broader range of management functions.

In the diagram above, the method names in **bold** can be executed by both the operator and the application service provider. The methods names in non-bold can be executed only by the operator. XXX indicates that the methods are valid for both service provider accounts and application accounts.

PRM Users in the Context of Services Gatekeeper Administrative Users

PRM users are a subsection of all Oracle Communications Services Gatekeeper Administrative Users. They are created and managed in the same way as all other Oracle Communications Services Gatekeeper Administrative Users, using the Management Users MBean. For more information on managing Administrative Users, see the *System Administrator's Guide*. At least one PRM-OP user must be set up before the PRM interfaces can be used.

Table 2–1 describes the characteristics of all Administrative Users stored in the Oracle Communications Services Gatekeeper database:

Table 2–1 Contents of wlng_mgmt_users Database Table

Field	Type	Description
username	varchar(255)	Authentication name of the administrative user.

Table 2–1 (Cont.) Contents of wlng_mgmt_users Database Table

Field	Type	Description
state	int(11)	Possible values are: ■ 0: Activated ■ 1: Deactivated Transitional states (Registered, Update pending, Delete pending) are stored temporarily as properties of the account.
type	int(11)	Kind of user. Options are: ■ 0: OAM (Console-based user) ■ 1: PRM-OP (Operator using PRM) ■ 2: PRM-SP (Service Provider using PRM)
password	varchar(255)	Administrative user password. 3DES Encrypted.
userlevel	int(11)	Privilege level of user. See Table 2–2 below for values.
groupname	varchar(255)	Allows administrative us.ers to be grouped for ease of management
stored_ts	bigint(20)	Tablespace

Table 2–2 Privilege Levels

Level	Oracle Communications Services Gatekeeper Role Type
1000	Equivalent to Administrative Access on WLS. Can: ■ Manage servers and server configuration ■ Deploy applications ■ Control all Oracle Communications Services Gatekeeper management functions
666	Equivalent to Deployer Access on WLS. Can: ■ View server configuration and make some changes ■ Have read-write access on Oracle Communications Services Gatekeeper management functions
333	Equivalent to Monitor Access on WLS. Can ■ View server configuration ■ Have read-only access to Oracle Communications Services Gatekeeper management functions
0	Equivalent to Anonymous Access on WLS. Can: ■ Use servers. PRM-SP users have this privilege level

Note: Service providers may also have direct access to account management functions via JMX if the service provider has appropriate user permissions. This is a decision made by the operator.

3

The Interfaces

The Partner Relationship Management module consists of two sets of interfaces:

- The Service Provider interface set
- The Operator interface set

The Service Provider interface set contains a subset of the functionality of the Operator set. It can be used to perform operations for only one specific Service Provider Account; the one with which the user of the CRM/PRM application has authenticated. The Operator set can perform operations on all Service Provider Accounts and also has access to alarm reports. All operations are synchronous.

The Service Provider set of interfaces consists of the following groups:

- Service Provider Service Interface, see "Service Provider Service Interfaces".
- Service Provider CDR Utility interface, see "Service Provider CDR Utility Interface".
- Operator Statistics Utility Interfaces, see "Service Provider Statistics Utility Interface".
- Service Provider Login interface, see "Service Provider Login Interface".

> **Note:** The login portion of this interface is provided only for backwards compatibility purposes. The current implementation is sessionless and the only use for this interface in the current implementation is to request a new Service Provider Account be set up.

The Operator set of interface consists of the following groups:

- Operator Service Interface, see "Operator Interfaces".
- Operator Alarm interface, see "Operator Alarm Utility Interface"
- Operator CDR Utility interface, see "Operator CDR Utility Interface".
- Operator Statistics Utility Interfaces, see "Operator Statistics Utility Interface"
- Operator Login interface, see "Operator Login Interface".

> **Note:** This interface is only supplied for backwards compatibility. The current implementation is sessionless.

WSDLs

The following is a list of the Web Services interfaces available for integration with CRM/PRM applications. The interfaces use document/literal encoding and assume SOAP over HTTP.

The WSDL files that correspond to the interface sets can be found at the URIs listed below:

- SpService, WSDL at http://<host>:<port>/prm_sp/services/SpService?wsdl, see "Service Provider Interfaces".
- SpCdrUtil, WSDL at http://<host>:<port>/prm_sp/services/SpCdrUtil?wsdl, see "Service Provider CDR Utility Interface".
- SpStatUtil, WSDL at http://<host>:<port>/prm_sp/services/SpStatisticsUtil?wsdl, see "Service Provider Statistics Utility Interface".
- SpLogin, WSDL at http://<host>:<port>/prm_sp/services/SpLogin?wsdl, see "Service Provider Login Interface".

  Note: The login portion of this interface is provided only for backwards compatibility purposes. The current implementation is sessionless and the only use for this interface in the current implementation is to request a new Service Provider Account be set up.

- OpService, WSDL at http://<host>:<port>/prm_op/services/OpService?wsdl, see "Operator Interfaces".
- OpAlarmUtil, WSDL at http://<host>:<port>/prm_op/services/OpAlarmUtil?wsdl, see "Operator Alarm Utility Interface".
- OpCdrUtil, WSDL at http://<host>:<port>/prm_op/services/OpCdrUtil?wsdl, see "Operator CDR Utility Interface".
- OpStatUtil, WSDL at http://<host>:<port>/prm_op/services/OpStatisticsUtil?wsdl, see "Operator Statistics Utility Interface".
- OpLogin,WSDL at http://<host>:<port>/prm_op/services/OpLogin?wsdl, see "Operator Login Interface".

  Note: This interface is only supplied for backwards compatibility. The current implementation is sessionless.

Service Provider Interfaces

Service Provider interfaces allow application service providers to request changes in their account and to monitor their accounts activities

Service Provider Service Interfaces

The Service Provider Service interface provides ways to interact with the following entities:

- Management User
- Service Provider Accounts

- Application Account
- Application Instances

Management User

The management user account is the account by which the service provider is authenticated with the PRM system. The following operation can be performed on an ongoing management user account:

- Change the password for the Service Provider Account

Service Provider Accounts

The following operations can be performed on Service Provider Accounts:

- Activate
- Deactivate
- Request deletion
- Request update
- Get information
- Get SLA
- Get state

Application Account

The following operations can be performed on an Application Account:

- Register new
- Activate
- Deactivate
- Request deletion
- Update
- Get information
- List current Application Accounts
- Get the SLA
- Get the state

Application Instances

The following operations can be performed on an Application Instance:

- Register new
- Activate
- Deactivate
- Request deletion
- Request update
- Get information
- Get the state

- List current Application Instances
- Set the password that the Application Instance uses when accessing Oracle Communications Services Gatekeeper.

Service Provider CDR Utility Interface

The following operations are available:

- Count the number of CDRs that have been generated
- List the CDRs that have been generated

Service Provider Statistics Utility Interface

The following operations are available:

- Retrieve generated statistics information.
- List available statistics types.

Service Provider Login Interface

- Backwards compatibility only:
  - Login
  - Logout
- Request a new Service Provider Account

Operator Interfaces

Operator interfaces allow operators to approve changes in Service Providers' accounts and to perform other account maintenance tasks.

Operator Service Interfaces

The Operator Service interface provides ways to interact with the following entities:

- Management User
- Service Provider Account
- Service Provider Group
- Application Account
- Application Account Group
- Application Instance

Management User

The following operations can be performed on a management user account:

- Change password for the Operator Account
- Get user level for an Operator Account
- Change password that the Service Provider uses with PRM-SP

Service Provider Account

The following operations can be performed on a Service Provider Account:

- Approve or disapprove service provider's request to register a new Service Provider Account
- Approve or disapprove service provider's request to delete an existing Service Provider Account
- Approve or disapprove service provider's request to update an existing Service Provider Account
- Activate
- Deactivate
- Connect (move) an account into a Service Provider Group
- Make an operator request to delete an existing Service Provider Account
- Make an operator request to update an existing Service Provider Account
- Make an operator request to register a new Service Provider Account
- Get a list of accounts that have pending requests, and need approval or disapproval by the operator
- Get Service Provider Account information
- List all Service Provider Accounts
- Get Service Provider Account state

Service Provider Group

The following operations can be performed on a Service Provider Group:

- Create
- Update
- Delete
- Get information about the group
- List all groups
- Get the Service Provider Group to which a specific Application Account is assigned

Application Account

The following operations can be performed on an Application Account:

- Approve or disapprove service provider's request to register a new Application Account
- Approve or disapprove service provider's request to update an existing Application Account
- Approve or disapprove service provider's request to delete an existing Application Account
- Get a list of accounts that have pending requests, and need approval or disapproval by the operator
- Activate

- Deactivate
- Connect (move) the account into an Application Account Group
- Get information about the account
- Get the account's state
- List all accounts
- Make an operator request to delete an existing account
- Make an operator request to register a new account
- Make an operator request to update an existing account

Application Account Group

The following operations can be performed on an Application Account group:

- Create
- Update
- Delete
- Get the Application Account Group to which a specific Service Provider Account and Application Account are assigned
- Get information about the group
- List all groups

Application Instance

The following operations can be performed on an Application Instance:

- Approve or disapprove a service provider's request to delete an existing Application Instance
- Approve or disapprove a service provider's request to register a new Application Instance
- Approve or disapprove a service provider's request to update an existing Application Instance
- Activate
- Deactivate
- Get a list of instances that have pending requests, and need approval or disapproval by the operator
- Get information about the instance
- Get Application Instance's state
- List all instances
- Set password to use when an application authenticates to Oracle Communications Services Gatekeeper
- Make an operator request to delete an existing instance
- Make an operator request to register new Application Instance
- Make an operator request to update an existing instance
- Get the authenticated operator's user level

Operator Alarm Utility Interface

The following operations are available:

- Count the number of alarms that have been generated
- List the alarms that have been generated

Operator CDR Utility Interface

The following operations are available:

- Count the number of charging data records that have been generated.
- List the charging data records that have been generated.

Operator Statistics Utility Interface

The following operations are available:

- Retrieve generated statistics information.
- List available statistics types.

Operator Login Interface

The operator login interface is provided only for backwards compatibility purposes.

4

Common PRM Use Scenarios

This section outlines common use patterns using PRM.

Registering a new Service Provider Account

1. The Service Provider applies for a Service Provider Account using Service Provider Login::registerSPAccountReq(...) providing basic information such as desired account name, contact details, and so forth as part of the application.
2. The Operator lists new requests using Operator::listSpAccounts(...). The list can be filtered to display only service provider accounts in the REGISTERED state.
3. The request is inspected by the Operator, and can be changed using Operator::updateSpAccount(...).
4. If the request is approved by the Operator, the Service Provider Account is first associated with a Service Provider Group and SLA and then approved using Operator::registerSpAccountRes(...). For information on setting up a Service Provider Group, see "Operator: Creating a Service Provider Group".

