# BEA WebLogic Network Gatekeeper™

## Managing Service Providers and Applications

Version 3.0 ™
Document Revised: 14 September 2007

# Copyright



# Restricted Rights Legend



# Trademarks and Service Marks

# Contents:


## Introduction and Document Roadmap

Document Scope and Audience . . . . . 1-1

Guide to this Document . . . . . 1-2

## Creating and maintaining service provider and application accounts

Typical workflow . . . . . 2-3

## Managing Service Provider Groups

Listing service provider groups . . . . . 3-1

Creating a service provider group . . . . . 3-2

Viewing information about a service provider group . . . . . 3-2

View data about the service provider group . . . . . 3-3

View SLA for the service provider group . . . . . 3-3

Get number of service provider accounts in a service provider group . . . . . 3-3

List service provider accounts associated with a service provider group . . . . . 3-3

Registering an updated SLA for a service provider group . . . . . 3-4

Deleting a service provider group . . . . . 3-4

## Managing Service Provider Accounts

Creating a service provider account . . . . . 4-1

Activating a service provider account . . . . . 4-2

Viewing information about a service provider account . . . . . 4-2

Get Account Data for a service provider account . . . . . 4-3
View current state of a service provider account . . . . . 4-3
View SLA for a service provider account . . . . . 4-3
List application accounts for a service provider . . . . . 4-3
Changing the SLA associated with a service provider account . . . . . 4-4
Logging out a service provider account . . . . . 4-4
Deactivating a service provider account . . . . . 4-4
Deleting a service provider account . . . . . 4-5

# Managing Application Groups

Listing application groups . . . . . 5-1
Creating an application group . . . . . 5-2
Viewing information about an application group . . . . . 5-2
View data about the application group . . . . . 5-2
View SLA file for the application group . . . . . 5-3
Get number of application accounts in an application group . . . . . 5-3
List application accounts associated with application group . . . . . 5-3
Registering an updated SLA for an application group . . . . . 5-3
Deleting an application group . . . . . 5-4

# Managing Application Accounts

Listing application accounts . . . . . 6-2
Finding an existing application account ID through the service provider account ID 6-2
Finding an existing application account ID through the application group ID . . . . 6-2
Creating an application account . . . . . 6-2
Activating an application account . . . . . 6-3
Viewing information about an application account . . . . . 6-4
View the application account data . . . . . 6-4

View the application account's state. . . . . 6-4
View the SLA for the application account . . . . . 6-4
Get number of application instance group IDs associated with an application . . . . 6-5
List application instance group IDs . . . . . 6-5
Changing the SLA associated with an application account. . . . . 6-5
Logging out an application account. . . . . 6-6
Deactivating an application account . . . . . 6-6
Deleting an application account. . . . . 6-6

# Managing Application Instance Group IDs

Listing application instance group IDs . . . . . 7-2
Listing logged in application instance group IDs . . . . . 7-2
Creating an application instance group ID . . . . . 7-3
Activating an application instance group ID (a username) . . . . . 7-3
Viewing information about an application instance group ID. . . . . 7-4
View application instance group account data . . . . . 7-4
View state . . . . . 7-4
View application instance group SLA . . . . . 7-4
Registering an updated SLA for an application instance group . . . . . 7-5
Logging out an application instance group ID . . . . . 7-5
Unlocking an application instance group . . . . . 7-5
Changing the password for an application instance group ID. . . . . 7-6
Deactivating an application instance group ID . . . . . 7-6
Deleting an application instance group . . . . . 7-6

# Provisioning specific for traffic paths

Backwards compatible Multimedia Messaging traffic paths. . . . . 8-1
Short Messaging traffic paths . . . . . 8-2

Presence traffic paths . . . . . 8-2
Backwards compatible Short Messaging traffic paths . . . . . 8-3
Reference: Provisioning for ESPA_messaging . . . . . 8-3

# Defining Service Provider Level and Application Level Service Agreements

Structure of a Service Level Agreement . . . . . 9-1
<Sla> . . . . . 9-3
<serviceContract> . . . . . 9-3
<startdate> . . . . . 9-3
<enddate> . . . . . 9-3
<scs> . . . . . 9-4
<contract> . . . . . 9-4
<overrides> . . . . . 9-4
<override> . . . . . 9-4
<enforceAcrossGeoSites> . . . . . 9-6
Contract structure . . . . . 9-6
<contract> . . . . . 9-8
<serviceCode> . . . . . 9-8
<guarantee> . . . . . 9-8
<methodRestrictions> . . . . . 9-9
<methodAccess> . . . . . 9-12
<params> . . . . . 9-12
<requestContext> . . . . . 9-13
SLA tags for individual traffic paths . . . . . 9-14
Parlay X 2.1 Third Party Call (BC) . . . . . 9-14
Backwards compatible Parlay X 2.1 Short Messaging and MultiMedia Messaging (BC) 9-14
Parlay X 2.1 Payment . . . . . 9-16

Extended Web Services WAP Push . . . . . 9-17
SLA attributes for individual traffic paths . . . . . 9-17

# Managing mailboxes

About Mailbox administration . . . . . 10-1
Mailboxes . . . . . 10-2
Creating mailboxes . . . . . 10-2
Deleting mailboxes by address . . . . . 10-2
Deleting mailboxes by owner (application) . . . . . 10-3
Message translation . . . . . 10-3
Adding a message translation . . . . . 10-3
Deleting a message translation . . . . . 10-4
Reference of provisioning attributes and operations for mailboxes . . . . . 10-4

# Mapping applications and service providers to OSA/Parlay accounts

Understanding OSA/Parlay account mappings . . . . . 11-1
Overall workflow when creating a OSA client mapping . . . . . 11-3
Information and certificate exchange with OSA/Parlay administrator . . . . . 11-3
Create OSA client . . . . . 11-4
Map the Network Gatekeeper account to an OSA/Parlay client . . . . . 11-5
Provisioning for specific traffic paths . . . . . 11-6
Reference: attributes and operations for Plugin_OSA_access . . . . . 11-6

# Reference: Provisioning operations for accounts and groups

Management operations related to service provider groups . . . . . 12-1
Management operations related to service provider accounts . . . . . 12-2
Management operations related to application accounts . . . . . 12-3
Management operations related to application groups . . . . . 12-3

Management operations related to application instance group IDs . . . . . 12-4
Reference: attributes and operations for ESPA_access. . . . . 12-5

# Reference: Provisioning operations for SESPA_access

Settings related to overall application characteristics . . . . . 13-1
Settings related to applications and traffic paths. . . . . 13-2
Reference: attributes and operations for SESPA_access . . . . . 13-2

CHAPTER 1

# Introduction and Document Roadmap

The following sections describe the audience for and organization of this document:


- Document Scope and Audience
- Guide to this Document


## Document Scope and Audience

This document describes how to perform service provider and application provisioning, consisting of:

- An overview of the administration model
- How to manage service provider groups
- How to manage service providers
- How to manage application groups
- How to manage application accounts
- How to manage application instance groups (account usernames)

The document will be of use to telecom operators, especially in relation to service provider and application provisioning. Managers, support engineers, and sales and marketing people will also find information of value here.

# Guide to this Document

The document contains the following chapters:

- Chapter 1, “Introduction and Document Roadmap”: This chapter
- Chapter 2, “Creating and maintaining service provider and application accounts”: An overview of the Network Gatekeeper service provider and application administration model, including a high level workflow
- Chapter 3, “Managing Service Provider Groups”: Step-instructions for managing service provider groups
- Chapter 4, “Managing Service Provider Accounts”: Step-instructions for managing service provider accounts
- Chapter 5, “Managing Application Groups”: Step-instructions for managing application groups
- Chapter 6, “Managing Application Accounts”: Step-instructions for managing application accounts
- Chapter 7, “Managing Application Instance Group IDs”: Step-instructions for managing application instance group IDs (login usernames)
- Chapter 8, “Provisioning specific for traffic paths”: Description of traffic path specific provisioning data that must be defined
- Chapter 9, “Defining Service Provider Level and Application Level Service Agreements”: Description of Service provider and Application SLAs
- Chapter 10, “Managing mailboxes”: Description of managing mailboxes
- Chapter 11, “Mapping applications and service providers to OSA/Parlay accounts”: Description of mapping applications and service providers to OSA/parlay accounts.
- Chapter 12, “Reference: Provisioning operations for accounts and groups”: Reference for management operations for service providers and applications.
- Chapter 13, “Reference: Provisioning operations for SESPA_access”: Reference for management operations for service providers and applications related to traffic paths.

CHAPTER 2

# Creating and maintaining service provider and application accounts

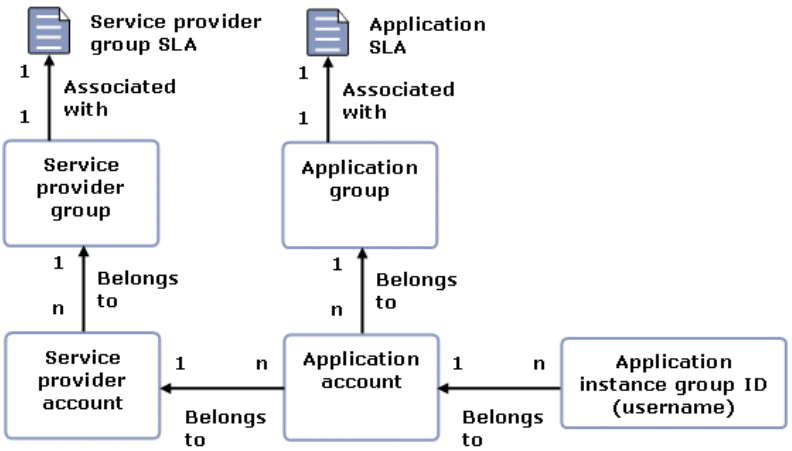
An essential part of managing WebLogic Network Gatekeeper is dealing with service provider partners. Network Gatekeeper provides a partner administration model to help operators handle the needs and demands of their partners in a flexible and powerful way:

- Each application that uses the application-facing interfaces in Network Gatekeeper must establish an account. The account is uniquely identified by an *application instance group account ID* (also known as an *account username*).
- All *application instance group accounts* belong to an *application account.* An application account can have zero or more associated application instance groups
- All *application accounts* belong to a *service provider account*. A service provider account can have zero or more associated application accounts.
- All *service provider accounts* are associated with a *service provider group*. The group is associated with a service level agreement on the service provider-level.
- All *application accounts* are associated with an *application group.* The group is associated with a service level agreement on the application-level.

The separation of accounts and groups provides a flexible way defining class-of-service.

See Figure 2-1 for more information on the relationship among account types.

**Figure 2-1 Service Provider and Application administration model**




Service provider groups allow the Network Gatekeeper administrator to categorize different types of service providers (via service provider SLAs). For example, a Network Gatekeeper administrator might create three different categories:

- Bronze
- Silver
- Gold

Each of these would have different privileges when it comes to maximum throughput and QoS parameters. When a new service provider is provisioned, the service provider is associated with an appropriate group based on, for example, the amount of network traffic the service provider is projected to generate. If the outcome is not what was predicted, the service provider account can be reassigned to another service provider group with a different QoS parameter defined in the service provider level SLA.

Application groups allow the Network Gatekeeper administrator to categorize different types of applications (via application level SLAs).

For example, a Network Gatekeeper administrator may create three different categories:

- Bronze

- Silver
- Gold

Each of these groups would have different privileges when it comes to maximum throughput and QoS parameters. When a new application account is provisioned, that account is associated with an appropriate group based on, for example, the amount of network traffic the application is projected to generate. If the outcome is not what was predicted, the application account can be reassigned to another application group with a different QoS parameter defined in the application level SLA.

Service provider level SLAs have precedence over application level SLAs.

The service provider and application account and group registrations are performed either internally through the WebLogic Network Gatekeeper Management Console or through external management systems integrated with WebLogic Network Gatekeeper using the Network Gatekeeper Partner Relationship Management Interfaces. This guide describes many of the tasks associated with establishing and maintaining these accounts and groups using the WebLogic Network Gatekeeper Management Console.

The service provider accounts and application instance groups have states, which means that they can be temporarily be taken out of service. The states can be changed via OAM. If a state is changed for a service provider account, it affects all applications associated with the service provider.

**Table 2-1 Possible states for a service provider account or an application account.**

| Value | State |
|---|---|
| 0 | Unknown |
| 1 | Activated, traffic is allowed. |
| 2 | Deactivated, traffic is not allowed. |
| 3 | Locked, locked due to too many failed login attempts. Only for application instance groups. |

# Typical workflow

1. To define the set of service classes, a corresponding set of SLAs is created, at both the service provider level and the application level, see Defining Service Provider Level and Application Level Service Agreements.

2. Create the service provider and application groups that correspond to these SLAs, see Managing Service Provider Groups and Managing Application Groups.
3. Create a service provider account and associate it with the appropriate service provider group. See Managing Service Provider Accounts.
4. Create an application account and associate it with the appropriate application provider group. See Managing Application Accounts.
5. Create an application instance group ID (the login username for the application). See Managing Application Instance Group IDs.
6. Depending on which traffic path is being used, provision traffic path specific data. See Provisioning specific for traffic paths.
7. Distribute the credentials to be used by the application to the service provider.

**Note:** It is also possible to create unique SLAs and groups for an individual application. Accounts can be associated with other groups in runtime, so the class of service can be adjusted over time.

CHAPTER 3

# Managing Service Provider Groups

The following sections describe how to administer service provider groups:

- Listing service provider groups
- Creating a service provider group
- Viewing information about a service provider group
  - View data about the service provider group
  - View SLA for the service provider group
  - Get number of service provider accounts in a service provider group
  - List service provider accounts associated with a service provider group
- Registering an updated SLA for a service provider group
- Deleting a service provider group

All management operations are performed using Network Gatekeeper Management console in the managed object ESPA_access.

## Listing service provider groups

1. Use Operation: listServiceProviderGroups. Based on Attribute: NumberOfServiceProviderGroups (r), which displays the number of registered service provider groups, choose a suitable **offset** (start listing) and **length** (how many to list) for the list, and click **Invoke**.

The selected range of existing service provider group IDs is displayed in alphabetic order

2. If necessary, browse the list by repeating the above steps with a different offset and length.

# Creating a service provider group

Follow the steps below to create a service provider group. The service provider group is associated with an SLA that regulates one or more service provider accounts.

A service provider group SLA must have been previously created, see Defining Service Provider Level and Application Level Service Agreements.

1. Choose a suitable (descriptive) ID for the service provider group according to your naming conventions.
2. Verify that the service provider ID is not already being used, see Listing service provider groups.
3. Use Operation: addServiceProviderGroupWithSLAUrl.

   **Note:** The Operation: addServiceProviderGroupWithSLAString can be used as an alternative.

   The result of the operation is displayed.

   The service provider group is now created.

   **Note:** To add service provider accounts to the service provider group, see Creating a service provider account.

# Viewing information about a service provider group

Follow the steps below to view information about a service provider group. The following types of information can be viewed independently of each other:

- View data about the service provider group
- View SLA for the service provider group
- Get number of service provider accounts in a service provider group
- List service provider accounts associated with a service provider group

## View data about the service provider group

To view all data about a service provider group:

1. Find the service provider group ID, see Listing service provider groups.
2. Use Operation: getServiceProviderGroup.

   The service provider group's SLA and OAM properties, if any, are displayed.

## View SLA for the service provider group

To view only the SLA:

1. Find the service provider group ID, see Listing service provider groups.
2. Use Operation: getSLAForServiceProviderGroup.

   The service provider group’s SLA is displayed.

## Get number of service provider accounts in a service provider group

To get the number of service provider accounts associated with a certain service provider group:

1. Find the service provider group ID, see Listing service provider groups.
2. Use Operation: getNumberOfServiceProvidersInGroup.

   The number of service provider accounts related to the service provider group is displayed.

## List service provider accounts associated with a service provider group

To get a list of the names of service provider accounts related to the service provider group:

1. Find out how many service provider accounts are associated with the service provider group, see Get number of service provider accounts in a service provider group.
2. Use Operation: listServiceProvidersInGroup.

   The selected range of service provider account IDs is displayed in alphabetic order.
3. If necessary, browse the list by repeating the above steps with a different offset and length.

# Registering an updated SLA for a service provider group

To register an updated SLA for a service provider group:

1. Find the service provider group ID, see Listing service provider groups.
2. View the current SLA, see View SLA for the service provider group.
3. Select Operation: updateServiceProviderGroupNodeSlaUrl.

   **Note:** The Operation: updateServiceProviderGroupNodeSlaString can be used as an alternative.

   The service provider group SLA is now updated and in use.

   **Note:** To add service provider accounts to the service provider group, see Creating a service provider account.

# Deleting a service provider group

To delete a service provider group:

**WARNING:** When deleting the service provider group, all related service provider accounts, application accounts and application instance group IDs will also be deleted.

1. Find the service provider group ID, see Listing service provider groups.
2. Select Operation: deleteServiceProviderGroup.

   The service provider group and all related service provider accounts, application accounts and application instance group IDs are now deleted.

CHAPTER 4

# Managing Service Provider Accounts

The following sections describe how to administer service provider accounts:

- Creating a service provider account
- Activating a service provider account
- Viewing information about a service provider account
  - Get Account Data for a service provider account
  - View current state of a service provider account
  - View SLA for a service provider account
  - List application accounts for a service provider
- Changing the SLA associated with a service provider account
- Logging out a service provider account
- Deactivating a service provider account
- Deleting a service provider account

All management operations are performed using Network Gatekeeper Management Console in the managed object ESPA_access.

## Creating a service provider account

To create a service provider account:

**Note:** The service provider group to which this service provider account is to be connected must be created before the service provider account is created. See Creating and maintaining service provider and application accounts for more information on the relationship.

1. Find an existing service provider group ID, see Listing service provider groups.
2. Choose a suitable (descriptive) ID for the service provider account according to your naming conventions.
3. Verify that the service provider account ID is not already being used, see List service provider accounts associated with a service provider group.
4. Use Operation: addServiceProviderAccount.

   The service provider account is now created.

   **Note:** To add application accounts to the service provider account, see Creating an application account.

   **Note:** The service provider account must be activated before the service provider's applications can access any services in the network, see Activating a service provider account. The account is activated when created.

# Activating a service provider account

**Note:** The service provider account must be activated before any of its applications can start accessing services in the network.

To activate a service provider account:

1. Find the service provider account ID, see List service provider accounts associated with a service provider group.
2. View current state of the service provider account, see View current state of a service provider account.
3. Use Operation: activateServiceProviderAccount.

   The service provider account's current state is now *Activated*.

# Viewing information about a service provider account

Follow the steps below to view information about a service provider account. The following types of information can be viewed independently of each other:

- Service provider account data, including related service provider group, Service Provider Reference, if any, and OAM properties, if any, see Get Account Data for a service provider account.
- Service provider account state, see View current state of a service provider account.
- The SLA registered to the service provider group to which the service provider account is assigned, see View SLA for a service provider account.
- The service provider accounts related to the service provider group, see List service provider accounts associated with a service provider group.

## Get Account Data for a service provider account

1. Find the service provider account ID, see List service provider accounts associated with a service provider group.
2. Use Operation: getServiceProviderAccount.

   The related service provider group ID, the Service Provider Reference, if any, and the account's OAM properties, if any, are displayed.

## View current state of a service provider account

1. Find the service provider account ID, see List service provider accounts associated with a service provider group.
2. Use Operation: getStateForServiceProviderAccount.

   The service provider account’s current state is displayed. See Table 2-1 for information about the state.

## View SLA for a service provider account

The SLA is correlated to the service provider group, see View SLA for the service provider group.

## List application accounts for a service provider

See List service provider accounts associated with a service provider group.

# Changing the SLA associated with a service provider account

To change the SLA associated with a service provider account, the account must be assigned to a different service provider group, see below.

(As an alternative, the SLA for the whole service provider group can be updated, see Registering an updated SLA for a service provider group.)

1. Find the new service provider group, see Listing service provider groups.
2. Use Operation: updateServiceProviderAccount from the **Select An Operation** drop-down list.

   The result of the operation is displayed.

# Logging out a service provider account

**WARNING:** This procedure logs out all applications related to this service provider account. To log out only the applications related to an application account, see Logging out an application account. To log out a single application instance group ID, see Logging out an application instance group ID.

To log out a service provider account, including all applications connected to the service provider account:

1. Find the service provider account ID, see List service provider accounts associated with a service provider group.
2. Use Operation: logoutServiceProviderAccount.

   All applications and application instance group IDs associated with the service provider account are logged out.

# Deactivating a service provider account

To deactivate, or temporarily stop the flow of traffic to and from, the service provider's applications:

1. Find the service provider account ID, see List service provider accounts associated with a service provider group.

2. View the service provider account's current state, see View current state of a service provider account.
3. Use Operation: deactivateServiceProviderAccount.

   The service provider account is deactivated.

   See Activating a service provider account for information on how to activate the account.

# Deleting a service provider account

**WARNING:** Deleting the service provider account also deletes all related application accounts and application instance group IDs.

To delete a service provider account:

1. Find the service provider account ID, by following the instructions in List service provider accounts associated with a service provider group.
2. View the service provider account's current state, see View current state of a service provider account.
3. Use Operation: deleteServiceProviderAccount.

   The service provider account and all applications and application instance group IDs are deleted.

CHAPTER 5

# Managing Application Groups

The following sections describe how to administer application groups:

- Listing application groups
- Creating an application group
- Viewing information about an application group
  - View data about the application group
  - View SLA file for the application group
  - Get number of application accounts in an application group
  - List application accounts associated with application group
- Registering an updated SLA for an application group
- Deleting an application group

All management operations are performed using Network Gatekeeper Management Console in the managed object ESPA_access.

## Listing application groups

1. Use Operation: listApplicationGroups. Based on Attribute: NumberOfApplicationGroups (r), which displays the number of registered application groups, choose a suitable **offset** (start listing) and **length** (how many to list) for the list, and click **Invoke**.

   The selected range of existing application group IDs are displayed in alphabetic order

2. If necessary, browse the list by repeating the above steps with a different offset and length.

# Creating an application group

The application group is associated with an SLA that can be used by one or more application accounts. An application group SLA must have been created, see Defining Service Provider Level and Application Level Service Agreements.

To create an application group:

1. Choose a suitable (descriptive) ID for the application group according to your naming conventions.
2. Verify that the application group ID is not already in use, see Listing application groups.
3. Use Operation: addApplicationGroupWithSLAUrl.

   The Operation: addApplicationGroupWithSLAString can be used as an alternative.

   The result of the operation is displayed.

   The application group is now created.

   **Note:** To add application accounts to the application group, see Creating an application account.

# Viewing information about an application group

Follow the steps below to view information about an application group. The following types of information can be viewed independently of each other:

- The application group data (SLA and OAM properties, if any), see View data about the application group.
- The application group SLA, see View SLA file for the application group.
- The number of application accounts associated with this group, see Get number of application accounts in an application group.
- The application accounts currently associated with this group, see List application accounts associated with application group.

## View data about the application group

To view all data about an application group:

1. Find the application group ID, see Listing application groups.
2. Use Operation: getApplicationGroup.

   The application group's SLA and OAM properties, if any, are displayed.

## View SLA file for the application group

To view only the SLA:

1. Find the application group ID, see Listing application groups.
2. Use Operation: getSLAForApplicationGroup.

   The application group's SLA is displayed.

## Get number of application accounts in an application group

To get the number of applications associated with a certain application group:

1. Find the application group ID, see Listing application groups.
2. Use Operation: getNumberOfApplicationsInApplicationGroup.

   The number of application accounts related to the service provider group is displayed.

## List application accounts associated with application group

To get a list of the names of application accounts related to the application group:

1. Find out how many application accounts are associated with the application group, see Get number of application accounts in an application group.
2. Use Operation: listApplicationAccountsInGroup. Base **offset** and **length** on the outcome of Get number of application accounts in an application group.

   The selected range application account IDs is displayed in alphabetic order.
3. If necessary, browse the list by repeating the above steps with a different offset and length.

# Registering an updated SLA for an application group

To register an updated SLA for an application group:

1. Find the application group ID, see Listing application groups

2. View current SLA for the application group, see View SLA file for the application group.
3. Use Operation: updateApplicationGroupSLAUrl.

   **Note:** The Operation: updateApplicationGroupSLAString can be used as an alternative.

   The application group SLA is now updated and in use.

   **Note:** To add application accounts to the application group, see Creating an application account.

# Deleting an application group

**WARNING:** Deleting an application group also deletes all related application accounts and application instance group IDs.

To delete an application group:

1. Find the application group ID, see Listing application groups
2. Use Operation: deleteApplicationGroup.

   The application group and its related application accounts and application instance group IDs are now deleted.

CHAPTER 6

# Managing Application Accounts

The following sections describe how to administer application accounts:

- Listing application accounts
  - Finding an existing application account ID through the service provider account ID
  - Finding an existing application account ID through the application group ID
- Creating an application account
- Activating an application account
- Viewing information about an application account
  - View the application account data
  - View the application account's state
  - View the SLA for the application account
  - Get number of application instance group IDs associated with an application
  - List application instance group IDs
- Changing the SLA associated with an application account
- Logging out an application account
- Deactivating an application account
- Deleting an application account

All management operations are performed using Network Gatekeeper Management console in the managed object ESPA_access.

# Listing application accounts

There are two ways to list application accounts:

- Through listing application accounts for a service provider account, see Finding an existing application account ID through the service provider account ID.
- Through listing application accounts related to a specific application group, see Finding an existing application account ID through the application group ID

## Finding an existing application account ID through the service provider account ID

To identify an application account ID through the service provider account ID:

1. Find the service provider account ID, see List application accounts for a service provider.

## Finding an existing application account ID through the application group ID

To find an application account ID through the application group ID:

1. Use Operation: getNumberOfApplicationsInApplicationGroup.
2. Use Operation: listApplicationAccountsInGroup. Based on the number of application accounts associated with the application group**,** choose a suitable **offset** (start listing) and **length** (how many to list) for the list.

   The selected range of application account IDs is displayed in alphabetic order.
3. If the range you have chosen does not include the application account ID you are checking for, repeat the process using a new range.

# Creating an application account

**Notes:** The service provider account and the application group to which this application account is to be connected must be created before the application account is created. See Creating

and maintaining service provider and application accounts for more information on the relationship.