   Note: The Service Provider Group that the Service Provider Account belongs to can be changed at any time using Operator::moveSpToGroup(...).

5. Finally, the Service Provider is notified that the request has been approved via any of the contact channels that were detailed when the request was submitted. The CRM/PRM application implementor must set up a mechanism for this communication, or for the communication of any other account changes that may occur.

Registering a new Application Account

1. The Service Provider applies for an application account via Service Provider::registerAppAccountReq(...).
2. The Operator lists new request using Operator::listAppAccounts(...) The list can be filtered to display only REGISTERED applications belonging to a specific service provider account.
3. The request is inspected by the Operator, and can be changed using Operator::updateAppAccount(...).
4. If the request is approved by the Operator, the Application Account is first associated with an Application Account Group and SLA and then approved using

Operator::registerSpAccountRes(...). For information on setting up an Application Account Group, see "Operator: Creating an Application Account Group".

Note: The Application Account Group that the Application Account belongs to can be changed at any time using Operator::moveAppAccountToGroup(...).

---

5. Finally, the Service Provider is notified that the request has been approved via any of the contact channels that were supplied with the original request for a Service Provider Account. The CRM/PRM application implementor must set up a mechanism for this communication, or for the communication of any other account changes that may occur.

At this point the Application Account is in ACTIVE state. To actually begin sending traffic to Oracle Communications Services Gatekeeper, however, the application must now apply for an Application Instance ID to be used to authenticate the account. For information on Application Instances, see "Application Instances"; for information on how to register an Application Instances, see "Registering a new Application Instance".

Registering a new Application Instance

1. The Service Provider applies for an Application Instance via Service Provider::registerAppInstGroupReq(...). . The request include desired properties to be associated with the instance.
2. The Operator lists new requests using Operator::listAppInstGroups(...) The list can be filtered to display only requests from specific SP Accounts and App Accounts for instances in the REGISTERED state.
3. The request is inspected by the Operator, and can be changed using Operator::updateAppInstGroup(...).
4. If the request is approved by the Operator, it is approved using Operator::registerAppInstGroupRes(...).
5. Finally, the Service Provider is notified that the request has been approved via any of the contact channels that was detailed when applying for a Service provider Account. The CRM/PRM application implementor must set up a mechanism for this communication, or for the communication of any other account changes that may occur.

Once the Service Provider has a Service Provider Account, an Application Account, and an Application Instance ID, traffic can be sent to Oracle Communications Services Gatekeeper.

Operator: Creating a Service Provider Group

1. The Operator defines the use privileges for a particular Service Provider Group.
2. These use privileges are formalized in a Service Provider Service Level Agreement XML file.
3. The Operator creates the Service Provider Group using Operator::createSpGroupByType(...) and assigns the newly created Service Provider SLA to it. An ID for the group is also defined.

Operator: Creating an Application Account Group

1. The Operator defines the use privileges for a particular Application Account Group.
2. These use privileges are formalized in an Application Service Level Agreement XML file.
3. The Operator creates the Application Account Group using Operator::createAppGroupByType(...) and assigns the newly created Application SLA to it. An ID for the group is also defined.

Service Provider Requests an Account Update

The Service Provider can request an update to any of its account entities. The request might cover SLA data or user defined properties. The Operator is responsible for approving or disapproving the request.

1. To request an update for a Service Provider account, the Service Provider uses Service Provider::update<SpAccount | AppAccount | AccountAppInstGroup>Req(...) as appropriate.
2. Once the Service Provider has requested an update, the state of the account is changed to UPDATE_PENDING until the Operator has inspected the update request and either approved it or disapproved it using Operator::update<SpAccount | AppAccount | AccountAppInstGroup>Res(...).

> **Note:** When an account is in the UPDATE_PENDING state, no further requests to update the account are allowed until the initial request has been approved or disapproved.

Service Provider Deactivates an Account

The Service Provider can deactivate any of its account entities. The deactivation takes affect immediately. No traffic is allowed through an account that is deactivated. The impact of a deactivation is cascaded through the Service Providers system:

- When a Service Provider Account is deactivated, none of the applications run by the service provider are able to send traffic through Oracle Communications Services Gatekeeper.
- When an Application Account is deactivated, none of the applications that are associated with that Application Account are able to send traffic through Oracle Communications Services Gatekeeper.
- When an Application Instance is deactivated, only that Application Instance is unable to send traffic through Oracle Communications Services Gatekeeper. Other applications are not affected.

An account must always be deactivated before it can be deleted.

1. To deactivate an account the Service Provider can use either Service Provider::deactivate<SpAccount | AppAccount | AccountAppInstGroup>Req(...), depending on the type of account.
2. To deactivate an account the Operator can use either Operator::deactivate<SpAccount | AppAccount | AccountAppInstGroup>Req(...), depending on the type of account.

Service Provider Requests an Account Deletion

The Service Provider can request to have any of its account entities deleted. The deletion does not take effect until after the request has been approved by the Operator.

An account must always be in state INACTIVE before it can be deleted.

1. To request to delete an account, either Service Provider::delete<SpAccount | AppAccount | AppInstGroup>Req(...) is used.
2. When the Service Provider has requested an account deletion, the state of the account is changed to DELETE_PENDING until the Operator has inspected the request and either approved it or disapproved it using Operator::delete<SpAccount | AppAccount | AppInstGroup>Res(...).

Note: When the request to delete an account is approved the account is deleted from the database. It is the Operator's responsibility to make sure that outstanding charging data records are processed before the deletion takes place.

If the request to delete the account is disapproved, the state of the account becomes INACTIVE.

Communicating General Information Between Service Provider and Operator

The Service Provider can communicate desired updates to the Operator using the update methods, as described in "Service Provider Requests an Account Update". Each update request can contain a set of properties in the form of name-value pairs, which are defined by the implementors of the CRM/PRM application.

Retrieving Charging Data Records

Both the Service Provider and the Operator can retrieve Charging Data Records using <Operator | Service Provider>::listCdrs(...). The operator can retrieve CDRs for all Service Providers, while the Service Provider only has access to Charging Data Records generated by its own applications. Results can be filtered.

Retrieving Statistics

Both the Service Provider and the Operator can retrieve statistics using <Operator | Service Provider>::getStatistics(...). The operator can retrieve statistics for all Service Providers, while the Service Provider only has access to its own applications. A set of filters can be used, including Application Account IDs and time intervals.

Retrieving Alarms

The Operator can retrieve alarms using Operator::listAlarms(...). The operator can retrieve alarms generated by all Service Providers, as well as platform related alarms. A set of filters can be used, including timestamps and severity levels.

5

Service Provider Login

Oracle Communications Services Gatekeeper supports the following functionality in the Service Provider Login Web Service interface:

- login/logout capabilities for backwards compatibility only
- a way for prospective service providers to request a Service Provider Account.

Interface: SpLogin

The endpoint for this interface is:

`http://host:port/prm_sp/services/SpLogin`

where the value of *host* and *port* depend on the Oracle Communications Services Gatekeeper deployment.

login

Note that the **login** method is deprecated. It is described here for backwards compatibility only.

When supported, **login** was used by an application to log in. This method returns a loginTicket that serves as an identifier for the session in the Service Provider interface.

The syntax for **login**:

```
login(spAccountId,password)
```

Input Parameters

Table 5–1 describes the input parameters:

Table 5–1 Input Parameters for Login

Parameter Name	Type	Description
spAccountId	xsd:string	The ID of Service provider Account. See "registerSpAccountReq" for information on how to request a Service Provider Account.
password	xsd:string	The password associated with the Service Provider login account.

Return Parameters

The **login** method returns a **loginTicket** which is then used to identify the login session. This ticket must be supplied in the SOAP header for each method invocation. The **loginTicket** element is a **xsd:string** type.

Possible Exceptions

The possible exceptions are:

- **ACCESS_DENIED**
- **CommonException**

logout

Note that **logout** method is deprecated. It is described here for backwards compatibility only.

When supported, **logout** was used by an application to log out. This method destroys the login session and the corresponding **loginTicket** .

The syntax for **logout**:

```
logout(loginTicket)
```

where, *loginTicket* is the identifier retrieved at the time of logging in.

Return Parameters

None

Possible Exceptions

The possible exceptions are:

- **ACCESS_DENIED**
- **CommonException**

registerSpAccountReq

The **registerSpAccountReq** method is used to request a new Service Provider Account. When this request has been approved by the operator, see "registerSpAccountRes", the Service Provider can log in to the Partner Relationship Management module. For more information on theapproval by the operator, see "registerSpAccountRes".

The syntax for **registerSpAccountReq**:

```
registerSpAccountReq(spAccountId, spAccount, password)
```

Input Parameters

Table 5–2 describes the input parameters:

Table 5–2 Input Parameters for registerSpAccountReq

Parameter Name	Type	Description
spAccountId	xsd:string	The desired ID the Service Provider Account.

Table 5–2 (Cont.) Input Parameters for registerSpAccountReq

Parameter Name	Type	Description
spAccount	tns1:SpAccount	Data structure with details on the Service Provider account. See "SpAccount".
password	xsd:string	The password associated with the Service Provider login account.

Return Parameters

Void

Possible Exceptions

The possible exceptions are:

- **ACCESS_DENIED**
- **CommonException**

Exceptions

The Service Provider Login Web Service interface throws the following exceptions.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Complex data types

The Service Provider Login Web Service interface uses the following complex datatypes.

SpAccount

Table 5–3 provides a description of a Service Provider Account, including contact details.

Table 5–3 Contents of an SpAccount

Element name	Datatype	Description
Name	xsd:string	Name of the Service Provider.
Address	xsd:string	Address of the Service Provider.
EMailAddress	xsd:string	E-mail address of the Service Provider.
ContactPerson	xsd:string	Contact person at the Service provider.
PhoneNumber	xsd:string	Phone number to the Service Provider.
Properties	mpl:ArrayOf_tns1_Property	CRM/PRM application-defined name value pairs. See "Property".

Property

This is an array of name-value pairs. This datatype is used in several other datatypes specific for this interface. The properties are accessible from the Service Provider interface and the Operator interface, so they can be used for communicating information between the Service Provider and the Operator.

Table 5–4 Property

Element name	Datatype	Description
Name	xsd:string	Name of the property, with the value defined in Value. Unique with the array.
Value	xsd:string	The data associated with Name.

6

Service Provider Service

The Service Provider Service Web Service provides the Service Provider with operations for handling Service Provider Accounts, Application Accounts, and Applications Instances

Interface: SpService

The endpoint for this interface is:

`http://host:port/prm_sp/services/SpService`

where the value of *host* and *port* depend on the Oracle Communications Services Gatekeeper deployment.

deleteSpAccountReq

The **deleteSpAccountReq** method makes a request to delete the Service Provider Account. The request must be approved before the Service Provider Account is deleted. This is done by the operator, using "deleteSpAccountRes".

Input Parameters

This operation takes no input parameters.

Return Parameters

Void

Possible Exceptions

The possible exceptions are:

- **ACCESS_DENIED**
- **CommonException**
- **INVALID_STATE**: The Service Provider Account's state cannot be deleted. For a Service Provider Account to be deleted, it must be in the INACTIVE state.

deactivateSpAccount

Deactivates the Service Provider Account, which changes the state of the account to INACTIVE. This operation takes no input parameters.

Table 6–1 deactivateSpAccount()

Parameter Name	Type	Description
deactivateSpAccountReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The Service Provider Account is not in an appropriate state to allow the account to be deactivated.

activateSpAccount

Activates the Service Provider Account. which changes the state of the account to ACTIVE. This operation takes no input parameters.

Table 6–2 activateSpAccount()

Parameter Name	Type	Description
activateSpAccountReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The Service Provider Account is not in an appropriate state to allow the account to be activated.

getSpAccount

Retrieves details about the Service Provider Account. The details include contact details and CRM/PRM application-defined properties in the form of name-value pairs. This operation takes no input parameters.

Table 6–3 getSpAccount()

Parameter Name	Type	Description
getSpAccountReturn	tns1:SpAccount	Return parameter. A data structure with details on the Service Provider account. See "SpAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getSpAccountState

Retrieves the state of the Service Provider Account. This operation takes no input parameters.

Table 6–4 getSpAccountState()

Parameter Name	Type	Description
getSpAccountStateReturn	tns1:State	Return parameter. Information of the state of the Service Provider Account. See "State".

Table 6–4 (Cont.) getSpAccountState()

Parameter Name	Type	Description
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

registerAppAccountReq

Requests registration of an Application Account for the Service Provider Account. This request must be approved by the Operator: see "registerAppAccountRes".

Table 6–5 registerAppAccountReq(appAccountId, app)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
app	tns1:AppAccount	Input parameter. A data structure with details about the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "AppAccount".
registerAppAccountReqReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

deleteAppAccountReq

Requests deletion of an Application Account associated with the Service Provider Account. The request must be approved by the Operator before the Application Account is delete. This is done using "deleteAppAccountRes". The Application Account must be in state INACTIVE in order for this call to be accepted.

Table 6–6 deleteAppAccountReq(appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
deleteAppAccountReqReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateSpAccountReq

Requests an update of the Service Provider Account with new data.

Table 6–7 updateSpAccount(spAccountId, spAccount)

Parameter Name	Type	Description
spAccount	tns1:SpAccount	Input parameter. The data structure with details on the Service Provider Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "SpAccount".
updateSpAccountReturn	void	Return paremeter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateAppAccountReq

Requests an update to an Application Account associated with the Service Provider with new data.

Table 6–8 updateAppAccount(appAccountId, appAccount)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
app	tns1:AppAccount	Input parameter. A data structure with details on the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "AppAccount".
updateAppAccountReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateAppInstGroupReq

Requests an update on an Application Instance. The request must be approved by the Operator. See "updateAppInstGroupRes".

Table 6–9 updateAppInstGroup(appAccountId, appInstGroupId, appInstGroup)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.

Table 6–9 (Cont.) updateAppInstGroup(appAccountId, appInstGroupId, appInstGroup)

Parameter Name	Type	Description
appInstGroup	tns1:AppInstGroup	Input parameter. A data-structure with the Application Instance CRM/PRM application-defined properties in the form of name-value pairs. See "AppInstGroup".
updateAppInstGroupReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

deleteAppInstGroupReq

Requests a deletion of an Application Instance. The request must be approved by the Operator. This is done using "deleteAppInstGroupRes".

Table 6–10 deleteAppInstGroupReq(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
deleteAppInstGroupReqReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

listAppAccounts

Lists all Application Account IDs for the Service Provider. The result is filtered based on the state of the Application Account

Table 6–11 listAppAccounts(state)

Parameter Name	Type	Description
state	tns1:State	Input parameter. Indicates the states on which states to filter the result. "State".
listAppAccountsReturn	Array of xsd:string	Return parameter. A list of IDs of Application Accounts.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppAccount

Retrieves details about a specific Application Account. The operation returns a data structure with details on the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs.

Table 6–12 getAppAccount(appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getAppAccountReturn	tns1:AppAccount	Return parameter. A data structure with details about the Application Account. See "AppAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppAccountState

Retrieves the state of a specific Application Account.

Table 6–13 getAppAccountState(appAccountId

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getAppAccountStateReturn	tns1:State	Return parameter. Information on the state of the Application Account. See "State"
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

activateAppAccount

Activate an Application Account, which changes the state of the account to ACTIVE. The current state of the account must be INACTIVE.

Table 6–14 activateAppAccount(appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
activateAppAccountReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Table 6–14 (Cont.) activateAppAccount(appAccountId)

Parameter Name	Type	Description
INVALID_STATE	N/A	Exception. The Application Account is not in an appropriate state to allow the account to be activated.

deactivateAppAccount

Deactivates an Application Account, which changes the state of the account to INACTIVE. The current state of the account must be ACTIVE.

Table 6–15 deactivateAppAccount(appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
deactivateAppAccountReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The state of the Application Account does not allow the account to be deactivated.

activateAppInstGroup

Activates an Application Instance, which changes the state of the instance to ACTIVE.

Table 6–16 activateAppInstGroup(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Inpout parameter. The ID of the Application Instance.
activateAppInstGroupReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The Application Instance is not in an appropriate state to allow the account to be activated.

getAppInstGroupState

Gets the state of an Application Instance.

Table 6–17 getAppInstGroupState(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
getAppInstGroupStateReturn	tns1:State	Return parameter. The state of the Application Instance. See "State".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

deactivateAppInstGroup

Deactivates an Application Instance, which changes the state of the group to INACTIVE.

Table 6–18 deactivateAppInstGroup(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
deactivateAppInstGroupReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The Application Instance is not in an appropriate state to allow the account to be deactivated.

registerAppInstGroupReq

Requests the registration of an Application Instance for a specific Application Account. When this request has been approved by the Operator, see "registerAppInstGroupRes", an application has all credentials necessary to be authenticated on the traffic interfaces of the Oracle Communications Services Gatekeeper.

Table 6–19 registerAppInstGroupReq(appAccountId, appInstGroupId, appInstGroup, password)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.

Table 6–19 (Cont.) registerAppInstGroupReq(appAccountId, appInstGroupId, appInstGroup, password)

Parameter Name	Type	Description
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance to be registered.
appInstGroup	tns1:AppInstGroup	Input parameter. The CRM/PRM application-defined properties in the form of name-value pairs. See "AppInstGroup".
password	xsd:string	Input parameter. The password the application will use when authenticating on Oracle Communications Services Gatekeeper
registerAppInstGroupReqReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

deactivateAppInstGroup

Deactivate an Application Instance. Change the state of the instance to INACTIVE.

Table 6–20 deactivateAppInstGroup(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
deactivateAppInstGroupReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception. The Application Instance is not in an appropriate state to allow the account to be activated.

listAppInstGroups

Lists all Application Instances for an Application Account. Filtering is possible on the state of the Application Instance

Table 6–21 listAppInstGroups(appAccountId, state)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
state	tns1:State	Input parameter. Indicates which states to filter the result on. See "State"
listAppInstGroupsReturn	Array of xsd:string	Return parameter. A list of all Service Application Instance IDs matching the given criteria.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppInstGroup

Retrieves details about a specific Application Instance.

Table 6–22 getAppInstGroup(appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
getAppInstGroupReturn	tns1:AppInstGroup	return parameter. A data structure with details about the Application Instance. See "AppInstGroup".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getSpAccountSla

Deprecated. Use "getSpAccountSlaByType".

Note: The new mechanism (getSpAccountSlaByType) adds a parameter, slaType. Requests made using getSpAccountSla have this parameter hardcoded to a value of service_provider.

Retrieves the Service Provider SLA of the currently logged in service provider. This operation takes no input parameters.

Table 6–23 getSpAccountSla()

Parameter Name	Type	Description
getSpAccountSlaReturn	xsd:string	Return parameter. The Service Provider SLA.

Table 6–23 (Cont.) getSpAccountSla()

Parameter Name	Type	Description
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getSpAccountSlaByType

Retrieves an SLA of a given type for the Service Provider Account.

Table 6–24 getSpAccountSlaByType(slaType)

Parameter Name	Type	Description
slaType	xsd:string	Input parameter. The SLA type to retrieve. Use: ▪ service_provider ▪ system:geo_service_provider ▪ service_provider_node ▪ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
getSpAccountSlaByType	xsd:string	Return parameter. The SLA.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppAccountSla

Deprecated. Use "getAppAccountSlaByType".

> **Note:** The new mechanism (getAppAccountSlaByType) adds a parameter, slaType. Requests made using getAppAccountSla have this parameter hardcoded to a value of application.

Retrieves the Application-level SLA.

Table 6–25 getAppAccountSla(appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The Application Account to retrieve the SLA for.
getAppAccountSlaReturn	xsd:string	Return parameter. The application-level SLA.
ACCESS_DENIED	N/A	Exception

Table 6–25 (Cont.) getAppAccountSla(appAccountId)

Parameter Name	Type	Description
CommonException	N/A	Exception

getAppAccountSlaByType

Retrieves an SLA of a given type for the Application Account.

Table 6–26 getAppAccountSlaByType(slaType, appAccountId)

Parameter Name	Type	Description
slaType	xsd:string	Input parameter. The SLA type to retrieve. Use: ■ application ■ system:geo_ application ■ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
appAccountId	xsd:string	Input parameter. The Application Account to retrieve the SLA for.
getAppAccountSlaRetur n	xsd:string	Return parameter. The application-level SLA.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

setAppInstGroupPassword

Set the password associated with an Application Instance. This password is a part of the credentials an application uses to be authenticated on the traffic interfaces exposed by Oracle Communications Services Gatekeeper.

Table 6–27 setAppInstGroupPassword(appAccountId, appInstGroupId, newPassword)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
newPassword	xsd:string	Input parameter. The new password.
setAppInstGroupPasswordReturn	void	Return parameter
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

changeSpAccountPassword

Change the Service provider Account password. This password is the password the Service Provider use to login to the Service Provider part of the Partner Management Interfaces exposed by Oracle Communications Services Gatekeeper.

Table 6–28 changeSpAccountPassword(oldPassword, newPassword

Parameter Name	Type	Description
oldPassword	xsd:string	Input parameter. The current password.
newPassword	xsd:string	Input parameter. The new password.
changeSpAccountPasswordReturn	void	Return parameter.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

These exceptions are thrown by this interface.