To create an application account:

1. Find the ID of the service provider account to which the application account is to be assigned, see Listing service provider groups.
2. Find the ID of the application group to be used for the application account, see Listing application groups.
3. Choose a suitable (descriptive) ID for the application account according to your naming conventions.
4. Verify that the application account ID is not already in use, see Listing application accounts.
5. Use Operation: addApplicationAccount.

   The new application account is created.

**Notes:** To add application instance group IDs (login usernames) to the application account, see Creating an application instance group ID.

The application account must be activated before the application account's application instance group ID users can access any services in the network. See Activating an application account. The account is activated when it is created.

If the application is to access services in the network through an OSA/Parlay gateway, the application account must be connected to the OSA/Parlay gateway.

# Activating an application account

**Note:** The application account must be activated before the application can start accessing services in the network.

To activate an application account:

1. Find the application account ID, see Listing application accounts.
2. View the application account’s current state, see View the application account’s state.
3. Use Operation: activateApplicationAccount.

   The application account’s current state is now *Activated*.

# Viewing information about an application account

Follow the steps below to view information about an application account. The following types of information can be viewed independently of each other:

- The application account data (related application group ID; application reference, if any; OAM properties), see View the application account data.
- The application account state, see View the application account's state.
- The SLA related to the application account, see View the SLA for the application account.
- The number of application instance group IDs associated with an application account, see Get number of application instance group IDs associated with an application.
- The application instance group IDs related to the application account, List application instance group IDs.

## View the application account data

1. Find the application account ID, see Listing application accounts.
2. Use Operation: getApplicationAccount.

   The related application group's ID, the application reference, if any, and the account's OAM properties, if any, are displayed.

## View the application account's state

1. Find the application account ID, see Listing application accounts.
2. Use Operation: getStateForApplicationAccount.

   The application account's current state is displayed. See Table 2-1 for information about the state.

## View the SLA for the application account

The SLA is associated with the application group. To view the SLA, see View SLA file for the application group.

## Get number of application instance group IDs associated with an application

To get the number of application instance group IDs (login usernames) associated with a certain application:

1. Find the application account, see List application accounts for a service provider.
2. Operation: getNumberOfApplicationInstanceGroupsInApplications.

   The number of application instance group IDs assigned to the application account is displayed.

## List application instance group IDs

To list application instance group IDs:

1. Find the application account, see List application accounts for a service provider.
2. Get the number of application instance group IDs associated with the application account, see Get number of application instance group IDs associated with an application.
3. Use Operation: listApplicationInstanceGroups. Based on the number of application instance group IDs associated with the application account, choose a suitable **offset** (start listing) and **length** (how many to list) for the list.

   The selected range of application instance group IDs is displayed in alphabetic order.
4. If necessary, browse the list by repeating the above steps with a different offset and length

# Changing the SLA associated with an application account

To change the SLA associated with an application account, that account must be reassigned to a different application group:

(As an alternative, the SLA for the whole group can be updated, see Registering an updated SLA for an application group)

1. Examine which SLA suits the application by repeating the procedure described in View SLA file for the application group or create a new group, see Creating an application group.
2. Use Operation: updateApplicationAccount.

   The application account is reassigned to the new application group and its associated SLA.

# Logging out an application account

**WARNING:** This procedure logs out all application instance group ID users associated with this application account. To log out only a single application instance group ID (a single login username), see Logging out an application instance group.

To log out an application account:

1. Find the application account ID, See Listing application accounts.
2. Use Operation: logoutApplicationAccount.

   All application instance group ID users related to the application account are logged out.

# Deactivating an application account

To deactivate, or temporarily stop the flow of traffic to and from, this account's applications:

1. Find the application account ID, see Listing application accounts.
2. Use Operation: deactivateApplicationAccount.

   The application account’s state is changed to deactivated.

# Deleting an application account

**WARNING:** Deleting an application account also deletes all the application instance group IDs assigned to that account.

To delete an application account:

1. Find the application account ID, see Listing application accounts.
2. Use Operation: deleteApplicationAccount.

   The application account and all related application instance group IDs are deleted.

CHAPTER 7

# Managing Application Instance Group IDs

The following sections describe how to administer application instance group IDs:

- Listing application instance group IDs
- Listing logged in application instance group IDs
- Creating an application instance group ID
- Activating an application instance group ID (a username)
- Viewing information about an application instance group ID
  - View application instance group account data
  - View state
  - View application instance group SLA
- Registering an updated SLA for an application instance group
- Logging out an application instance group ID
- Unlocking an application instance group
- Changing the password for an application instance group ID
- Deactivating an application instance group ID
- Deleting an application instance group

All management operations are performed using Network Gatekeeper Management console in the managed object ESPA_access

# Listing application instance group IDs

To list application instance group IDs:

1. Find the service provider group, see Listing service provider groups.
2. Find the service provider account ID, see List service provider accounts associated with a service provider group.
3. Find the application account ID, see Finding an existing application account ID through the service provider account ID.
4. Use Operation: getNumberOfApplicationsInApplicationGroup.

   The number of application instance group IDs assigned to the application account is displayed.
5. Use Operation: listApplicationInstanceGroups. Based on the number of application instance group IDs associated with the service provider and application account combination, choose a suitable **offset** (start listing) and **length** (how many to list) for the list.

   The selected range of application instance group IDs is displayed in alphabetic order.

# Listing logged in application instance group IDs

To list currently logged in application instance group IDs:

1. Find the service provider group, see Listing service provider groups.
2. Find the service provider account ID, see List service provider accounts associated with a service provider group.
3. Find the application account ID, see Finding an existing application account ID through the service provider account ID.
4. Use Operation: getNumberOfLoggedInApplicationInstanceGroups.

   The number of logged in application instance group IDs is displayed.
5. Use Operation: listLoggedInApplicationInstanceGroups. Based on the number of logged in application instance group IDs, choose a suitable **offset** (start listing) and **length** (how many to list) for the list.

The selected range of logged in application instance group IDs is displayed in alphabetic order.

# Creating an application instance group ID

Follow the steps below to create an application instance group. An application instance group ID serves as the login username the application. It must be unique. For more information, see Creating and maintaining service provider and application accounts.

Additional tasks may need to be performed, depending on the traffic path the requests will be using, see Mapping applications and service providers to OSA/Parlay accounts and Provisioning specific for traffic paths.

1. Find the service provider group, see Listing service provider groups.
2. Find the service provider account ID, see List service provider accounts associated with a service provider group.
3. Find the service provider account ID, see Finding an existing application account ID through the service provider account ID.
4. Choose a suitable (descriptive) ID for the application instance group ID according to your naming conventions. It must
5. List existing application instance group IDs to make sure the application instance group ID is not already in use, see Listing application instance group IDs.
6. Use Operation: addApplicationInstanceGroupSLAString.

   The new application instance group ID is created. The instance group account ID must be activated before the application account's application instance group IDs users can access any services in the network. The application instance group is activated when created. See Activating an application instance group ID (a username).

# Activating an application instance group ID (a username)

To activate an application instance group ID:

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: activateApplicationInstanceGroup.

   The application instance group ID’s state is changed to activated. See Table 2-1 for information about the state.

# Viewing information about an application instance group ID

Follow the steps below to view information about an application instance group ID. The following types of information can be viewed independently of each other:

- The application account data (related application group, application instance group ID reference, OAM properties), see View application instance group account data.
- Application instance group ID's account state, View state.
- Application instance group's SLA - only listed for completeness.

## View application instance group account data

To view the application instance group ID account data (related application group, application instance group reference, OAM properties):

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: getApplicationInstanceGroup.

   The application instance group ID's OAM properties, if any, and application instance group ID's reference are displayed.

## View state

To view the application instance group's state:

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: getStateForApplicationInstanceGroup.

   The application instance group's current state is displayed in the Messages. See Table 2-1 for information about the state.

## View application instance group SLA

To view the application instance group's SLA:

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: getSLAForApplicationInstanceGroup ID.

The parameters for the operation are displayed.

The SLA contents, indicating number of allowed concurrently logged in application instances, is displayed. Is always 1, since only one instance is allowed.

# Registering an updated SLA for an application instance group

Deprecated because always set to 1. To register an updated the SLA for an application instance group.

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: updateApplicationInstanceGroupSLAString.

   The application instance group ID's SLA is now updated and in use.

# Logging out an application instance group ID

To log out the application logged in with the username related to the application instance group ID.

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: logoutApplicationInstanceGroup.

   The application with the application instance group ID is logged out.

# Unlocking an application instance group

To unlock an application instance group ID that has been locked because of too many failed login attempts.

**Note:** When an application instance group ID is unlocked, its state is changed to activated.

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: unlockApplicationInstanceGroup.

   The application instance group ID is unlocked and its state is changed to activated.

# Changing the password for an application instance group ID

To change password for an application instance group ID (a login username).

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: setPasswordForApplicationInstanceGroup.

   The application instance group's password is changed.

# Deactivating an application instance group ID

To deactivate, or temporarily stop the flow of traffic to and from, this application instance group ID:

1. Find the application instance group ID, see Listing application instance group IDs.
2. Use Operation: deactivateApplicationInstanceGroup.

   The application instance group ID's state is changed to deactivated. See Table 2-1 for information about the state.

# Deleting an application instance group

To delete an application instance group ID:

**Note:** If any mailboxes, charging accounts and/or Parlay X settings have been created for this application instance group ID (see Managing mailboxes), they must be removed before removing the application instance group ID. This is only relevant when using backwards compatible traffic paths for messaging.

1. Find the application instance group ID, see Listing application instance group IDs.
2. Select Operation: deleteApplicationInstanceGroup.

   The application instance group ID is now deleted.

CHAPTER 8

# Provisioning specific for traffic paths

Below is a list of traffic-patch specific provisioning that needs to be defined per traffic path:

- Backwards compatible Multimedia Messaging traffic paths
- Short Messaging traffic paths
- Presence traffic paths
- Backwards compatible Short Messaging traffic paths

## Backwards compatible Multimedia Messaging traffic paths

The following needs to be provisioned for backwards compatible Multimedia Messaging traffic paths:

1. Mailboxes needs to be created. See Managing mailboxes.
2. Define how parameter senderAddress in operation SendMessage shall be used by an application:
   a. If the application shall use the format

      `tel:<mailbox ID>\<mailbox password>\tel:<originator address>`

      No additional step is necessary.

   b. If the application shall use the format

`SenderName`

application properties have to be defined. Use Operation: setAppProperties in SESPA_access to define mappings between senderName and a mailbox.

3. If polling for mobile originating messages shall be supported:

   As parameter registrationIdentifier in operation getReceivedMessages, applications must use the mailbox ID. The format of this parameter is tel:<mailboxID>\<mailbox password>.

4. If starting of notifications shall be supported, define how parameter messageServiceActivationNumber in operation startMessageNotification shall be used by an application:

   a. If the application shall use the format:

      tel:<mailboxID>;mboxPwd=<mailbox password> (for example `tel:123;mboxPwd=123`)

      No additional step is necessary.

   b. If the application shall use the format:

      mailboxID (for example `tel:123`)

      application properties have to be defined. Use Operation: setAppProperties in SESPA_access to define the mailbox password.

# Short Messaging traffic paths

The following needs to be provisioned for enhanced Short Messaging traffic paths:

For SMPP and OSA Multimedia Messaging plug-in for Short Messaging the following can be provisioned:

- Off-line registration for notifications.
- Adds an offline notification for applications that will poll for mobile originated short messages.
- If short code translations shall be used, define these.

See Managing and Configuring Short Messaging Traffic Paths.

# Presence traffic paths

The following needs to be provisioned for Presence traffic paths:

For SIP integration for the Parlay X 2.1 Presence traffic path, a mapping between the SIP URI and an application Instance Group must be defined, using the operation setApplicationInstanceGroupSIPURI in Plugin_presence_sip.

See Managing and Configuring the Presence Traffic Path.

# Backwards compatible Short Messaging traffic paths

The following needs to be provisioned for backwards compatible Short Messaging traffic paths:

1. Mailboxes needs to be created. See Managing mailboxes.
2. Define how parameter senderAddress in operation SendSms shall be used by an application:
   a. If the application shall use the format

      `tel:<mailbox ID>\<mailbox password>\tel:<originator address>`

      No additional step is necessary.

   b. If the application shall use the format

      `senderName`

      application properties have to be defined. Use Operation: setAppProperties in SESPA_access to define mappings between sendername and the application.
3. If polling for mobile originating messages shall be supported:

   As parameter registrationIdentifier in operation getReceivedSms, applications must use the mailbox ID. The format of this parameter is tel:<mailboxID>.
4. If starting of notifications shall be supported:

   As parameter smServiceActivationNumber in operation startSmsNotificationRequest, the format of this parameter must be tel:<mailboxID>.

# Reference: Provisioning for ESPA_messaging

- Operation: listNotifications

## Operation: listNotifications

Lists enabled notifications, with details about each notification, for a given service provider account and application account.

```
listNotifications(   applicationId:String,
                     serviceProviderId:String,
                     startIndex:int,
                     maxCount:int)
```

**Table 8-1 listNotifications**

| listNotifications | |
|---|---|
| **Parameters** | **Description** |
| applicationId | The application account ID. |
| serviceProviderId | The service provider account ID. |
| startIndex | The start index of the first notification in the matching list. |
| maxCount | Maximum number of entries reported.<br>Use 0 to list all entries. |

CHAPTER 9

# Defining Service Provider Level and Application Level Service Agreements

An application's access rights to BEA WebLogic Network Gatekeeper are specified in Service Level Agreement (SLA) XML files. There is an SLA associated with the service provider group and one associated with the application group. For more information on this administration model, see Creating and maintaining service provider and application accounts.