ACCESS_DENIED

Exceptions of this type are raised when the operation is not permited. The user does not have adequate privileges to perform the operation.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

INVALID_STATE

The operation is not valid since the operation results in a state transition, and the transition from the current state to the state the operation results in is not allowed.

Data types

These are data types used by this interface.

AppAccount

Description of an Application Account.

Table 6–29 appAccount

Element name	Datatype	Description
Name	xsd:string	Descriptive name of the Application Account.
Description	xsd:string	Short description of the Application Account.
Properties	impl:ArrayOf_tns1_Property	CRM/PRM application-defined name value pairs. See "Property".

AppInstGroup

Container for the CRM/PRM application-defined properties.

Table 6–30 AppGroup

Element name	Datatype	Description
Name	xsd:string	Descriptive name of the Application Instance.
Description	xsd:string	Short description of the Application Instance.
SLA	xsd:string	The SLA for the Application Instance. Is always "1" as the SLA feature is no longer used.
Properties	mpl:ArrayOf_tns1_ Property	CRM/PRM application-defined properties. Name-value pairs. See "Property".

Property

Array of name-value pairs. This datatype is used in several other datatypes specific for this interface. The properties are accessible from the Service Provider interface and the Operator interface, so they can be used for communicating information between the Service Provider and the Operator

Table 6–31 Property

Element name	Datatype	Description
Name	xsd:string	Name of the property, with the value defined in Value. Unique with the array.
Value	xsd:string	The data associated with Name.

SpAccount

Description of a Service Provider Account, including contact details.

Table 6–32 appAccount

Element name	Datatype	Description
Name	xsd:string	Name of the Service Provider.
Address	xsd:string	Address of the Service Provider.
EMailAddress	xsd:string	E-mail address of the Service Provider.
ContactPerson	xsd:string	Contact person at the Service provider.
PhoneNumber	xsd:string	Phone number to the Service Provider.
Properties	mpl:ArrayOf_tns1_ Property	CRM/PRM application-defined name value pairs. See "Property".

State

Defines the state of a Service Provider Account, Service Provider Group, Application Account, Application Group, or Application Instance. Enumeration.

See"Account States" for more information about states, and transitions between different states.

Table 6–33 State

Element name	Datatype	Description
REGISTERED	xsd:string	The account or group is has been registered. The Operator must respond to this registration request.
ACTIVE	xsd:string	Normal mode.

Table 6–33 (Cont.) State

Element name	Datatype	Description
INACTIVE	N/A	No traffic is allowed through the Oracle Communications Services Gatekeeper when the account or group is in this state.
UPDATE_PENDING	N/A	There is a pending update request. The request must be responded to by the Operator.
DELETE_PENDING	N/A	There is a pending delete request on the account or group. The request must be responded to by the Operator
LOCKED	N/A	Only valid for an Applicator Instances. For backwards compatibility only. The group can be locked due to too many consecutive failed login attempts from an application.

7

Service Provider CDR Utility

The Service Provider Utility Web Service allows the Service Provider to retrieve the CDRs it has generated from Oracle Communications Services Gatekeeper.]

Interface: SpCdrUtil

The endpoint for this interface is: http://<host>:<port>/prm_sp/services/SpCdrUtil

where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

countCdrs

Counts the number of CDRs for a certain Service for a specified time interval.

Note: A Service is the generic name for a Oracle Communications Services Gatekeeper communication service, without regard for the Web Service version or the network plug-in being used. So, for example, the Service name for Parlay X 2.1 Third Party Call using SIP or INAP or Parlay X 3.0 using Parlay 3.3 MPCC is simply Third Party Call.

Table 7–1 countCdrs(serviceName, fromDate, toDate, completionStatus, appAccountId)

Parameter Name	Type	Description
serviceName	xsd:string	Input parameter. The name of the Service for which to retrieve CDRs. Use null to not filter on this parameter
fromDate	xsd:dateTime	Input parameter. From date and time. Use null to not filter on this parameter
toDate	xsd:dateTime	Input parameter. To date and time. Use null to not filter on this parameter.

Table 7–1 (Cont.) countCdrs(serviceName, fromDate, toDate, completionStatus,

Parameter Name	Type	Description
completionStatus	tns1:CdrCompletionStatus	Input parameter. Completion status of the CDR. See "CdrCompletionStatus". Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter
countCdrsReturn	xsd:long	Return parameter. The number of CDRs matching the given criteria.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

listCdrs

Retrieves all CDRs matching the given criteria.

Table 7–2 listCdrs(serviceName, fromDate, toDate, completionStatus, appAccountId, startIndex, maxEntries)

Parameter Name	Type	Description
serviceName	xsd:string	Input parameter. The name of the Service for which to retrieve CDRs. Use null to not filter on this parameter
fromDate	xsd:dateTime	Input parameter. From the date and time. Use null to not filter on this parameter.
toDate	xsd:dateTime	Input parameter. To the date and time. Use null to not filter on this parameter.
completionStatus	tns1:CdrCompletionStatus	Input parameter. Completion status of the CDR.See "CdrCompletionStatus". Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter.

Table 7–2 (Cont.) listCdrs(serviceName, fromDate, toDate, completionStatus, appAccountId, startIndex, maxEntries)

Parameter Name	Type	Description
startIndex	xsd:long	nput parameter. Which entry, in the overall result set, to start the result list on (cursor).
maxEntries	xsd:int	nput parameter. The maximum number of alarms returned.
listCdrsReturn	Array of tns1:CdrInfo	Return parameter. List of CDRS. See "CdrInfo".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

This exception is thrown by this interface.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These complex data types are used by this interface.

CdrInfo

Data structure defining a CDR. All services that produce charging data do not use all fields, and they use the fields in a slightly different, depending on the type of the service. See *Communication Service Guide* for details of which fields that are relevant for the different services.

Table 7–3 CdrInfo

Element name	Datatype	Description
transactionId	xsd:long	The Oracle Communications Services Gatekeeper transaction sequence number.
serviceName	xsd:string	The communication service whose use is being tracked
timeStamp	xsd:dateTime	The time at which the event was triggered (in milliseconds from midnight 1 January 1970)
origAddr	xsd:string	The address of the originating party.
destAddr	xsd:string	The address of the destination party.
spAccountD	xsd:string	The ID of the Service Provider that generated the CDR.
appAccountID	xsd:string	The ID of the Application Account that generated the CDR.
completionStatus	tns1:CdrCompletionStatus	Completion status of the CDR. See "CdrCompletionStatus".

Table 7–3 (Cont.) CdrInfo

Element name	Datatype	Description
info	xsd:string	Additional info provided by the communication service
additionalProperties	impl:ArrayOf_tns1_Property	Application defined data. See "CdrCompletionStatus".

CdrCompletionStatus

Defines the completion status of a CDR. Enumeration

Table 7–4 CdrCompletionStatus

Element name	Datatype	Description
COMPLETED	xsd:string	The operation generating the CDR succeeded.
FAILED	xsd:string	The operation generating the CDR failed.
PARTIAL	xsd:string	The operation generating the CDR succeeded partially. May be supported, depending on the communication service.
COMPLETED_NOTIFICATION_FAILED	xsd:string	The CDR is completed, but the notification was not delivered to the application.
POLICY_DENIED	xsd:string	Policy denied the operation.

Property

Array of name-value pairs.

Table 7–5 Property

Element name	Datatype	Description
Name	xsd:string	Name of the property, with the value defined in Value. Unique within the array.
Value	xsd:string	The data associated with Name.

8

Service Provider Statistics Utility

The Service Provider Statistics Utility Web Service allows the Service Provider to retrieve the statistics it has generated from Oracle Communications Services Gatekeeper.

Interface: SpService

The endpoint for this interface is: http://<host>:<port>/prm_sp/services/SpStatisticsUtil

Where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

listStatisticTypes

Lists the statistics types registered in Oracle Communications Services Gatekeeper. This operation takes no input parameters.

Table 8–1 listStatisticTypes()

Parameter Name	Type	Description
listStatisticTypesReturn	Array of tns1:StatisticTypeDescriptor	Return parameter. Descriptions of available statistics types. See "StatisticTypeDescriptor".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getStatistics

Retrieves statistics matching the given criteria.

Table 8–2 getStatistics(statisticType, fromDate, toDate, spAccountId, appAccountId)

Parameter Name	Type	Description
statisticType	xsd:int	Input parameter. Number representing the type of statistics to retrieve. Use null to not filter on this parameter

Table 8–2 (Cont.) getStatistics(statisticType, fromDate, toDate, spAccountId,

Parameter Name	Type	Description
fromDate	xsd:dateTime	Input parameter. From date and time. Use null to not filter on this parameter.
toDate	xsd:dateTime	Input parameter. To date and time. Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter.
getStatisticsReturn	Array of tns1:StatisticsInfo	Return parameter. Statistics. See "StatisticsInfo".
CommonException	N/A	Exception

Exceptions

This exception is thrown by this interface.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These complex data types are used by this interface.

StatisticsInfo

Data structure defining a statistics record.All services that produces statistics do not use all fields, and they use the fields in a slightly different, depending on the type of the service. See *Communication Service Guide* for details of which fields that are relevant for the different services.

Table 8–3 StatisticsInfo

Element name	Datatype	Description
statisticsType	xsd:string	The statistics type. See WebLogic Product Description for information on available statistics types.
timeStampStart	xsd:dateTime	The starting time of the interval during which the statistics were gathered.
timeStampEnd	xsd:dateTime	The end time of the interval during which the statistics were gathered.
numberOfTransactions	xsd:int	The number of transactions during the interval.
spAccountD	xsd:string	The ID of the Service Provider that generated the statistics.
appAccountID	xsd:string	The ID of the Application Account that generated the statistics.

StatisticTypeDescriptor

Holds a description for each type of statistics.

Table 8–4 StatisticTypeDescriptor

Element name	Datatype	Description
transactionTypeName	xsd:string	Name of a transaction type that statistics can be generated for.
transactionTypeID	xsd:int	Numeric ID of the transaction type.

9

Operator Login

The Operator Login Web Service is provided for backwards compatibility only. The current implement of PRM is sessionless.

Interface: OpLogin

The endpoint for this interface is: http://<host>:<port>/prm_op/services/OpLogin

Where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

login (Deprecated. For backwards-compatibility only)

Used by a CRM/PRM application to login and retrieve a loginTicket that servers as an identifier when the Operator part of the Partner Relationship Management interface is used.