If there is a conflict of values between the service provider SLA and the application SLA, the most restrictive value always applies.

- Structure of a Service Level Agreement
- Contract structure
- SLA tags for individual traffic paths
- SLA attributes for individual traffic paths

## Structure of a Service Level Agreement

The xsds for the SLAs are found in directory $DOMAIN_HOME/policy/sla_schema. The application group SLA xsd is app_sla_file.xsd and the service provider group SLA is sp_sla_file.xsd.

The SLA contains two main types of information specified by the attributes in the <Sla> tag and the contents of the <serviceContract> tags. Listing D-1 Service level agreement XML file overview shows the service provider level SLA XML file's main structure and the relation between the <Sla> and the <serviceContract> tags. Differences between the SLA files on service provider and application level are described in the tag descriptions below the listing.

**WARNING:** If your SLA XML file has spaces before the <?xml...> tag, the SLA will not load.

**Listing 9-1 Listing D-1 Service level agreement XML file overview**

```
<Sla [serviceProviderGroupID | applicationGroupID] ="spGroup1"
        xmlns:xsi=
        "http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation=
        "file:./policy/sla_schema/sp_sla_file.xsd">
        <serviceContract>
                <startDate></startDate>
                <endDate></endDate>
                <scs></scs>
                <contract>
                        ....
                </contract>
                <overrides>
                        <override>
                                <startDate></startDate>
                                <endDate></endDate>
                                <startDow></startDow>
                                <endDow></endDow>
                                <startTime></startTime>
                                <endTime></endTime>
                                <contract>
                                        ...
                                </contract>
                        </override>
```

```
            </overrides>
            <enforceAcrossGeoSites></enforceAcrossGeoSites>
        </serviceContract>
</Sla>
```

# <Sla>

This tag contains a number of service contracts specifying under which conditions a service provider or an application is allowed to access and use service capabilities.

The **serviceProviderGroupID** attribute specifies service provider group the service provider belongs to and for which the SLA is valid. For SLAs on application level the **applicationGroupID** attribute is used instead. It specifies the application group the application belongs to and for which the SLA is valid.

The **xmlns:xsi** and **xsi:noNamespaceSchemaLocation** attributes contains processing information and should not be changed.

# <serviceContract>

This tag contains contractual data specifying under which conditions a service provider or an application is allowed to access and use specific services in Network Gatekeeper. One <**serviceContract**> tag is needed for each service capability a service provider and application shall have access to.

The data to be defined in the <**serviceContract**> tag for each service capability is described below.

# <startdate>

This tag specifies the date the application can start using the service capability. Use format YYYY-MM-DD.

**Note:** A later start date on the service provider level service contract overrides this date.

# <enddate>

This tag specifies the last date the application can use the service capability. Use format YYYY-MM-DD.

**Note:** A earlier end date on the service provider level service contract overrides this date.

## <scs>

This tag specifies the name of the service capability. Each traffic path has different identifiers, see Network Gatekeeper Traffic Path Reference.

## <contract>

This tag contains all contractual data.

The <**contract**> tag directly following the <**scs**> tag contains the default restrictions.

This tag, including the sub tags, can also be defined within the <override> tag in order to override the default restrictions, see <overrides> and <overrides>.

## <overrides>

This tag is a container of one or more <**override**> tags.

## <override>

The contract data specified within this tag is used for overriding the default contractual data given in the <**contract**> tag directly following the <**scs**> tag.

When an override occurs, only the restrictions given in the <**override**> section are used. That is, all restrictions that are present in the normal contract will be disregarded if not explicitly restated in the override section. No quota definitions are valid within the override section.

In each occurrence of this tag, the following sub tags can be used:

- <**startDate**>, specifies the start date the contract data given within the <**override**> tag is valid. Use format YYYY-MM-DD. Optional tag. If omitted, the start date for the service contract is used.
- <**endDate**>, specifies the end date the contract data given within the <**override**> tag is valid. Use format YYYY-MM-DD. If omitted, the end date for the service contract is used.
- <**startDow**>, specifies the starting weekday for which the contract data given within the <**override**> tag is valid. Use 1 for Sunday, 2 for Monday and so on. Optional tag.
- <**endDow**>, specifies the end weekday for which the contract data given within the <**override**> tag is valid. Use 1 for Sunday, 2 for Monday and so on. Optional tag if <**startDow**> is not given.

- <**startTime**>, specifies the start time for which the contract data given within the <**override**> tag is valid. Use format hh:mm:ss, where hh can be 00-24. Optional tag.
- <**endTime**>, specifies the end time for which the contract data given within the <override> tag is valid. Use format hh:mm:ss, where hh can be 00-24. Optional tag if <startTime> is not given.
- <**contract**>, specifies the contract data that overrides the contract data given directly after the <**scs**> tag. The tags that can be defined within the <**contract**> tag are described in the sections describing the contract data for each service capability.

**Note:** For an override to be active all of the following must be true:

- Today's date must be the same as or later than startDate
- Todays' date must be earlier than endDate (must not be the same date)
- Current time must be between startTime and endTime. endTime being earlier than **startTime** means the limit spans midnight.
- Current day of week must be between startDow and endDow or equal to startDow or endDow. endDow being less than startDow means the limit spans the week end.

Since several <**override**> tags can be defined, with different settings for the time periods for which the restrictions applies, make sure the time periods do not overlap. In the case of overlapping time periods, there is no guarantee which contract that applies.

Below are two examples of <**override**> sections.

**Listing 9-2 Listing D-2 <override> example**

```
<overrides>
        <override>
                <startDate>2008-11-01</startDate>
                <endDate>2008-11-30</endDate>
                <startDow>2</startDow>
                <endDow>2</endDow>
                <startTime>09:00:00</startTime>
                <endTime>10:00:00</endTime>
```

```
            <contract>
                  ...
            </contract>
      </override>
      <override>
            <startDate>2008-12-01</startDate>
            <endDate>2008-12-30</endDate>
            <startDow>2</startDow>
            <endDow>2</endDow>
            <contract>
                  ...
            </contract>
      </override>
</overrides>
```

## <enforceAcrossGeoSites>

This tag specifies is the SLA shall be enforced across geo-redundant sites.

Specify **<enforceAcrossGeoSites>**true**<enforceAcrossGeoSites>** if the SLA shall be enforced across geo-redundant site, otherwise **false**.

Optional tag. If not present, default is false.

**Note:** If set to **true**, the **<override>** tag is ignored.

# Contract structure

The listing below illustrates the structure of the <contract> segment of an SLA.

**Listing 9-3 Contract structure**

```
<contract>
        <serviceCode></serviceCode>
        <guarantee>
                <methodGuarantee>
                        <methodNameGuarantee></methodNameGuarantee>
                        <reqLimitGuarantee></reqLimitGuarantee>
                        <timePeriodGuarantee></timePeriodGuarantee>
                </methodGuarantee>
        </guarantee>
        <methodRestrictions>
                <methodRestriction>
                        <methodName></methodName>
                        <rate>
                                <reqLimit></reqLimit>
                                <timePeriod></timePeriod>
                        </rate>
                        <quota>
                                <qtaLimit></qtaLimit>
                                <days></days>
                                <limitExceedOK></limitExceedOK>
                        </quota>
                </methodRestriction>
        </methodRestrictions>
        <methodAccess>
                <blacklistedMethod>
                        <methodName></methodName>
```

```
            </blacklistedMethod>
        </methodAccess>
</contract>
```

## <contract>

This tag serves to hold together all service-specific data. A <**contract**> tag occurs once for every <**scs**> and once for every <**override**>.

## <serviceCode>

Optional. Application level SLAs only.

This tag is used to specify a service code that can be used for charging purposes. A service code specified by the application in the application-facing interfaces will be replaced with this code.

## <guarantee>

Optional.

This tag is used to specify a number of method requests the service provider or application is guaranteed during a specified time period (in milliseconds). Method requests from service providers and applications having the method tagged as guaranteed will have precedence above requests from service providers and applications not having the method tagged as guaranteed. Requests are given high priority if the request rate is below either the request rate stated in the service provider level SLA or the application level SLA, whichever has the higher request rate. For example, if the service provider level SLA guarantees 30 requests per second and the application level SLA guarantees 40 requests per second, the application level SLA applies.

The <**guarantee**> tag must encapsulate the <**methodNameGuarantee**>, <**reqLimit**>, and <**timePeriod**> tags within a **<methodGuarantee>** tag.

**Note:** The **<methodGuarantee>** tag is ignored for mobile-originated traffic.

Each method in <**methodNameGuarantee**> must also be defined in <**methodRestrictions**>. The time period defined for a certain method must be identical in both the <**guarantee**> tag and the <**methodRestriction**>. See <methodRestriction>.

### <methodNameGuarantee>

The name of the method to have guaranteed precedence. Each traffic path has different identifiers, see Network Gatekeeper Traffic Path Reference.

### <reqLimit>

The number of requests to be guaranteed over the time period given in <**timeperiod**>.

### <timePeriod>

The timeperiod to apply the guarantee, given in milliseconds.

## <methodRestrictions>

Optional.

This tag is used for restricting the number of method requests an application is allowed to do over a short time period, the short time period is referred to as a *rate*, typically spanning a few seconds. A longer time period, referred to as a quota, typically spans several days.

**Note:** The <**methodRestrictions**> tag is ignored for mobile-originated traffic.

One <**methodRestrictions**> tag, including one or more <**methodRestriction**> tags, is needed for each method that should have restricted usage. Only one <**methodRestrictions**> tag is allowed.

Below is an example with usage restrictions both for quotas and rates.

**Listing 9-4 Example of a usage restriction**

```
<methodRestrictions>
        <methodRestriction>
                <methodName>putMessage</methodName>
                <rate>
                        <reqLimit>5</reqLimit>
                        <timePeriod>1000</timePeriod>
                </rate>
                <quota>
```

```
                <qtaLimit>600</qtaLimit>
                <days>3</days>
                <limitExceedOK>true</limitExceedOK>
        </quota>
    </methodRestriction>
    <methodRestriction>
        <methodName>putMmMessage</methodName>
        <rate>
                <reqLimit>5</reqLimit>
                <timePeriod>1000</timePeriod>
        </rate>
        <quota>
                <qtaLimit>600</qtaLimit>
                <days>3</days>
                <limitExceedOK>true</limitExceedOK>
        </quota>
    </methodRestriction>
</methodRestrictions>
```

The example above specifies the usage restrictions for the methods putMessage and putMmMessage. The usage restriction for the short time period is 5 requests per second (5 requests divided by 1000 milliseconds). The usage restriction for the longer time period, the quota, is 600 requests over a 3 day period.

If the application does not have any usage restrictions within the allowed methods, the whole <**restriction**> tag should be deleted or commented out.

The most restrictive limit is always enforced, so if a restriction in a service provider level SLA is more restrictive than a restriction defined in an application level SLA, the service provider level SLA is the one being enforced.

## <methodRestriction>

Contains restrictions for a method in the application-facing interface. Encapsulates the tags:

- <**methodName**>
- <**rate**>
- <**quota**>

## <methodName>

The name of the method to have restrictions on. Each traffic path has different identifiers, see Network Gatekeeper Traffic Path Reference.

## <rate>

This tag defines the maximum number of requests to be guaranteed over a short time period, *rate*.

The <**rate**> tag contains two sub-tags:

- <**reqLimit**> defines the maximum number of requests over the timeperiod given in <**timePeriod**>
- <**timePeriod**> defines the time period, given in milliseconds.

## <quota>

This tag defines the maximum number of requests over a longer time period.

The counters associated with the quota are reset at the beginning of each time period.

The <**quota**> tag contains the tags:

- <**qtaLimit**> defines the maximum number of requests over the timeperiod defined in <**qtaLimit**>.
- <**days**> defines the timeperiod period, given in days (only integers are valid). The starting day (day 0) is the same day as the <**startdate**> for the service contract. Only one quota limit and time period can be specified per <**quota**> tag.
- <**limitExceedOK**> is used to specify what action to take if the quota limit is exceeded. If it is defined as true, the request will be allowed even if the quota is exceeded. If it is defined as false, the request will be rejected. An alarm is always emitted if the limit is exceeded.

## <methodAccess>

Optional.

This tag is used to block the application from accessing one or more methods in the service capability. If the application is allowed to access all methods, the <**methodAccess**> tag should be omitted.

This tag encapsulates one or more <**blacklistedMethod**> tags.

## <params>

This tag is used to either allow or deny certain parameters values to be provided by applications.

The <**params**> tag can encapsulate one or more <**methodParameters**> tags, each containing a <**methodName**>, <**parameterName**>, <**parameterValues**> and <**acceptValues**> tag.

In the example below, the parameter Content-type is allowed to contain only the parameter values text/application and text/plain for the method sendMessageReq. No other values for this parameter are allowed.

**Listing 9-5 Example of <params>**

```
<params>
        <methodParameters>
                <methodName>sendMessageReq</methodName>
                <parameterName>Content-type</parameterName>
                <parameterValues>text/application text/plain</parameterValues>
                <acceptValues>true</acceptValues>
        </methodParameters>
</params>
```

## <methodParameters>

Encapsulates the <**methodName**>, <**parameterName**>, <**ParameterValue**> and <**acceptValues**> tag.

- <**methodName**> contains name of the method whose parameters are to be allowed or denied.
- <**parameterName**> contains the name of the parameter whose values are to be defined as allowed or denied.
- <**parameterValue**> tag contains a list of allowed values. Several values can be defined, separated by white space.
- <**acceptValues**> tag defines if the parameter values shall be allowed or denied. If the tag contains true, the parameter values given are allowed and all other values are denied. If the tag contains false, the parameter values given are denied and all other values are allowed.

# <requestContext>

Encapsulates one or more <**contextAttribute**> tags.

A <**contextAttribute**> tag contains one <**attributeName**> and one <**attributeValue**> tag.

A plug-in can retrieve the value specified in <**attributeValue**> using the name specified in <**attributeName**>.

**Note:** The plug-in must implement the functionality for fetching the value by name, see *Extension Toolkit - Developer's Guide* for more information.

**Listing 9-6 Example of <requestContext>**

```
<requestContext>
  <contextAttribute>
    <attributeName>com.bea.wlcp.wlng.plugin.sms.testName1</attributeName>
    <attributeValue>testValue1</attributeValue>
  </contextAttribute>
  <contextAttribute>
    <attributeName>com.bea.wlcp.wlng.plugin.sms.testName2</attributeName>
    <attributeValue>testValue2</attributeValue>
  </contextAttribute>
</requestContext>
```

# SLA tags for individual traffic paths

## Parlay X 2.1 Third Party Call (BC)

### <maxNoOfActiveCalls>

This tag is used to specify the maximum number of active calls the application is allowed to have. If there is no restriction, the tag is deleted or commented out.

### <maxNoOfCallLegsInCall>

This tag is used to specify the maximum number of call legs in a call the application is allowed to have. If there is no restriction, the tag should be deleted or commented out.

## Backwards compatible Parlay X 2.1 Short Messaging and MultiMedia Messaging (BC)

### <persistMessage>

This tag is used to define if incoming, mobile originated messages are to be stored in the mailbox.

The tag contains a boolean value; true or false.

If **true**, incoming messages are stored in the mailbox and applications can poll for and list messages stored in the mailbox. When a message arrives to the inbox, an acknowledgement will be sent to the network that the message has been delivered.

If **false**, incoming messages are not stored in the mailbox. Applications will not be able to poll for messages. When a messages is delivered to an application, an acknowledgement will be sent to the network that the message has been delivered.

The tag is optional, and if omitted the behavior is the same as defining the tag value as false. For performance reasons it is better to omit the tag than to set it to false.

The tag is valid for application level SLAs only.

## <allowedCharging>

This tag is used for specifying the allowed charging amounts for SMSes. The amounts are specified as a string with spaces between the allowed amounts.

## <maxMessageSize>

This tag is used for specifying the maximum number of characters allowed in a SMS message. This size can be larger than 160.

## <allowedMmCharging>

This tag is used for specifying the allowed charging amounts for MMSes. Specified as a string with spaces between the allowed amounts.

## <maxMmMessageSize>

This tag is used for specifying the maximum size of a MMS message in bytes.

## <allowedContentTypes>

This tag is used for specifying which content types are allowed in MMS messages. The allowed content types are specified as a string with spaces between the values representing the content types. The content types are specified as MIME types (Type/Subtype), for example `image/gif`.

## <allowedEncodingTypes>

This tag is used for specifying which content types are allowed in SMS messages sent from an application. The allowed content types are specified as a string with spaces between the values representing the content types.

The content types are specified as:

- `NORMAL` for text SMSs.
- `RINGTONE|SM` for ringtones in SmartMessaging format.
- `RINGTONE|EMS` for ringtones in EMS format.
- `LOGO|SM` for logos in SmartMessaging format.
- `LOGO|EMS` for logos in EMS format.

# Parlay X 2.1 Payment

## <currencyRestriction>

This tag is used to specify the currencies allowed in the transactions handled by the charging service. The currency code is specified according to the ISO 4217 standard. To avoid too small and too large amounts in transactions, it is possible to specify a maximum and a minimum amount for each currency.

One <allowedCurrency> tag (including the <currencyCode>, <minAmount>, and <maxAmount> tags) is needed for each allowed currency. If all currencies are allowed, the tag is deleted or commented out.

**Listing 9-7 Example of usage of <currencyRestriction>**

```
<currencyRestriction> <!-- Optional -->
  <allowedCurrency>
    <currencyCode>USD</currencyCode>
    <minAmount>1</minAmount>
    <maxAmount>10</maxAmount>
  </allowedCurrency>
  <allowedCurrency>
   <currencyCode>EUR</currencyCode>
   <minAmount>1</minAmount>
   <maxAmount>10</maxAmount>
   </allowedCurrency>
</currencyRestriction>
```

## Extended Web Services WAP Push

### <maxMessageSize>

This tag is used for specifying the maximum number of bytes allowed in a message.

### <allowedContentTypes>

This tag is used for specifying which content types are allowed in a message. The allowed content types are specified as a string with spaces between the values representing the content types.

The content types are specified as MIME types (Type/Subtype), for example `image/gif`.

For MIME multipart messages, the individual body parts are not evaluated. In order to allow multipart messages, multipart/mixed and/or multipart/related should be included in this tag.

# SLA attributes for individual traffic paths

For information about SLA attributes to set for the individual traffic paths, see Network Gatekeeper Traffic Path Reference.

CHAPTER 10

# Managing mailboxes

Mailboxes are used by backwards compatible Short Messaging and Multimedia Messaging traffic paths.


- About Mailbox administration
- Mailboxes
  - Creating mailboxes
  - Deleting mailboxes by address
  - Deleting mailboxes by owner (application)
- Message translation
  - Adding a message translation
  - Deleting a message translation
- Reference of provisioning attributes and operations for mailboxes


## About Mailbox administration

All messages used by the backwards compatible Parlay X Short Messaging and backwards compatible Parlay X Multimedia Messaging traffic paths are managed using mailboxes. Each application that uses the Messaging Service capability must have one or more mailboxes created and assigned to it. When it is created, each mailbox has an address.

Mailbox addresses can also be mapped to destination address short codes and message prefixes. The combination of a mailbox address, destination address short code and message prefix is called a mailbox translation.

# Mailboxes

To create or delete mailboxes, use the following procedures.

## Creating mailboxes

To create one or more mailboxes for an application account.

**Note:** As the number of mailboxes on BEA WebLogic Network Gatekeeper may become very large, doing searches on mailbox data can become quite time-consuming. As a result, keeping track of each application's mailboxes, mailbox address ranges, and mailbox passwords in a separate file - for example, an Excel file - is highly recommended.

1. Identify a free address range to be used for the mailboxes. Check the file mentioned in the note above.
2. Start in the configuration and operations page for **ESPA_Messaging**.
3. List mailboxes using Operation: listMailboxes
4. Verify that the address range is free.

   If the address range is free, no mailboxes will be displayed.
5. To create a range of mailboxes, use Operation: createMailboxRange. To create a single mailbox, use Operation: createMailbox.

   The result is displayed and the mailboxes are created.
6. Distribute the mailbox addresses and password to the service provider.

## Deleting mailboxes by address

To delete one or more mailboxes based on address:

**Note:** Mailboxes can be deleted for individual addresses or by address range.

1. Identify the mailbox address or address range in the file you created to keep track of the mailbox addresses.
2. Start in the configuration and operations page for **ESPA_Messaging**.

3. When the method list appears, verify that the address or address range you want to delete exists and that the desired service provider is the owner of all mailboxes to delete using Operation: listMailboxes.

   Check to make sure the range you wish to delete exists in the list that is displayed.

4. Remove a rage of mailboxes using Operation: removeMailboxRange. Remove a single mailbox using Operation: removeMailbox.

The result is displayed in the Messages and the mailbox(es) are deleted.

## Deleting mailboxes by owner (application)

To delete all mailboxes owned by a specific application:

1. Identify the mailbox address or address range in the file you created to keep track of the mailbox addresses.
2. Start in the configuration and operations page for **ESPA_Messaging**.
3. Verify that the address or address range you want to delete exists for the owner (application) by using Operation: listMailboxesByOwner.
4. Check to make sure the owner whose range you wish to delete exists in the list that is displayed.
5. Remove the mailbox using Operation: removeMailboxByOwner.

The result is displayed and the application's mailbox(es) are deleted.

# Message translation

To add or delete a mapping from a messaging mailbox to a short code and message prefix.

## Adding a message translation

To specify a destination address short code and message prefix for a mailbox:

**Note:** The routing for the destination address short code towards BEA WebLogic Network Gatekeeper must be defined in the network.

1. Start in the configuration and operations page for **ESPA_Messaging**.
2. Verify the mailbox address using Operation: listMailboxes.

   Make sure the mailbox is displayed.

3. List current translations using Operation: listMailboxTranslationsForAddress.

   The current translations for that mailbox are displayed.

4. Add translation using Operation: addMailboxTranslation.

The result is displayed in the Messages pane and the message translation is added.

## Deleting a message translation

To delete a message translation for a mailbox.

1. Start in the configuration and operations page for **ESPA_Messaging**.
2. List current translations using Operation: listMailboxTranslationsForAddress.

The current translations are displayed.

3. Delete translation using Operation: deleteMailboxTranslation.

**Note:** The actual mailboxes must be deleted separately. See Deleting mailboxes by address or Deleting mailboxes by owner (application) for more information.

# Reference of provisioning attributes and operations for mailboxes

Below is a list of attributes and operations for configuration and maintenance for mailboxes in the managed object ESPA_messaging:

- Operation: addMailboxTranslation
- Operation: createMailbox
- Operation: createMailboxRange
- Operation: deleteMailboxTranslation
- Operation: deleteMailboxTranslation
- Operation: listMailboxTranslationsForAddress
- Operation: listMailboxes
- Operation: listMailboxesByOwner
- Operation: removeMailbox

- Operation: removeMailboxByOwner
- Operation: removeMailboxRange


## Operation: addMailboxTranslation

Scope: Domain

Adds a mailbox translation rule.

Signature:

```
addMailboxTranslation(destAddr:String, msgPrefix:String, mailbox:String)
```

**Table 10-1 addMailboxTranslation**

| addMailboxTranslation | |
|---|---|
| Parameter | Description |
| destAddr | The destination address short code to be used instead of the real mailbox address. |
| msgPrefix | The keyword to be entered in the beginning of the message. The message prefix is case sensitive. If left empty, this mailbox will be the default mailbox for the destination address short code. That is, it will be used for messages that does not start with a valid message prefix. |
| mailbox | The ID of the mailbox to route to translate to. |

## Operation: createMailbox

Creates a new mailbox.

Signature:

```
createMailbox(addr:String, applicationId:String, serviceProviderId:String,
pwd:String)
```

**Table 10-2 createMailbox**

| createMailbox | |
|---|---|
| **Parameter** | **Description** |
| addr | Mailbox ID. |
| applicationId | The application account ID for the application using the mailbox. |
| serviceProviderId | The service provider account ID. |
| pwd | A password to be used by the application when accessing the mailboxes. |

## Operation: createMailboxRange

Creates a range of new mailboxes.

Signature:

```
createMailboxRange(startAddr:String, endAddr:String, applicationId:String,
serviceProviderId:String, pwd:String)
```

**Table 10-3 createMailboxRange**

| createMailboxRange | |
|---|---|
| **Parameter** | **Description** |
| startAddr | The first address in a range of free internal mailbox addresses in Network Gatekeeper. You do not have to enter the service centre address part of the mailbox address.<br>Integer (leading zeroes might have to be added to the internal mailbox address. This depends on the address format used). |
| endAddr | The last address in a range of free internal mailbox addresses. Must be greater than the address in startAddr. |
| applicationID | The application account ID of the application using the mailbox. |

**Table 10-3 createMailboxRange**

| createMailboxRange | |
|---|---|
| **Parameter** | **Description** |
| serviceProviderID | The service provider account ID. |
| pwd | A password to be used by the application when accessing the mailboxes. |

## Operation: deleteMailboxTranslation

Deletes a mailbox translation.

Signature:

```
deleteMailboxTranslation(destAddr:String, msgPrefix:String)
```

**Table 10-4 deleteMailboxTranslation**

| deleteMailboxTranslation | |
|---|---|
| **Parameters** | **Description** |
| destAddr | The destination address short code used instead of the real mailbox address. |
| msgPrefix | The keyword to be entered in the beginning of the message. The message prefix is case sensitive.<br>If left empty, all message translations related to the destination address short code will be deleted. |

## Operation: listMailboxTranslationsForAddress

Lists all mailbox translation rules for a specific address.

```
listMailboxTranslationsForAddress(destAddr:String)
```

**Table 10-5 emptyMailbox**

| emptyMailbox | |
|---|---|
| **Parameters** | **Description** |
| `destAddr` | The address for which to list mailbox translations. |

## Operation: listMailboxes

List a range of mailbox IDs and which service provider account and application account the mailboxes.