Table 9–1 login(operatorId, password)

Parameter Name	Type	Description
operatorId	xsd:string	Input parameter. The ID of operator account. This account is created using the Services Gatekeeper Administration console extension.
password	xsd:string	Input parameter. The password associated with the password.
loginReturn	xsd:string	Return parameter. A loginTicket used to identify the login session. This ticket shall be supplied in the SOAP header for each method invocation.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

logout (Deprecated. For backwards-compatibility only)

Used by a CRM/PRM application to logout and destroy the login session and the corresponding loginTicket

Table 9–2 logout(loginTicket)

Parameter Name	Type	Description
loginTicket	xsd:string	Input parameter. The loginTicket retrieved when logging in.
logoutReturn	void	Return parameter.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

These exceptions are thrown by this interface.

ACCESS_DENIED

Exceptions of this type are raised when access to the method is denied.

CommonException

This exception is raised when the login session has expired or there are communication problems with the underlying platform.

Complex data types

There are no complex data types.

10

Operator Service

The Operator Service Web Service provides the Operator with operations for handling Service Provider Accounts, Service Provider Groups, Application Accounts, Application Account Groups and Applications Instances.

Interface: OpService

The endpoint for this interface is: http://<host>:<port>/prm_op/services/OpService

where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

listAppGroups

Lists all Application Groups. This operation takes no input parameters.

Table 10–1 listAppGroups()

Parameter Name	Type	Description
listAppGroupsReturn	Array of xsd:string	Return parameter. A list of all Application Group IDs.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppGroup

Retrieves details about a specific Application Group.

Table 10–2 getAppGroup(appGroupId)

Parameter Name	Type	Description
appGroupId	xsd:string	Input parameter. The ID of the group.
getAppGroupReturn	tns1:AppGroup	Return parameter. A data structure with details about the Application Group. See "AppGroup".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

createAppGroup

Deprecated. Use "createAppGroupByType".

Creates a new Application Group with a certain ID

> **Note:** The new mechanism (createAppGroupByType) for creating application groups adds a parameter, slaType. Groups created using createAppGroup automatically have the slaType hardcoded to the value application.

Table 10–3 createAppGroup(appGroupId, appGroup)

Parameter Name	Type	Description
appGroupId	xsd:string	Input parameter. The ID of the Application Group.
appGroup	tns1:AppGroup	Input parameter. A data structure describing the group. See "AppGroup".
createAppGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

createAppGroupByType

Creates a new Application Group with a certain ID with an associated SLA.

Table 10–4 createAppGroupByType(slaType, appGroupId, appGroup)

Parameter Name	Type	Description
slaType	xsd:string	Input parameter. The SLA type to update. Use: ▪ application ▪ system:geo_ application ▪ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
appGroupId	xsd:string	Input parameter. ID of the Application Group.
appGroup	tns1:AppGroup	Input parameter. A data structure describing the group. See "AppGroup".
createAppGroupByType Return	void	Return Type
ACCESS_DENIED	N/A	Exception

Table 10–4 (Cont.) createAppGroupByType(slaType, appGroupId, appGroup)

Parameter Name	Type	Description
CommonException	N/A	Exception

deleteAppGroup

Deletes an Application Group. All Application Accounts must be removed from the group before it can be deleted

Table 10–5 deleteAppGroup(appGroupId)

Parameter Name	Type	Description
appGroupId	xsd:string	The ID of the group.
deleteAppGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

moveAppAccountToGroup

Associates an Application Account for a specific Service Provider with an Application Group.

Table 10–6 moveAppAccountToGroup(spAccountId, appAccountId, appGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appGroupId	xsd:string	Input parameter. The ID of the Application Account Group.
moveAppAccountToGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception.
CommonException	N/A	Exception thrown if none of the accounts or groups exist.

getAppGroupId

Retrieves the ID of the Application Group for a given Service Provider Account and Application Account combination.

Table 10–7 getAppGroupId(spAccountId, appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.

Table 10–7 (Cont.) getAppGroupId(spAccountId, appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getAppGroupIdReturn	xsd:string	Return parameter. The ID of the Application Group.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if any of the accounts does not exist or no group is associated.

updateAppGroup

Deprecated. Use "updateAppGroupByType".

Updates an Application Group with new SLA data.

Note: The new mechanism for updating application groups (updateAppGroupByType) adds a parameter, slaType. Using updateAppGroup automatically hardcodes the slaType to the value application.

Table 10–8 updateAppGroup(appGroupId, appGroup)

Parameter Name	Type	Description
appGroupId	xsd:string	Input parameter. The ID of the group.
appGroup	tns1:AppGroup	Input parameter. The Application-level SLA, and CRM/PRM application-defined properties in the form of name-value pairs. See "AppGroup"
updateAppGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

updateAppGroupByType

Updates an Application Group with a new SLA.

Table 10–9 updateAppGroupByType(slaType, appGroupId, appGroup)

Parameter Name	Type	Description
slaType	xsd:string	Input parameter. The SLA type to update. Use: ▪ application ▪ system:geo_ application ▪ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
appGroupId	xsd:string	Input parameter. The ID of the group.
appGroup	tns1:AppGroup	Input parameter. The Application-level SLA, and CRM/PRM application-defined properties in the form of name-value pairs. See "AppGroup".
updateAppGroupByType Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

listSpGroups

Lists all Service Provider Groups. This operation requires no input.

Table 10–10 listSpGroups()

Parameter Name	Type	Description
listSpGroupsReturn	Array of xsd:string	Return parameter. A list of all service Provider Group IDs.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getSpGroup

Retrieves details about a specific Service Provider Group.

Table 10–11 getSpGroup(spGroupId)

Parameter Name	Type	Description
spGroupId	xsd:string	Input parameter. The ID of the group.

Table 10–11 (Cont.) getSpGroup(spGroupId)

Parameter Name	Type	Description
getSpGroupReturn	tns1:SpGroup	Return parameter. A data structure with details about the Service Provider Group. See "SpGroup".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

createSpGroup

Deprecated. Use "createSpGroupByType".

Creates a new Service Provider Group with a certain ID.

Note: The new mechanism (createSpGroupByType) for creating service provider groups adds a parameter, slaType. Groups created using createSpGroup automatically have the slaType hardcoded to the value service_provider.

Table 10–12 createSpGroup(spGroupId, spGroup)

Parameter Name	Type	Description
spGroupId	xsd:string	Input parameter. The ID of the Service Provider Group ID.
spGroup	tns1:SpGroup	Input parameter. A data structure describing the group. See "SpGroup".
createSpGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

createSpGroupByType

Creates a new Service Provider Group with a certain ID with a certain SLA type associated.

Table 10–13 createSpGroupByType(slaType, spGroupId, spGroup)

Parameter Name	Type	Description
slaType	xsd:string	Input parameter. The SLA type to update. Use: ▪ service_provider ▪ system:geo_service_provider ▪ service_provider_node ▪ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
spGroupId	xsd:string	Input parameter. ID of the Service Provider Group ID.
spGroup	tns1:SpGroup	Input parameter. A data structure describing the group. See "SpGroup".
createSpGroupByTypeReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

deleteSpGroup

Deletes a Service Provider Group. All Service Provider Accounts associated with the group must be removed before it can be deleted

Table 10–14 deleteSpGroup(spGroupId)

Parameter Name	Type	Description
spGroupId	xsd:string	Input parameter. The ID of the group.
deleteSpGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if there are Service Provider accounts associated with the group.

moveSpToGroup

Associates a Service Provider Account with a Service Provider Group.

Table 10–15 moveSpToGroup(spAccountId, SpGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
SpGroupId	xsd:string	Input parameter. The ID of the Service provider Group.
moveSpToGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if any of the accounts does not exist.

getSpGroupId

Retrieves the ID of the Service Provider Group for a given Service Provider Account.

Table 10–16 getSpGroupID(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
getSpGroupIdReturn	xsd:string	Return parameter. The ID of the Service Provider Group.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if the account does not exist.

updateSpGroup

Deprecated. Use "updateSpGroupByType".

Updates a Service Provider Group with new SLA data.

Note: The new mechanism (updateSpGroupByType) for updating service provider groups adds a parameter, slaType. Groups updated using updateSpGroup automatically have the slaType hardcoded to the value service_provider.

Table 10–17 updateSpGroup(spGroupId,spGroup)

Parameter Name	Type	Description
spGroupId	xsd:string	Input parameter. The ID of the group.

Table 10–17 (Cont.) updateSpGroup(spGroupId,spGroup)

Parameter Name	Type	Description
spGroup	tns1:SpGroup	Input parameter. The Service Provider-level SLA, and CRM/PRM application-defined properties in the form of name-value pairs. See "SpGroup".
updateSpGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

updateSpGroupByType

Updates a Service Provider Group with a new SLA.

Table 10–18 updateSpGroupByType(slaType, spGroupId,spGroup)

Parameter Name	Type	Description
slaType	xsd:String	Input parameter. The SLA type to update. Use: ▪ service_provider ▪ system:geo_service_provider ▪ service_provider_node ▪ a custom SLA type ID For information on the different types, see section Managing SLAs in *Accounts and SLAs Guide*.
spGroupId	xsd:string	Input parameter. The ID of the group.
spGroup	tns1:SpGroup	Input parameter. The Service Provider-level SLA, and CRM/PRM application-defined properties in the form of name-value pairs. See "SpGroup".
updateSpGroupByTypeReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

listAppInstGroups

Lists all Application Instances for a given combination of Service Provider Account and Application Account. Filtering is possible on the state of the Application Instance

Table 10–19 listAppInstGroups(spAccountId, appAccountId, state)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account. Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. The ID of the Application Account. Use null to not filter on this parameter.
state	tns1:State	Input parameter. Indicates which states to filter the result on. See "State".
listAppInstGroupsReturn	Array of tns1:AppInstGroupRef	Return parameter. A data structure containing a references to the Application Instance. See "AppInstGroupRef".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppInstGroup

Retrieves details about a specific Application Instance.

Table 10–20 getAppInstGroup(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
Input		
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
getAppInstGroupReturn	tns1:AppInstGroup	Return parameter. A data structure with details about the Application Instance. See "AppInstGroup".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppInstGroupState

Gets the state of specific Application Instance.

Table 10–21 getAppInstGroupState(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
getAppInstGroupStateReturn	tns1:State	Return parameter. The state of the Application Instance. See "State".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

registerAppInstGroupReq

Requests registration of an Application Instance for a specific combination of a Service Provider Account and Application Account. When this request has been approved (see "registerAppInstGroupRes") the newly registered application has all the credentials necessary to be authenticated on the traffic interfaces of the Oracle Communications Services Gatekeeper.