Signature:

```
listMailboxes(fromAddress:String, toAddress:String)
```

**Table 10-6 listMailboxes**

| listMailboxes | |
|---|---|
| **Parameters** | **Description** |
| fromAddress | The ID of the first mailbox in the range. |
| toAddress | The ID of the last mailbox in the range.<br>If set to 0, no last address in the range is given but instead the next 50 mailboxes, starting with the fromAddress, will be included in the list. |

## Operation: listMailboxesByOwner

Lists the IDs of all mailboxes belonging to a given service provider and application.

```
listMailboxesByOwner(applicationId:String, serviceProviderId:String)
```

**Table 10-7 listMailboxesByOwner**

| listMailboxesByOwner | |
|---|---|
| **Parameters** | **Description** |
| applicationId | The application account ID. |
| serviceProviderId | The service provider account ID. |

## Operation: removeMailbox

Deletes a mailbox.

Signature:

```
removeMailbox(addr:String)
```

**Table 10-8 removeMailbox**

| removeMailbox | |
|---|---|
| **Parameters** | **Description** |
| addr | The mailbox ID |

## Operation: removeMailboxByOwner

Deletes all mailboxes owned by a given service provider account and application account.

Signature:

```
removeMailboxByOwner(applicationId:String, serviceProviderId:String)
```

**Table 10-9 removeMailboxByOwner**

| removeMailboxByOwner | |
|---|---|
| **Parameters** | **Description** |
| applicationId | The application account ID. |
| serviceProviderId | The service provider account ID. |

## Operation: removeMailboxRange

Deletes a range of mailboxes.

Signature:

```
removeMailboxRange(startAddr:String, endAddr:String)
```

**Table 10-10 removeMailboxRange**

| removeMailboxRange | |
|---|---|
| **Parameters** | **Description** |
| startAddr | The ID of the first mailbox in the range. |
| endAddr | The ID of the last mailbox in the range. |

CHAPTER 11

# Mapping applications and service providers to OSA/Parlay accounts

The following sections describe how to map service providers and service providers to OSA/Parlay accounts:

- Understanding OSA/Parlay account mappings
- Overall workflow when creating a OSA client mapping
- Information and certificate exchange with OSA/Parlay administrator
- Create OSA client
- Map the Network Gatekeeper account to an OSA/Parlay client
- Provisioning for specific traffic paths

## Understanding OSA/Parlay account mappings

The Network Gatekeeper can connect to a OSA/Parlay Gateway either as:

1. One single user
2. One user per Network Gatekeeper service provider account
3. One user per Network Gatekeeper service provider and application account combination

Combinations of the above are not allowed. The Network Gatekeeper administrator must choose one of these connection modes, and use the same mode for all Network Gatekeeper applications In the first case, the connection is a system-wide configuration, in the other two cases, the

connection is setup as a part of the provisioning chain for Network Gatekeeper service providers and their applications.

In order to provision the connection, Network Gatekeeper has the concept of **OSA client** and **OSA client mapping**.

An **OSA client** represents the account in the OSA/Parlay Gateway. An OSA client has the following attributes:

- OSA client application ID, constituted of the Enterprise Operator ID and the Application ID as provisioned in the OSA/Parlay Gateway,
- Depending on the authentication method used, a private key (with associated password and keystore password) and public certificate to be used when authenticating.

An **OSA client mapping** maps an OSA client with OSA/Parlay SCSs. There must be (at least) one OSA client mapping per OSA SCS being used. If the traffic path uses *n* OSA SCSs, *n* Client Mappings must be defined. Wildcard mechanisms are used in the OSA client mapping, as described below:

  – The OSA client mapping may be set up per application level, so there is a one to one mapping between a Network Gatekeeper service provider and application account combination and the equivalent OSA Client. This means that every transaction originating from a specific application results in a transaction in the OSA Gateway that is traceable to that specific application.

  – The OSA client mapping can use a wildcard for the application level, but specify the service provider, so multiple Network Gatekeeper applications that originate from a common service provider are mapped to a single OSA client. In this case, the transactions in the OSA/Gateway are traceable only to the service provider.

  – The OSA client mapping can use wildcards for both the service provider and the application level, so all applications from all service providers are mapped to a single OSA Client. In this case, transactions in the OSA/Gateway are traceable only to Network Gatekeeper.

Defining the OSA client mapping is a part of the provisioning chain in when setting up service provider and application accounts. If the authentication method used between the Network Gatekeeper and an OSA/Parlay Gateway requires certificates and keys, these are set up when establishing user mapping details.

The following attributes are specified per OSA client mapping:

- The mapping to service provider account and application account, if any:

  - Service provider account ID,
  - Application account ID
- SCS type,
- Enterprise Operator ID, as provisioned in the OSA Parlay Gateway,
- Application ID, as provisioned in the OSA Parlay Gateway,
- Authentication type to use
- Encryption method to use
- Signing algorithm
- ID of the OSA/Parlay Gateway.
- Flag that indicates if the connection should be established immediately or when the first application request is executed.

# Overall workflow when creating a OSA client mapping

Follow the steps below for information on how to connect an application account to an OSA/Parlay Gateway:

Step 1: Information and certificate exchange with OSA/Parlay administrator.

Step 2: Create OSA client.

Step 3: Map the Network Gatekeeper account to an OSA/Parlay client.

# Information and certificate exchange with OSA/Parlay administrator

The OSA/Parlay Gateway administrator must provide the following information with regards to the OSA/Parlay Gateway account and OSA/Parlay Framework:

- The **entOpId** (Enterprise Operator ID) - Depending on how the OSA/Parlay operator administers applications (OSA/Parlay clients) the entOpId can be valid for:
  - All applications registered in WebLogic Network Gatekeeper
  - All applications connected to a service provider account
  - A single application account

- The **appId** (Application ID) to be used for the application account (clientAppId=entOpId + \ + appId)
- The OSA/Parlay **service types** for the OSA/Parlay SCSs to which the application is to be mapped
- The encryption method used
- The signing algorithm used

**Note:** The WebLogic Network Gatekeeper as a whole must be connected to the OSA/Parlay gateway before any individual applications are connected.

If the authentication method towards the OSA/Parlay Framework requires a certificate, the Network Gatekeeper administrator must generate one, and distribute it to the OSA/Parlay Gateway administrator. The associated key must be stored in the Network Gatekeeper keystore, this is done when the OSA client is created, see Create OSA client.

For non-production environments, the WebLogic Server CertGen utility can be used to create certificates and keys.

# Create OSA client

The OSA client is either:

- One single client for all service providers using Network Gatekeeper
- A service provider account
- A combination of a service provider account and an application account

The OSA client is the entity being used when creating the OSA client mapping.

1. Starting in the configuration and operations page for **OSA_access**, select **addClient** from the **Select An Operation** drop-down list.

   The parameters for the operation are displayed.

2. Enter:

   The application's clientAppId (and alias in the keystore) provided by the OSA/Parlay Gateway administrator in **osaClientAppId**.

   The directory path (including file name) to the private key in **clientKeyFile**.

   The directory path (including file name) to the OSA/Parlay Frameworks certificate in **clientCertFile**.

The private key password as defined when the private key was generated in **clientKeyPwd**.

The keystore's password as defined in when configuring the WebLogic Network Gatekeeper, in **keystorePwd**.

3. Click **Invoke**.

   The OSA client is created.

# Map the Network Gatekeeper account to an OSA/Parlay client

The mapping may be applied on service provider account, application account, or Network Gatekeeper level.

**Note:** One mapping must be created for each OSA/Parlay SCS (network service) the Network Gatekeeper application is using in the OSA/Parlay gateway.

1. Starting in the configuration and operations page for **OSA_access**, select **addMapping** from the **Select An Operation** drop-down list.

   The parameters for the operation are displayed.

2. Enter:

   ID of the service provider account the application is associated with in **mapping.serviceProviderID**. If left empty, the mapping will *not* be applied on service provider account and application account level.

   The ID of the application account in **mapping.applicationID**. If left empty, the mapping will *not* be applied on application account level.

   The OSA/Parlay service type of the OSA/Parlay SCS to which the Network Gatekeeper account is to be mapped in **mapping.serviceType**.

   The OSA/Parlay account's clientAppID, a string consisting of the entOpId followed by \, followed by the appId in **mapping.osaClientAppId**. Example: `sp1\app1`.

   OSA/Parlay service properties to be used in the look up (service discovery) phase when requesting a service (OSA/Parlay SCS) from the OSA/Parlay Gateway in **mapping.properties**. The properties are specified as a space separated list in the following way: `<propname1> <propval1> <propname2> <propval2>`

   The authentication type to be used in **mapping.authType**.The type is defined according to the OSA/Parlay standard. `P_AUTHENTICATION` is supported.

**Note:** When `P_AUTHENTICATION` is used, no encryption or signing algorithm will be used and the parameters **encryptionMethod** and **signingAlgorithm** can be left empty.

The method used for encryption in **mapping.encryptionMethod**. The type is defined according to OSA/Parlay standard. If the type is not specified, enter `P_RSA_1024`.

The signing algorithm in **mapping.signingAlgorithm**. The type is defined according to OSA/Parlay standard. If the type is not specified, enter `P_MD5_RSA_1024`.

The OSA/Parlay gateway ID in **mapping.gatewayId**. This ID was defined when the OSA/Parlay gateway was added.

A boolean indicating (`TRUE/FALSE`) if the connection to OSA/Parlay gateway should be initialized immediately in **initConnection**. That is, if authentication should performed when the **addClient** operation is invoked.

3. Click **Invoke**.

   The OSA client mapping is created.

# Provisioning for specific traffic paths

Applications may or may not use resources in the network, for example references to IVR messages, OSA/Parlay mailboxes. See Provisioning specific for traffic paths for information on the provisioning steps for the individual traffic paths.

# Reference: attributes and operations for Plugin_OSA_access

For a reference of attributes and operations see section *Managing and Configuring OSA/Parlay Gateway Connections* in *WebLogic Network Gatekeeper System Adminstrator's Guide*.

CHAPTER 12

# Reference: Provisioning operations for accounts and groups

The following sections contain a reference for the management attributes operations and attributes for service provider and application administration. All attributes and operations are available in the managed object ESPA_access in the Network Gatekeeper management console.


- Management operations related to service provider groups
- Management operations related to service provider accounts
- Management operations related to application accounts
- Management operations related to application groups
- Management operations related to application instance group IDs
- Reference: attributes and operations for ESPA_access


## Management operations related to service provider groups

Below is a list of management operations related to service provider groups:


- Attribute: NumberOfServiceProviderGroups (r)
- Operation: addServiceProviderGroupWithSLAString
- Operation: addServiceProviderGroupWithSLAUrl
- Operation: deleteServiceProviderGroup

- Operation: deleteServiceProviderGroupNodeSla
- Operation: getNumberOfServiceProvidersInGroup
- Operation: getSLAForServiceProviderGroup
- Operation: getServiceProviderGroup
- Operation: listServiceProviderGroupNodeSla
- Operation: listServiceProviderGroups
- Operation: listServiceProviderGroupsWithNodeSla
- Operation: listServiceProvidersInGroup
- Operation: setServiceProviderGroupNodeSlaString
- Operation: setServiceProviderGroupNodeSlaUrl
- Operation: updateServiceProviderGroupNodeSlaString
- Operation: updateServiceProviderGroupNodeSlaUrl
- Operation: updateServiceProviderGroupSLAString
- Operation: updateServiceProviderGroupSLAUrl

# Management operations related to service provider accounts

Below is a list of management operations related to service provider accounts:

- Attribute: NumberOfServiceProviderAccounts (r)
- Operation: activateServiceProviderAccount
- Operation: addServiceProviderAccount
- Operation: deactivateServiceProviderAccount
- Operation: deleteServiceProviderAccount
- Operation: getNumberOfApplicationsInServiceProvider
- Operation: getServiceProviderAccount

- Operation: getStateForServiceProviderAccount
- Operation: listApplicationAccounts
- Operation: listApplicationAccountsForServiceProvider
- Operation: listServiceProviderAccounts
- Operation: logoutServiceProviderAccount
- Operation: updateServiceProviderAccount

# Management operations related to application accounts

Below is a list of management operations related to application accounts:

- Operation: activateApplicationAccount
- Operation: addApplicationAccount
- Operation: deactivateApplicationAccount
- Operation: deleteApplicationAccount
- Operation: getApplicationAccount
- Operation: getNumberOfApplicationInstanceGroupsInApplications
- Operation: getStateForApplicationAccount
- Operation: logoutApplicationAccount
- Operation: updateApplicationAccount

# Management operations related to application groups

Below is a list of management operations related to application groups:

- Attribute: NumberOfApplicationGroups (r)
- Operation: addApplicationGroupWithSLAString
- Operation: addApplicationGroupWithSLAUrl
- Operation: deleteApplicationGroup
- Operation: getApplicationGroup

- Operation: getNumberOfApplications
- Operation: getNumberOfApplicationsInApplicationGroup
- Operation: getSLAForApplicationGroup
- Operation: listApplicationAccountsInGroup
- Operation: listApplicationGroups
- Operation: updateApplicationGroupSLAString
- Operation: updateApplicationGroupSLAUrl

# Management operations related to application instance group IDs

Below is a list of management operations related to application instance group IDs:

- Operation: activateApplicationInstanceGroup
- Operation: addApplicationInstanceGroupSLAString
- Operation: deactivateApplicationInstanceGroup
- Operation: deleteApplicationInstanceGroup
- Operation: getApplicationInstanceGroup
- Operation: getNumberOfApplicationInstanceGroupRefs
- Operation: getNumberOfLoggedInApplicationInstanceGroups
- Operation: getNumberOfLoggedInInstancesInInstanceGroup
- Operation: getSessionInfo
- Operation: getSLAForApplicationInstanceGroup ID
- Operation: getStateForApplicationInstanceGroup
- Operation: listApplicationInstanceGroupRefs
- Operation: listApplicationInstanceGroups
- Operation: listLoggedInApplicationInstanceGroups

- Operation: logoutApplicationInstanceGroup
- Operation: setPasswordForApplicationInstanceGroup
- Operation: unlockApplicationInstanceGroup
- Operation: updateApplicationInstanceGroupSLAString

# Reference: attributes and operations for ESPA_access

Below is a list of attributes and operations for configuration and maintenance:

- Attribute: NumberOfApplicationGroups (r)
- Attribute: NumberOfServiceProviderGroups (r)
- Attribute: OverloadPercentage
- Attribute: SevereOverloadPercentage
- Attribute: NumberOfServiceProviderAccounts (r)
- Operation: activateApplicationAccount
- Operation: activateApplicationInstanceGroup
- Operation: activateServiceProviderAccount
- Operation: addApplicationAccount
- Operation: addApplicationGroupWithSLAString
- Operation: addApplicationGroupWithSLAUrl
- Operation: addApplicationInstanceGroupSLAString
- Operation: addServiceProviderAccount
- Operation: addServiceProviderGroupWithSLAString
- Operation: addServiceProviderGroupWithSLAUrl f
- Operation: deactivateApplicationAccount
- Operation: deactivateApplicationInstanceGroup
- Operation: deactivateServiceProviderAccount

- Operation: deleteApplicationAccount
- Operation: deleteApplicationGroup
- Operation: deleteApplicationInstanceGroup
- Operation: deleteServiceProviderAccount
- Operation: deleteServiceProviderGroup
- Operation: deleteServiceProviderGroupNodeSla
- Operation: getApplicationAccount
- Operation: getApplicationGroup
- Operation: getApplicationInstanceGroup
- Operation: getNumberOfApplicationInstanceGroupRefs
- Operation: getNumberOfApplicationInstanceGroupsInApplications
- Operation: getNumberOfApplications
- Operation: getNumberOfApplicationsInApplicationGroup
- Operation: getNumberOfApplicationsInServiceProvider
- Operation: getNumberOfLoggedInApplicationInstanceGroups
- Operation: getNumberOfLoggedInInstancesInInstanceGroup
- Operation: getNumberOfServiceProvidersInGroup
- Operation: getSLAForApplicationGroup
- Operation: getSLAForApplicationInstanceGroup ID
- Operation: getSLAForServiceProviderGroup
- Operation: getServiceProviderAccount
- Operation: getServiceProviderGroup
- Operation: getSessionInfo
- Operation: getStateForApplicationAccount
- Operation: getStateForApplicationInstanceGroup

- Operation: getStateForServiceProviderAccount
- Operation: listApplicationAccounts
- Operation: listApplicationAccountsForServiceProvider
- Operation: listApplicationAccountsInGroup
- Operation: listApplicationGroups
- Operation: listApplicationInstanceGroupRefs
- Operation: listApplicationInstanceGroups
- Operation: listESPAServices
- Operation: listLoggedInApplicationInstanceGroups
- Operation: listServiceProviderAccounts
- Operation: listServiceProviderGroupNodeSla
- Operation: listServiceProviderGroups
- Operation: listServiceProviderGroupsWithNodeSla
- Operation: listServiceProvidersInGroup
- Operation: logoutApplicationAccount
- Operation: logoutApplicationInstanceGroup
- Operation: logoutServiceProviderAccount
- Operation: setPasswordForApplicationInstanceGroup
- Operation: setServiceProviderGroupNodeSlaString
- Operation: setServiceProviderGroupNodeSlaUrl
- Operation: unlockApplicationInstanceGroup
- Operation: updateApplicationAccount
- Operation: updateApplicationGroupSLAString
- Operation: updateApplicationGroupSLAUrl
- Operation: updateApplicationInstanceGroupSLAString

- Operation: updateServiceProviderAccount
- Operation: updateServiceProviderGroupNodeSlaString
- Operation: updateServiceProviderGroupNodeSlaUrl
- Operation: updateServiceProviderGroupSLAString
- Operation: updateServiceProviderGroupSLAUrl

## Attribute: NumberOfApplicationGroups (r)

Scope: Cluster

Unit: n/a

Format: int

Displays the number of registered application groups.

## Attribute: NumberOfServiceProviderGroups (r)

Scope: Cluster

Unit: n/a

Format: int

Displays the number of registered service provider groups.

## Attribute: OverloadPercentage

Scope: Server

Unit: n/a

Format: int [0...100]

Specifies the load percentage that defines when the software module will raise an overloaded alarm. Must be less than Attribute: SevereOverloadPercentage.

## Attribute: SevereOverloadPercentage

Scope: Server

Unit: n/a

Format: int [0...100]

Specifies the load percentage that defines when the plug-in will raise a severely overloaded alarm.

Must be larger than Attribute: OverloadPercentage.

## Attribute: NumberOfServiceProviderAccounts (r)

Scope: Cluster

Unit: n/a

Format: int

Displays the number of registered service provider accounts.

## Operation: activateApplicationAccount

Scope: Cluster

Activate an application account. The application account must be activated before the application can start accessing services in the network.

Signature:

```
activateApplicationAccount(serviceProviderAccountID:String,
applicationAccountID: String)
```

**Table 12-1 activateApplicationAccount**

| activateApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | Service provider account ID. |
| applicationAccountID | Application account ID. |

## Operation: activateApplicationInstanceGroup

Scope: Cluster

Activates an application instance group ID (the application's login username). The application instance group ID must be activated before the application can start accessing services in the network.

Signature:

```
activateApplicationInstanceGroup(serviceProviderAccountID:String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-2 activateApplicationInstanceGroup**

| activateApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | Service provider account ID. |
| applicationAccountID | Application account ID. |
| applicationInstanceGroupID | Application instance group ID. |

## Operation: activateServiceProviderAccount

Scope: Cluster

Activates a service provider account. The service provider account must be activated before any of its applications can start accessing services in the network.

Signature:

```
activateServiceProviderAccount(serviceProviderAccountID:String)
```

**Table 12-3 activateServiceProviderAccount**

| activateServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | Service provider account ID. |

## Operation: addApplicationAccount

Scope: Cluster

Creates an application account. The service provider account and the application group to which this application account is to be connected must be created before the application account is created

Signature:

```
addApplicationAccount(applicationAccount.serviceProviderAccountID:String,
applicationAccount.applicationAccountID: String,
applicationAccount.applicationGroupID: String,
applicationAccount.applicationRef:String, applicationAccount.properties)
```

**Table 12-4 addApplicationAccount**

| addApplicationAccount | |
|---|---|
| Parameter | Description |
| applicationAccount.serviceProviderAccountID | The ID for the service provider account that the application shall belong to. |
| applicationAccount.applicationAccountID | The ID of the new application account. |
| applicationAccount.applicationGroupID | ID for the application group that the application shall belong to. |
| applicationAccount.applicationRef | This is an optional reference used for internal mappings. Must not contain spaces. |
| applicationAccount.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: addApplicationGroupWithSLAString

Scope: Cluster

Creates an application group. The application group is associated with an SLA that can be used by one or more application accounts.

Signature:

```
addApplicationGroupWithSLAString(applicationGroup.applicationGroupID:
String, applicationGroup.slaContents: String,applicationGroup.properties)
```

**Table 12-5 addApplicationGroupWithSLAString**

| addApplicationGroupWithSLAString | |
|---|---|
| Parameter | Description |
| applicationGroup.applicationGroupID | ID for the new application group. |
| applicationGroup.slaContents | Content of SLA in string format. |
| applicationGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: addApplicationGroupWithSLAUrl

Scope: Cluster

Creates an application group. The application group is associated with an SLA that can be used by one or more application accounts.

Signature:

```
addApplicationGroupWithSLAUrl(applicationGroup.applicationGroupID: String,
applicationGroup.slaContents: String,applicationGroup.properties)
```

**Table 12-6 addApplicationGroupWithSLAUrl**

| addApplicationGroupWithSLAUrl | |
|---|---|
| Parameter | Description |
| applicationGroup.applicationGroupID | ID for the new application group. |
| applicationGroup.slaContents | The URL for the SLA file. |
| applicationGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: addApplicationInstanceGroupSLAString

Scope: Cluster

Creates an application instance group ID, which functions as the login username for the application. The parameter `applicationInstanceGroup.slaContents` must be set to 1.

**Note:** Application instance group IDs must be unique.

Signature:

```
addApplicationInstanceGroupSLAString(serviceProviderAccountID: String,
applicationInstanceGroup.applicationAccountID: String,
applicationInstanceGroup.applicationInstanceGroupID: String,
applicationInstanceGroup.applicationInstanceGroupRef: String,
applicationInstanceGroup.slaContents: String,
applicationInstanceGroup.properties:, password: String)
```

**Table 12-7 addApplicationInstanceGroupSLAString**

| addApplicationInstanceGroupSLAString | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID for the service provider account that the application instance group shall belong to. |
| applicationInstanceGroup.applicationAccountID | ID of the application account that the application instance group shall belong to. |
| applicationInstanceGroup.applicationInstanceGroupID | ID of the application instance group to be created. Must be unique.<br>An application use this as username when establishing a session (logging in). |
| applicationInstanceGroup.applicationInstanceGroupRef | This is an optional reference used for internal mappings. Must not contain spaces. |
| applicationInstanceGroup.slaContents | Must always be set to 1. |

**Table 12-7 addApplicationInstanceGroupSLAString**

| addApplicationInstanceGroupSLAString | |
|---|---|
| Parameter | Description |
| applicationInstanceGroup.properties | These properties can only be set from an integrated PRM/CRM system. |
| password | The password for the application instance group. Used by applications when establishing a session. |

## Operation: addServiceProviderAccount

Scope: Cluster

Creates a service provider account. The service provider group to which this service provider account is to be connected must be created before the service provider account is created.

Signature:

```
addServiceProviderAccount(serviceProviderAccount.serviceProviderAccountID:
: String, serviceProviderAccount.serviceProviderGroupID: String,
serviceProviderAccount.serviceProviderRef: String,
serviceProviderAccount.properties)
```

**Table 12-8 addServiceProviderAccount**

| addServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccount.serviceProviderAccountID | ID of the service provider account to be created. |
| serviceProviderAccount.serviceProviderGroupID | ID of the service provider group the service provider group shall belong to. |
| serviceProviderAccount.serviceProviderRef | This is an optional reference used for internal mappings. Must not contain spaces. |
| serviceProviderAccount.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: addServiceProviderGroupWithSLAString

Scope: Cluster

Creates service provider group. A service provider group SLA must exist. The SLA is provided as a string.

Signature:

```
addServiceProviderGroupWithSLAString(serviceProviderGroup.serviceProviderGroupID: String, serviceProviderGroup.slaContents: String, serviceProviderGroup.properties)
```

**Table 12-9 addServiceProviderGroupWithSLAString**

| addServiceProviderGroupWithSLAString | |
|---|---|
| Parameter | Description |
| serviceProviderGroup.serviceProviderGroupID | ID of the service provider group to be created. |
| serviceProviderGroup.slaContents | Content of the SLA in string format. |
| serviceProviderGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: addServiceProviderGroupWithSLAUrl

Scope: Cluster

Creates service provider group. A service provider group SLA must exist. A URL to the SLA is provided.

Signature:

```
addServiceProviderGroupWithSLAUrl(serviceProviderGroup.serviceProviderGroupID: String, serviceProviderGroup.slaContents: String, serviceProviderGroup.properties)
```

**Table 12-10 addServiceProviderGroupWithSLAUrl**

| addServiceProviderGroupWithSLAUrl | |
|---|---|
| Parameter | Description |
| serviceProviderGroup.serviceProviderGroupID | ID of the service provider group to be created. |
| serviceProviderGroup.slaContents | URL to the SLA. |
| serviceProviderGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: deactivateApplicationAccount

Scope: Cluster

Deactivates an application account. This deactivates, or temporarily prohibits the flow of traffic to and from, this account's applications.

Signature:

```
deactivateApplicationAccount(serviceProviderAccountID: String,
applicationAccountID: String)
```

**Table 12-11 deactivateApplicationAccount**

| deactivateApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: deactivateApplicationInstanceGroup

Scope: Cluster

Deactivates an application instance group ID. Deactivates, or temporarily prohibits the flow of traffic to and from, the application that established a session using this application instance group ID.

Signature:

```
deactivateApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-12 deactivateApplicationInstanceGroup**

| deactivateApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group ID belongs to. |
| applicationAccountID | ID of the application account the application instance group ID belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: deactivateServiceProviderAccount

Scope: Cluster

Deactivates a service provider account. This deactivates, or temporarily prohibits the flow of traffic to and from, this account's applications.

Signature:

```
deactivateServiceProviderAccount(serviceProviderAccountID: String)
```

**Table 12-13 deactivateServiceProviderAccount**

| deactivateServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account. |

## Operation: deleteApplicationAccount

Scope: Cluster

Deletes an application account.