Table 10–22 registerAppInstGroupReq(spAccountId, appAccountId, appInstGroupId, appInstGroup, password)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance to be registered.
appInstGroup	tns1:AppInstGroup	Input parameter. CRM/PRM application-defined properties in the form of name-value pairs.
password	xsd:string	Input parameter. The password the newly created Application Instance will use when authenticating on the Oracle Communications Services Gatekeeper

Table 10–22 (Cont.) registerAppInstGroupReq(spAccountId, appAccountId, appInstGroupId, appInstGroup, password)

Parameter Name	Type	Description
registerAppInstGroupReqReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

deleteAppInstGroupReq

Requests deletion of an Application Instance. The Application Instance must be in state INACTIVE in order for this call to be accepted. The request must be approved before the Application Instance is deleted. This is done using "deleteAppInstGroupRes"

Table 10–23 deleteAppInstGroupReq(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance
deleteAppInstGroupReqReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if there is an SLA associated with the combination of Service Provider Accounts, Application Account, and Application Instance associated with the group.
INVALID_STATE	N/A	Exception

deleteAppInstGroupRes

Responds to a request for deleting an Application Instance for a specific combination of Service Provider Account and Application Account.

It is possible to approve or disapprove the request. Both cases triggers a state transition for the Application Instance. If approved, the Application Instance is deleted, and the Application Instance can no longer be used to authenticate to send the traffic to Oracle Communications Services Gatekeeper.

Table 10–24 deleteAppInstGroupRes(spAccountId, appAccountId, appInstGroupId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
deleteAppInstGroupReq Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if there is an SLA associated with the set of Service Provider Accounts, Application Account, and Application Instance associated with the group.
INVALID_STATE	N/A	Exception

updateAppInstGroup

Updates an Application Instance Group with new data.

Table 10–25 updateAppInstGroup(spAccountId, appAccountId, appInstGroupId, appInstGroup)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
appInstGroup	tns1:AppInstGroup	Input parameter. A data-structure with the Application Instance SLA and application-defined properties in the form of name-value pairs. See "AppInstGroup".
updateAppInstGroupRet urn	void	Return Type

Table 10–25 (Cont.) updateAppInstGroup(spAccountId, appAccountId, appInstGroupId, appInstGroup)

Parameter Name	Type	Description
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

updateAppInstGroupRes

Responds to a request for updating Application Instance for a specific combination of Service Provider Account and Application Account.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Application Instance to state ACTIVE. If approved, the Application Instance is updated with the new information.

Table 10–26 updateAppInstGroupRes(spAccountId, appAccountId, appInstGroupId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
updateAppInstGroupRes Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if group does not exist or if the SLA contains errors.

getUpdatePendingAppInstGroup

Gets details about an specific Application Instance that is in state UPDATE_ PENDING. The details include CRM/PRM application-defined properties in the form of name-value pairs.

Table 10–27 getUpdatePendingAppInstGroup(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.

Table 10–27 (Cont.) getUpdatePendingAppInstGroup(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
getUpdatePendingAppInstGroupReturn	tns1:AppInstGroup	Input parameter. CRM/PRM application-defined properties in the form of name-value pairs. See "AppInstGroup".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the Application Instance is not in state UPDTATE_PENDING.

setAppInstGroupPassword

Sets the password associated with an Application Instance. This password is a part of the credentials an application uses to authenticate to send traffic to Oracle Communications Services Gatekeeper.

Table 10–28 setAppInstGroupPassword(spAccountId, appAccountId, appInstGroupId, newPassword)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
newPassword	xsd:string	Input parameter. The new password.
setAppInstGroupPasswordReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

unlockAppInstGroup

Unlock a locked Application Instance Group. The group may have been locked by too many faulty login attempts to the traffic interfaces exposed by the Oracle Communications Services Gatekeeper.

Table 10–29 unlockAppInstGroup(spAccountId, appAccountId, appInstGroupId, newPassword)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance Group.
unlockAppInstGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

activateSpAccount

Activates a Service Provider Account, which changes the state of the account to ACTIVE.

Table 10–30 activateSpAccount(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
activateSpAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the Service Provider Account is not in an appropriate state to allow the account to be activated.

deactivateSpAccount

Deactivates a a Service Provider Account, which changes the state of the account to INACTIVE.

Table 10–31 deactivateSpAccount(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
deactivateSpAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Table 10–31 (Cont.) deactivateSpAccount(spAccountId)

Parameter Name	Type	Description
INVALID_STATE	N/A	Exception thrown if the Service Provider Account is not in an appropriate state to allow the account to be deactivated.

getSpAccount

Retrieves details about a specific Service Provider Account. The details include contact details and CRM/PRM application-defined properties in the form of name-value pairs.

Table 10–32 getSpAccount(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the account.
getSpAccountReturn	tns1:SpAccount	Input parameter. ata structure with details on the Service Provider account. See "SpAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getSpAccountState

Retrieves the state of a specific Service Provider Account.

Table 10–33 getSpAccountState(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the account.
getSpAccountStateReturn	tns1:State	Return parameter. Information on the state of the Service Provider Account. See "State".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

registerAppAccountReq

Requests registration for an Application Account for a specific Service Provider Account. When this request has been approved by the Operator (see "registerAppAccountRes") the Application Account can be associated with an Application Account Group and an Application Instance.

Table 10–34 registerAppAccountReq(spAccountId, appAccountId, appAccount)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appAccount	tns1:AppAccount	Input parameter. A data structure with details on the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "AppAccount".
registerAppAccountReq Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

registerAppAccountRes

Responds to a request to register an Application Account for a specific Service Provider Account. An Application Account Group is also associated with the Application Account.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Application Account. If approved, the Application Account is transferred into state ACTIVE. If Disapproved, the Application Account is deleted.

Table 10–35 registerAppAccountRes(spAccountId, appAccountId, appGroupId, appAccountRef, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appGroupId	xsd:string	Input parameter. The ID of the Application Account Group to be associate with the Application Account.
appAccountRef	xsd:string	Input parameter. Internal ID of the Application Account. This ID is used to correlate the Application Account ID with an Operator-internal ID.

Table 10–35 (Cont.) registerAppAccountRes(spAccountId, appAccountId, appGroupId, appAccountRef, response)

Parameter Name	Type	Description
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
deleteAppInstGroupReq Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateAppAccountRes

Responds to a request to update an Application Account for a specific Service Provider.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Application Account state to Active. If approved, the Application Account is updated with the new information.

Table 10–36 updateAppAccountRes(spAccountId, appAccountId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
updateAppAccountResR eturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the status of the Application Account is not in state UPDATE_PENDING.

getUpdatePendingAppAccount

Gets details about pending update requests for a specific combination of Service Provider and Application Account. The details includes descriptions and CRM/PRM application-defined properties in the form of name-value pairs. Valid only for Application Accounts in state UPDATE_PENDING.

Table 10–37 getUpdatePendingAppAccount(spAccountId, appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getUpdatePendingAppAccountReturn	tns1:AppAccount	Return parameter. A data structure describing the Application Account. See "AppAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the status of the Application Account is not in state UPDATE_PENDING.

deleteAppAccountReq

Requests the deletion of an Application Account. In order to be deleted, there must be no Application Instance associated with the combination of Service Provider Account and Application Account. The request must be approved before the Application Account is deleted, which is done using "deleteAppAccountRes". The Application Account must be in state INACTIVE in order for this call to be accepted.

Table 10–38 deleteAppAccountReq(spAccountId, appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
deleteAppAccountReqReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

deleteAppAccountRes

Responds to a request to delete an Application Account for a specific Service Provider Account. The Application Account must be in state DELETE_PENDING in order for this call to be accepted.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Application Account. If approved, the Application Account is simply deleted. If Disapproved, the Application Account is transferred into state INACTIVE.

Table 10–39 deleteAppAccountRes(spAccountId, appAccountId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
deleteAppAccountResReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateAppAccount

Updates an Application Account with new data.

Table 10–40 updateAppAccount(spAccountId, appAccountId, appAccount)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appAccount	tns1:AppAccount	Input parameter. Data structure with details on the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "AppAccount".
updateAppAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

getAppAccount

Retrieves details about a specific Application Account. The return includes a data structure with details on the Application Account, including CRM/PRM application-defined properties in the form of name-value pairs.

Table 10–41 getAppAccount(spAccountId, appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.

Table 10–41 (Cont.) getAppAccount(spAccountId, appAccountId)

Parameter Name	Type	Description
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getAppAccountReturn	tns1:AppAccount	Return parameter. A data structure with details about the Application Account. See "AppAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getAppAccountState

Retrieves the state of a specific Application Account.

Table 10–42 getAppAccountState(spAccountId, appAccountId

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
getAppAccountStateReturn	tns1:State	Return parameter. Information of the state of the Application Account. See "State"
Possible Exceptions	N/A	Exception
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

activateAppAccount

Activates an Application Account, which changes the state of the account to ACTIVE. The current state of the account must be INACTIVE.

Table 10–43 activateAppAccount(spAccs appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
activateAppAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Table 10–43 (Cont.) activateAppAccount(spAccs appAccountId)

Parameter Name	Type	Description
INVALID_STATE	N/A	Exception thrown if the state of the Application Account does not allow the account to be activated.

deactivateAppAccount

Deactivates an Application Account, which changes the state of the account to INACTIVE. The current state of the account must be ACTIVE.

Table 10–44 deactivateAppAccount(spAccountId, appAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
deactivateAppAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the Application Account is not in an appropriate state to allow the account to be deactivated.

registerAppInstGroupRes

Responds to a request to register an Application Instance for a specific combination of a Service Provider Account and Application Group.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Application Instance. If approved, the Application Instance is transferred into state ACTIVE and the application can authenticate with the traffic interfaces exposed by Oracle Communications Services Gatekeeper. If disapproved, the Application Instance is deleted.

Table 10–45 registerAppInstGroupRes(spAccountId, appAccountId, appInstGroupId, appInstGroupRef, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.

Table 10–45 (Cont.) registerAppInstGroupRes(spAccountId, appAccountId, appInstGroupId, appInstGroupRef, response)

Parameter Name	Type	Description
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
appInstGroupRef	xsd:string	Input parameter. Internal ID of the Application Instance. This ID is used to correlate the Application Instance ID with an Operator-internal ID.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
registerAppInstGroupResReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

activateAppInstGroup

Activates an Application Instance, which changes the state of the Instance to ACTIVE.

Table 10–46 activateAppInstGroup(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
activateAppInstGroupReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

deactivateAppInstGroup

Deactivates an Application Instance, which changes the state of the Instance to INACTIVE.

Table 10–47 deactivateAppInstGroup(spAccountId, appAccountId, appInstGroupId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
appAccountId	xsd:string	Input parameter. The ID of the Application Account.
appInstGroupId	xsd:string	Input parameter. The ID of the Application Instance.
deactivateAppInstGroup Return	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the state of the Application Instance does not allow the account to be deactivated.

registerSpAccountReq

Requests registration for a Service Provider Account. Contact details are supplied in the request, together with CRM/PRM application-defined properties. This request must be approved by the Operator (see "registerSpAccountRes").

Table 10–48 registerSpAccountReq(spAccountId, spAccount, password)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The desired ID of the Service Provider Account. The Service Provider will use this ID when authenticating to the PRM-SP Web Services.
spAccount	xsd:string	Input parameter. Data structure with details about the Service Provider Account.
password	xsd:string	Input parameter. The password the Service Provider will use when authenticating to the Service Provider part of the Partner Relationship Management interface.
registerSpAccountReqRe turn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

listAppAccounts

Lists all Application Account IDs for a specific Service Provider. The result is filtered on the state of the Application Account

Table 10–49 listAppAccounts(spAccountId, state)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account. Use null to not filter on this parameter
state	tns1:State	Input parameter. Indicates which states to filter the result on. "State".
listAppAccountsReturn	Array of tns1:AppAccountRef	Return parameter. A data structure containing references to Application Account Groups. See "AppAccountRef".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

listSpAccounts

Lists all Service Provider Account IDs. The result is filtered on the state of state of the Service Provider Account

Table 10–50 listSpAccounts(state)

Parameter Name	Type	Description
state	tns1:State	Input parameter. Indicates which states to filter the result on. See "State".
listAppAccountsReturn	Array of xsd:string	Return paramenetr. A list of IDs of the Service Provider Accounts.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

registerSpAccountRes

Responds to a request to register a Service Provider Account.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Service Provider Account. If approved, the Service Provider Account is transferred into state ACTIVE and the Service provider can authenticate with the PRM-SP Web Services. If Disapproved, the Service Provider Account is deleted.

Table 10–51 registerSpAccountRes(spAccountId, spGroupId, spAccountRef, response)

Parameter Name	Type	Description
Input		

Table 10–51 (Cont.) registerSpAccountRes(spAccountId, spGroupId, spAccountRef,

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
spGroupId	xsd:string	Input parameter. The ID of the Service Provider Group the Service Provider Account should be associated with.
spAccountRef	xsd:string	Input parameter. Internal ID of the Service Provider. This ID is used to correlate the Service Provider ID with an Operator-internal ID.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
registerSpAccountResReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the status of the Service Provider Account is not in state REGISTERED.

deleteSpAccountReq

Requests deletion of a Service Provider Account. In order to be deleted, the Service Provider Account must be state INACTIVE. The request must be approved before the Service Provider Account it is deleted. This is done using "deleteSpAccountRes".

Table 10–52 deleteSpAccountReq(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Account.
deleteSpAccountReqReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception thrown if the status of the Service Provider Account is not in state INACTIVE.
INVALID_STATE	N/A	Exception

deleteSpAccountRes

Responds to a request to delete a Service Provider Account. The Service Provider Account must be in state DELETE_PENDING in order for this call to be accepted.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Service Provider Account. If approved, the Service Provider Account is simply deleted. If disapproved, the Service Provider Account is transferred into state INACTIVE.

Table 10–53 deleteSpAccountRes(spAccountId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
deleteSpAccountResReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the status of the Service Provider Account is not in state INACTIVE.

updateSpAccount

Updates a Service Provider Account with new data.

Table 10–54 updateSpAccount(spAccountId, spAccount)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
spAccount	tns1:SpAccount	Input parameter. Data structure with details on the Service Provider Account, including CRM/PRM application-defined properties in the form of name-value pairs. See "SpAccount".
updateSpAccountReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception

updateSpAccountRes

Responds to a request to update a Service Provider Account.

It is possible to approve or disapprove the request. Both cases trigger a state transition for the Service Provider Account to state ACTIVE. If approved, the Service Provider Account is updated with the new information.

Table 10–55 updateSpAccountRes(spAccountId, response)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
response	tns1:RequestResponse	Input parameter. The response to the request. See "RequestResponse".
updateAppInstGroupResReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	If the status of the Service Provider Account is not in state INACTIVE, this exception is thrown.

getUpdatePendingSpAccount

Gets details about a specific Service Provider account that is in state UPDATE_PENDING. The details include contact information and CRM/PRM application-defined properties in the form of name-value pairs.

Table 10–56 getUpdatePendingSpAccount(spAccountId)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
getUpdatePendingSpAccountReturn	tns1:SpAccount	Details about contact information and CRM/PRM application-defined properties in the form of name-value pairs. See "SpAccount".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception
INVALID_STATE	N/A	Exception thrown if the Service Provide r Account is not in state UPDTATE_PENDING.

setSpAccountPassword

Sets the password the Service Provider uses to authenticate to use the Partner Relationship Management Interface exposed by Oracle Communications Services Gatekeeper.

Table 10–57 setSpAccountPassword(spAccountId, newPassword)

Parameter Name	Type	Description
Input		

Table 10–57 (Cont.) setSpAccountPassword(spAccountId, newPassword)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. The ID of the Service Provider Account.
newPassword	xsd:string	Input parameter. The new password.
Returns		
setSpAccountPasswordReturn	void	
Possible Exceptions		
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

changeOpAccountPassword

Changes the password the Operator uses to authenticate with the Partner Relationship Management Interface exposed by Oracle Communications Services Gatekeeper. The Operator Account is the one the Operator is currently logged in as.

Table 10–58 changeOpAccountPassword(oldPassword, newPassword)

Parameter Name	Type	Description
oldPassword	xsd:string	Input parameter. The password to be changed.
newPassword	xsd:string	Input parameter. The new password.
changeOpAccountPasswordReturn	void	Return Type
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getUserLevel

Retrieves the user level of the currently logged in Operator Account. Different user levels have different privileges, and are authorized to different sets of operations. This operation takes no input.

Table 10–59 changeOpAccountPassword(oldPassword, newPassword)

Parameter Name	Type	Description
getUserLevelReturn	tns1:UserLevel	Return parameter. The user level of the currently logged in Operator Account. See "UserLevel".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

These exceptions are thrown by this interface.

ACCESS_DENIED

Exceptions of this type are raised when the operation is not permitted. The user does not have the appropriate privilege level to perform the operation.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These complex data types are used by this interface.

AppAccount

Description of an Application Account.

Table 10–60 appAccount

Element name	Datatype	Description
name	xsd:string	Descriptive name of the Application Account.
description	xsd:string	Short description of the Application Account.
properties	impl:ArrayOf_tns1_Property	CRM/PRM application-defined name value pairs. See "Property".

AppAccountRef

Reference to IDs of an Application Account.

Table 10–61 AppInstGroupRef

Element name	Datatype	Description
spAccountId	xsd:string	ID of the Service Provider Account associated with the Application Account.
appAccountId	xsd:string	ID of the Application Account.

AppInstGroupRef

Reference to IDs of an Application Instance.

Table 10–62 AppInstGroupRef

Element name	Datatype	Description
spAccountId	xsd:string	ID of the Service Provider Account associated with the Application Instance.
appAccountId	xsd:string	ID of the Application Account associated with the Application Instance.
appInstGroupId	xsd:string	ID of the Application Instance.

AppGroup

Container for Application-level SLA and CRM/PRM application-defined properties.

Table 10–63 AppGroup

Element name	Datatype	Description
sla	xsd:string	The SLA for the Application Group.
properties	mpl:ArrayOf_tns1_ Property	CRM/PRM application-defined properties. Name-value pairs. See "Property".

AppInstGroup

Container for Application Instance CRM/PRM application-defined properties.

Table 10–64 AppGroup

Element name	Datatype	Description
name	xsd:string	Descriptive name of the Application Instance.
description	xsd:string	Short description of the Application Instance.
sla	xsd:string	The SLA for the Application Instance. Always set to 1.
properties	mpl:ArrayOf_tns1_ Property	CRM/PRM application-defined properties. Name-value pairs. See "Property".

Property

Array of name-value pairs. This datatype is used in several other datatypes specific to this interface. The properties are accessible from the Service Provider interface and the Operator interface, so they can be used to communicate information between them

Table 10–65 Property

Element name	Datatype	Description
name	xsd:string	Name of the property, with the value defined in Value. Unique with the array.
value	xsd:string	The data associated with Name.

RequestResponse

Enumeration defining the operator's response to an request

Table 10–66 RequestResponse

Element name	Datatype	Description
APPROVE	xsd:string	Used when the operator approves the request.
DISAPPROVE	xsd:string	Used when the operator disapproves the request.

SpAccount

Description of a Service Provider Account, including contact details.

Table 10–67 appAccount

Element name	Datatype	Description
name	xsd:string	Name of the Service Provider.
address	xsd:string	Address of the Service Provider.
eMailAddress	xsd:string	E-mail address of the Service Provider.

Table 10–67 (Cont.) appAccount

Element name	Datatype	Description
contactPerson	xsd:string	Contact person at the Service provider.
phoneNumber	xsd:string	Phone number to the Service Provider.
properties	mpl:ArrayOf_tns1_ Property	CRM/PRM application-defined name value pairs. See "Property".

SpGroup

Container for Service Provider SLA and CRM/PRM application-defined properties.

Table 10–68 SpGroup

Element name	Datatype	Description
sla	xsd:string	The SLA for the Service Provider.
properties	impl:ArrayOf_tns1_ Property	CRM/PRM Application-defined properties. Name-value pairs. See "Property".

UserLevel

Defines the user level of the currently logged in Operator user. Enumeration.

The user level reflects the user levels defined for the operations and maintenance of Oracle Communications Services Gatekeeper. Each operation performed via the Partner Management Interface results in one or more standard OAM operations. The user level of the currently authenticated user must satisfy the user level necessary for each of these operations. If this is not the case, the operation performed through the Partner Management Interface is denied.

Table 10–69 State

Element name	Datatype	Description
UNAUTHORIZED	xsd:string	The currently authenticated in user is not authorized to perform any OAM operations.
READ_ONLY	xsd:string	The currently authenticated user is authorized to perform OAM read- or get- operations.
READ_WRITE	xsd:string	The currently authenticated user is authorized to perform OAM write- or set- operations.
ADMINISTRATOR	xsd:string	The currently authenticated user is authorized to perform administrator OAM operations tasks

State

Defines the state of a Service Provider Account, Service Provider Group, Application Account, Application Group, or Application Instance. Enumeration.

See "Account States" for more information about states, and transitions among different states.

Table 10–70 State

Element name	Datatype	Description
REGISTERED	xsd:string	The account or group is has been registered. The registration request must be responded to by the Service Provider.

Table 10–70 (Cont.) State

Element name	Datatype	Description
ACTIVE	xsd:string	Normal mode.
INACTIVE	N/A	No traffic is allowed through the Oracle Communications Services Gatekeeper when the account or group is in this state.
UPDATE_PENDING	N/A	There is a pending update request. The request must be responded to by the Service Provider.
DELETE_PENDING	N/A	There is a pending delete request on the account or group. The request must be responded to by the Service Provider.
LOCKED	N/A	Only valid for an Application Instance. The group can be locked due to too many consecutive failed login attempts from an application.

11

Operator CDR Utility

The Operator CDR Utility Web Service allows the Operator to retrieve CDRs from Oracle Communications Services Gatekeeper.

Interface: OpCdrUtil

The endpoint for this interface is: http://<host>:<port>/prm_op/services/OpCdrUtil

where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

countCdrs

Counts the number of CDRs for a certain Service for a specified time interval

> **Note:** A Service is the generic name for a Oracle Communications Services Gatekeeper communication service, without regard for the Web Service version or the network plug-in being used. So, for example, the Service name for Parlay X 2.1 Third Party Call using SIP or INAP or Parlay X 3.0 using Parlay 3.3 MPCC is simply Third Party Call.

Table 11–1 countCdrs(serviceName, fromDate, toDate, completionStatus, spAccountId, appAccountId)

Parameter Name	Type	Description
serviceName	xsd:string	Input parameter. The name of the Service for which to retrieve CDRs. Use null to not filter on this parameter.
fromDate	xsd:dateTime	Input parameter. Start date and time. Use null to not filter on this parameter.
toDate	xsd:dateTime	Input parameter. End date and time. Use null to not filter on this parameter.

Table 11–1 (Cont.) countCdrs(serviceName, fromDate, toDate, completionStatus, spAccountId, appAccountId)

Parameter Name	Type	Description
completionStatus	tns1:CdrCompletionStatus	Input parameter. Completion status of the CDR. See "CdrCompletionStatus".
spAccountId	xsd:string	Input parameter. ID of the Service Provider Account to filter the result on. Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter.
countCdrsReturn	xsd:long	Return parameter. The number of CDRs matching the given criteria.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

listCdrs

Retrieves all CDRs matching the given criteria.

Table 11–2 listCdrs(serviceName, fromDate, toDate, completionStatus, spAccountId, appAccountId, startIndex, maxEntries)

Parameter Name	Type	Description
serviceName	xsd:string	Input parameter. The name of the Service for which to retrieve CDRs. Use null to not filter on this parameter
fromDate	xsd:dateTime	Input parameter. From the date and time. Use null to not filter on this parameter
toDate	xsd:dateTime	Input parameter. To the date and time. Use null to not filter on this parameter.
completionStatus	tns1:CdrCompletionStatus	Input parameter. Completion status of the CDR.See "CdrCompletionStatus". Use null to not filter on this parameter.

Table 11–2 (Cont.) listCdrs(serviceName, fromDate, toDate, completionStatus, spAccountId, appAccountId, startIndex, maxEntries)

Parameter Name	Type	Description
spAccountId	xsd:string	Input parameter. ID of the Service Provider Account to filter the result on. Use null to not filter on this parameter.
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter.
startIndex	xsd:long	Input parameter. Which entry, in the overall result set, to start the result list on (cursor).
maxEntries	xsd:int	Input parameter. The maximum number of alarms returned.
listCdrsReturn	Array of tns1:CdrInfo	Return parameter. List of CDRS. See "CdrInfo".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

This exception is thrown by this interface.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These are complex data types used by this interface.

CdrInfo

Data structure defining a CDR. All services that produce charging data do not use all fields, and they use the fields slightly different, depending on the type of the service. See *Communication Service Guide* for details of which fields that are relevant for the different services.

Table 11–3 CdrInfo

Element name	Datatype	Description
transactionId	xsd:long	The Oracle Communications Services Gatekeeper transaction sequence number
serviceName	xsd:string	The communication service whose use is being tracked
timeStamp	xsd:dateTime	The time at which the event was triggered (in milliseconds from midnight 1 January 1970

Table 11–3 (Cont.) CdrInfo

Element name	Datatype	Description
origAddr	xsd:string	The address of the originating party.
destAddr	xsd:string	The address of the destination party.
spAccountId	xsd:string	The ID of the Service Provider that generated the CDR.
appAccountId	xsd:string	The ID of the Application Account that generated the CDR.
completionStatus	tns1:CdrCompletionStatus	Completion status of the CDR. See "CdrCompletionStatus".
info	xsd:string	Additional info provided by the service capability module.
additionalProperties	impl:ArrayOf_tns1_Property	Application defined data. See "Property".

CdrCompletionStatus

Defines the completion status of a CDR. Enumeration

Table 11–4 AlarmSeverity

Element name	Datatype	Description
COMPLETED	xsd:string	The operation generating the CDR succeeded.
FAILED	xsd:string	The operation generating the CDR failed
PARTIAL	xsd:string	The operation generating the CDR succeeded partially. May be supported, depending on the communication service.
COMPLETED_NOTIFICATION_FAILED	xsd:string	The CDR is completed, but the notification was not delivered to the application.
POLICY_DENIED	xsd:string	Policy denied the operation.

Property

Array of name-value pairs.

Table 11–5 Property

Element name	Datatype	Description
Name	xsd:string	Name of the property, with the value defined in Value. Unique within the array.
Value	xsd:string	The data associated with Name.

12

Operator Statistics Utility

The Operator Statistics Utility Web Service allows the Operator to retrieve the statistics generated in Oracle Communications Services Gatekeeper.

Interface: OpStatisticsUtil

The endpoint for this interface is: http://<host>:<port>/prm_op/services/OpStatisticsUtil

where the value for host and port depend on the Oracle Communications Services Gatekeeper deployment.

listStatisticTypes

Lists the statistics types registered in Oracle Communications Services Gatekeeper. This operation takes no input.

Table 12–1 listStatisticTypes()

Parameter Name	Type	Description
listStatisticTypesReturn	Array of tns1:StatisticTypeDescriptor	Return parameter. Descriptions of available statistics types. See "StatisticTypeDescriptor".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

getStatistics

Retrieve statistics matching the given criteria.

Table 12–2 getStatistics(statisticType, fromDate, toDate, spAccountId, appAccountId)

Parameter Name	Type	Description
statisticType	xsd:int	Input parameter. Number representing the type of statistics to retrieve.
fromDate	xsd:dateTime	Input parameter. From date and time. Use null to not filter on this parameter

Table 12–2 (Cont.) getStatistics(statisticType, fromDate, toDate, spAccountId,

Parameter Name	Type	Description
toDate	xsd:dateTime	Input parameter. To date and time. Use null to not filter on this parameter.
spAccountId	xsd:string	Input parameter. ID of the Service Provider Account to filter the result on. Use null to not filter on this parameter
appAccountId	xsd:string	Input parameter. ID of the Application Account to filter the result on. Use null to not filter on this parameter
getStatisticsReturn	Array of tns1:StatisticsInfo	Return parameter. Statistics. See "StatisticsInfo".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

This exception is thrown by this interface.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These complex data types are used by this interface.

StatisticsInfo

Data structure defining a statistics record. All services that produce statistics do not use all fields, and they use the fields slightly differently, depending on the type of the service.

Table 12–3 StatisticsInfo

Element name	Datatype	Description
statisticsType	xsd:string	The statistics type.
timeStampStart	xsd:dateTime	The starting time of the interval during which the statistics were gathered.
timeStampEnd	xsd:dateTime	The end time of the interval during which the statistics were gathered.
numberOfTransactions	xsd:int	The number of transactions during the interval.

Table 12–3 (Cont.) StatisticsInfo

Element name	Datatype	Description
spAccountId	xsd:string	The ID of the Service Provider that generated the statistics.
appAccountId	xsd:string	The ID of the Application Account that generated the statistics.

StatisticTypeDescriptor

Defines the type of statistics.

Table 12–4 AlarmSeverity

Element name	Datatype	Description
transactionTypeName	xsd:string	Name of a transaction type that statistics can be generated for.
transactionTypeId	xsd:int	Numeric ID of the transaction type.

13

Operator Alarm Utility

The Operator Alarm Utility Web Service allows the Operator to retrieve alarms from Oracle Communications Services Gatekeeper.

Interface: OpAlarmUtil

The endpoint for this interface is: http://<host>:<port>/prm_op/services/OpAlarmUtil

where the value of host and port depend on the Oracle Communications Services Gatekeeper deployment.

countAlarms

Counts the number of alarms of a certain type of a given severity for a specified time interval

Table 13–1 countAlarms(alarmId, severity, fromDate, toDate)

Parameter Name	Type	Description
alarmId	xsd:int	Input parameter. The ID of the type of alarm.
severity	tns1:AlarmSeverity	Input parameter. Severity of the alarm. See "AlarmSeverity". Use null to not filter on this parameter
fromDate	xsd:dateTime	Input parameter. Start date and time. Use null to not filter on this parameter.
toDate	xsd:dateTime	Input parameter. End date and time. Use null to not filter on this parameter.
countAlarmsReturn	xsd:long	Return parameter. The number of alarms.
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

listAlarms

Retrieves all alarms matching the given criteria

Table 13–2 listAlarms(alarmId, severity, fromDate, toDate, startIndex, maxEntries)

Parameter Name	Type	Description
alarmId	xsd:int	Input parameter. The ID of the type of alarm.
severity	tns1:AlarmSeverity	Input parameter. Severity of the alarm. See "AlarmSeverity". Use null to not filter on this parameter.
fromDate	xsd:dateTime	Input parameter. From date and time.Use null to not filter on this parameter.
toDate	xsd:dateTime	Input parameter. To date and time. Use null to not filter on this parameter.
startIndex	xsd:long	Input parameter. Which entry, in the overall result set, to start the result list on (cursor).
maxEntries	xsd:int	Input parameter. The maximum number of alarms returned.
listAlarmsReturn	Array of tns1:AlarmInfo	Return parameter. List of alarms. See "AlarmInfo".
ACCESS_DENIED	N/A	Exception
CommonException	N/A	Exception

Exceptions

This exception is thrown by this interface.

CommonException

This exception is raised when the login session has expired (BC only) or there are communication problems with the underlying platform.

Data types

These complex data types are used by this interface.

AlarmInfo

Data structure defining an alarm.

Table 13–3 AlarmInfo

Element name	Datatype	Description
alarmInstanceId	xsd:long	The ID of the emitted alarm. Unique identifier for an emitted alarm.
alarmId	xsd:int	The identifier for the alarm type.
source	xsd:string	Specifies the name of the software module that raised the alarm and the IP address of the server the service is installed in.

Table 13–3 (Cont.) AlarmInfo

Element name	Datatype	Description
severity	tns1:AlarmSeverity	Specifies the alarm's severity level. See "AlarmSeverity".
timeStamp	xsd:dateTime	Specifies the time and date the alarm was raised.
info	xsd:string	Alarm information provided by the software module the raised the alarm.
additional_info	xsd:string	Additional information depending on context.

AlarmSeverity

Defines the severity of an alarm. Enumeration.

Table 13–4 AlarmSeverity

Element name	Datatype	Description
WARNING	xsd:string	Severity level is Warning.
MINOR	xsd:string	Severity level is Minor.
MAJOR	xsd:string	Severity level is Major.
CRITICAL	xsd:string	Severity level is Critical.