**Note:** Deleting an application account also deletes all the application instance group IDs assigned to that account.

Signature:

```
deleteApplicationAccount(serviceProviderAccountID: String,
applicationAccountID: String)
```

**Table 12-14 deleteApplicationAccount**

| deleteApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: deleteApplicationGroup

Scope: Cluster

Deletes an application group.

Signature:

```
deleteApplicationGroup(applicationGroupID: String)
```

**Table 12-15 deleteApplicationGroup**

| deleteApplicationGroup | |
|---|---|
| Parameter | Description |
| applicationGroupID | If of application group. |

## Operation: deleteApplicationInstanceGroup

Scope: Cluster

Deletes an application instance group ID.

Signature:

```
deleteApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-16 deleteApplicationInstanceGroup**

| deleteApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group ID belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: deleteServiceProviderAccount

Scope: Cluster

Deletes a service provider account.

**WARNING:** Deleting the service provider account also deletes all related application accounts and application instance group IDs.

Signature:

```
deleteServiceProviderAccount(serviceProviderAccountID: String)
```

**Table 12-17 deleteServiceProviderAccount**

| deleteServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account. |

## Operation: deleteServiceProviderGroup

Scope: Cluster

Deletes a service provider group.

**WARNING:** When deleting the service provider group, all related service provider accounts, application accounts and application instance group IDs will also be deleted.

Signature:

```
deleteServiceProviderGroup(serviceProviderGroupID: String)
```

**Table 12-18 deleteServiceProviderGroup**

| deleteServiceProviderGroup | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: deleteServiceProviderGroupNodeSla

Scope: Cluster

Deletes a node SLA for a certain service provider group.

Signature:

```
deleteServiceProviderGroupNodeSla(serviceProviderGroupID: String)
```

**Table 12-19 deleteServiceProviderGroupNodeSla**

| deleteServiceProviderGroupNodeSla | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: getApplicationAccount

Scope: Cluster

Displays information about an application account. This information includes the related application group IDs, the application reference, if any, and the account's OAM properties, if any.

Signature:

```
getApplicationAccount(serviceProviderAccountID: String,
applicationAccountID: String)
```

**Table 12-20 getApplicationAccount**

| getApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: getApplicationGroup

Scope: Cluster

Displays information about an application group. The application group's SLA and OAM properties, if any, are displayed.

Signature:

```
getApplicationGroup(applicationGroupID: String)
```

**Table 12-21 getApplicationGroup**

| getApplicationGroup | |
|---|---|
| Parameter | Description |
| applicationGroupID | ID of the application group. |

## Operation: getApplicationInstanceGroup

Scope: Cluster

Displays information about an application instance group ID (the login username for the application). The application instance group ID's OAM properties, if any, and application instance group reference are displayed.

Signature:

```
getApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-22 getApplicationInstanceGroup**

| getApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: getNumberOfApplicationInstanceGroupRefs

Scope: Cluster

Displays information about application instance group ID references.

Signature:

```
getNumberOfApplicationInstanceGroupRefs(serviceProviderAccountID: String,
applicationAccountID: String, state: int)
```

**Table 12-23 getNumberOfApplicationInstanceGroupRefs**

| getNumberOfApplicationInstanceGroupRefs | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |
| state | Filter on state. See Table 2-1. |

## Operation: getNumberOfApplicationInstanceGroupsInApplications

Scope: Cluster

Display the number of application instance group IDs assigned to the application account.

Signature:

```
getNumberOfApplicationInstanceGroupsInApplications(serviceProviderAccountI
D: String, applicationAccountID: String)
```

**Table 12-24 getNumberOfApplicationInstanceGroupsInApplications**

| getNumberOfApplicationInstanceGroupsInApplications | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: getNumberOfApplications

Scope: Cluster

Displays the number of application accounts in a certain state.

Signature:

```
getNumberOfApplications(serviceProviderAccountID: String, state: int)
```

**Table 12-25 getNumberOfApplications**

| getNumberOfApplications | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account to filter on.<br>Leave empty for no filter. |
| state | State to filter on. See Table 2-1. |

## Operation: getNumberOfApplicationsInApplicationGroup

Scope: Cluster

Displays the number of application accounts connected to the application group

Signature:

```
getNumberOfApplicationsInApplicationGroup(applicationGroupID: String)
```

**Table 12-26 getNumberOfApplicationsInApplicationGroup**

| getNumberOfApplicationsInApplicationGroup | |
|---|---|
| Parameter | Description |
| applicationGroupID | ID of the application group. |

## Operation: getNumberOfApplicationsInServiceProvider

Scope: Cluster

Displays the number of application accounts connected to the service provider.

Signature:

```
getNumberOfApplicationsInServiceProvider(serviceProviderAccountID: String)
```

**Table 12-27 getNumberOfApplicationsInServiceProvider**

| getNumberOfApplicationsInServiceProvider | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account ID. |

## Operation: getNumberOfLoggedInApplicationInstanceGroups

Scope: Cluster

Displays the number of applications by ID that have sessions established with Network Gatekeeper.

Signature:

```
getNumberOfLoggedInApplicationInstanceGroups(serviceProviderAccountID:
String, applicationAccountID: String)
```

**Table 12-28 getNumberOfLoggedInApplicationInstanceGroups**

| getNumberOfLoggedInApplicationInstanceGroups | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: getNumberOfLoggedInInstancesInInstanceGroup

Scope: Cluster

Displays the number of application instance group IDs that have sessions established with Network Gatekeeper. Will always be 1.

Signature:

```
getNumberOfLoggedInInstancesInInstanceGroup(serviceProviderAccountID:
String, applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-29 getNumberOfLoggedInInstancesInInstanceGroup**

| getNumberOfLoggedInInstancesInInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: getNumberOfServiceProvidersInGroup

Scope: Cluster

Displays the number of service provider accounts related to a service provider group.

Signature:

```
getNumberOfServiceProvidersInGroup(serviceProvidersGroupID: String)
```

**Table 12-30 getNumberOfServiceProvidersInGroup**

| getNumberOfServiceProvidersInGroup | |
|---|---|
| Parameter | Description |
| serviceProvidersGroupID | ID of the service provider group. |

## Operation: getSLAForApplicationGroup

Scope: Cluster

Displays the SLA for an application group.

Signature:

```
getSLAForApplicationGroup(applicationGroupID: String)
```

**Table 12-31 getSLAForApplicationGroup**

| getSLAForApplicationGroup | |
|---|---|
| Parameter | Description |
| applicationGroupID | ID of the application group. |

## Operation: getSLAForApplicationInstanceGroup ID

Scope: Cluster

N/A

Signature:

```
getSLAForApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-32 getSLAForApplicationInstanceGroup**

| getSLAForApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: getSLAForServiceProviderGroup

Scope: Cluster

Displays the SLA for a service provider group.

Signature:

```
getSLAForServiceProviderGroup(serviceProviderGroupID : String)
```

**Table 12-33 getSLAForServiceProviderGroup**

| getSLAForServiceProviderGroup | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: getServiceProviderAccount

Scope: Cluster

Displays information about a service provider account. The related service provider group ID, the service provider reference, if any, and the account's OAM properties, if any, are displayed.

Signature:

```
getServiceProviderAccount(serviceProviderAccountID : String)
```

**Table 12-34 getServiceProviderAccount**

| getServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account. |

## Operation: getServiceProviderGroup

Scope: Cluster

Displays information about a service provider group. The service provider group’s SLA and OAM properties, if any, are displayed.

Signature:

```
getServiceProviderGroup(serviceProviderGroupID : String)
```

**Table 12-35 getServiceProviderGroup**

| getServiceProviderGroup | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of service provider group. |

## Operation: getSessionInfo

Scope: Cluster

Displays information about a session. The session is established between an application and Network Gatekeeper.

Signature:

```
getSessionInfo(sessionId : String)
```

**Table 12-36 getSessionInfo**

| getSessionInfo | |
|---|---|
| Parameter | Description |
| sessionId | ID of the session to get information about. |

## Operation: getStateForApplicationAccount

Scope: Cluster

Displays information about the state of an application account.

Signature:

```
getStateForApplicationAccount(serviceProviderAccountID : String,
applicationAccountID: String)
```

**Table 12-37 getStateForApplicationAccount**

| getStateForApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account hat application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: getStateForApplicationInstanceGroup

Scope: Cluster

Displays information about the state of an application instance group ID.

Signature:

```
getStateForApplicationInstanceGroup(serviceProviderAccountID : String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-38 getStateForApplicationInstanceGroup**

| getStateForApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: getStateForServiceProviderAccount

Scope: Cluster

Displays information about the state of a service provider account.

Signature:

```
getStateForServiceProviderAccount(serviceProviderAccountID : String)
```

**Table 12-39 getStateForServiceProviderAccount**

| getStateForServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account. |

## Operation: listApplicationAccounts

Scope: Cluster

Displays a list of application account IDs related to an application group.

Signature:

```
listApplicationAccounts(serviceProviderAccountID : String, state: int,
offset: int, length:int)
```

**Table 12-40 listApplicationAccounts**

| listApplicationAccounts | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account to list application accounts for. |
| state | Filter on state. See Table 2-1. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listApplicationAccountsForServiceProvider

Scope: Cluster

Displays a list of application account related to a service provider.

Signature:

```
listApplicationAccountsForServiceProvider(serviceProviderAccountID :
String, offset: int, length:int)
```

**Table 12-41 listApplicationAccountsForServiceProvider**

| listApplicationAccountsForServiceProvider | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account to list application accounts for.. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listApplicationAccountsInGroup

Scope: Cluster

Displays a list of application account IDs related to an application group.

Signature:

```
listApplicationAccountsInGroup(applicationGroupID : String, offset: int,
length:int)
```

**Table 12-42 listApplicationAccountsInGroup**

| listApplicationAccountsInGroup | |
|---|---|
| Parameter | Description |
| applicationGroupID | ID of the application group to list application accounts for. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listApplicationGroups

Scope: Cluster

Displays a list of application groups.

Signature:

```
listApplicationGroups(offset: int, length:int)
```

**Table 12-43 listApplicationGroups**

| listApplicationGroups | |
|---|---|
| Parameter | Description |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listApplicationInstanceGroupRefs

Scope: Cluster

Displays a list of application instance group ID references.

Signature:

```
listApplicationInstanceGroupRefs(serviceProviderAccountID: String,
applicationAccountID: String, state: int, offset: int, length:int)
```

**Table 12-44 listApplicationInstanceGroupRefs**

| listApplicationInstanceGroupRefs | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | Service provider account ID. |
| applicationAccountID | Application account ID. |
| state | State to filter on. See Table 2-1. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listApplicationInstanceGroups

Scope: Cluster

Displays a list of application instance group IDs per application account.

Signature:

```
listApplicationInstanceGroups(serviceProviderAccountID: String,
applicationAccountID: String, offset: int, length:int)
```

**Table 12-45 listApplicationInstanceGroups**

| listApplicationInstanceGroups | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listESPAServices

Scope: Cluster

Displays which ESPA services are registered. These are used in backwards compatible traffic paths.

Signature:

```
listESPAServices()
```

**Table 12-46 listESPAServices**

| listESPAServices | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: listLoggedInApplicationInstanceGroups

Scope: Cluster

Displays a list of application instance group IDs that currently have sessions established with Network Gatekeeper by application accounts.

Signature:

```
listApplicationInstanceGroups(serviceProviderAccountID: String,
applicationAccountID: String, offset: int, length:int)
```

**Table 12-47 listLoggedInApplicationInstanceGroups**

| listLoggedInApplicationInstanceGroups | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listServiceProviderAccounts

Scope: Cluster

Displays a list of service provider accounts.

Signature:

```
listServiceProviderAccounts(offset: int, length:int)
```

**Table 12-48 listServiceProviderAccounts**

| listServiceProviderAccounts | |
|---|---|
| Parameter | Description |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listServiceProviderGroupNodeSla

Scope: Cluster

Displays the node SLA for service provider group, if any.

Signature:

```
listServiceProviderGroupNodeSla(serviceProviderGroupID: String)
```

**Table 12-49 listServiceProviderGroupNodeSla**

| listServiceProviderGroupNodeSla | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: listServiceProviderGroups

Scope: Cluster

Displays a list of service provider groups.

Signature:

```
listServiceProviderGroups(offset: int, length:int)
```

**Table 12-50 listServiceProviderGroups**

| listServiceProviderGroups | |
|---|---|
| Parameter | Description |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: listServiceProviderGroupsWithNodeSla

Scope: Cluster

Displays a list of service provider groups that have node SLAs associated.

Signature:

```
listServiceProviderGroupsWithNodeSla()
```

**Table 12-51 listServiceProviderGroupsWithNodeSla**

| listServiceProviderGroupsWithNodeSla | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: listServiceProvidersInGroup

Scope: Cluster

Displays a list of service provider accounts related to a service provider group.

Signature:

```
listServiceProvidersInGroup(serviceProviderGroupID: String, offset: int,
length:int)
```

**Table 12-52 listServiceProvidersInGroup**

| listServiceProvidersInGroup | |
|---|---|
| Parameter | Description |
| serviceProviderGroupID | ID of the service provider group. |
| offset | Offset for the displayed list. |
| length | Number of entries in the displayed list. |

## Operation: logoutApplicationAccount

Scope: Cluster

Logs out all applications associated with an application account. De-establish sessions between Network Gatekeeper and the matching applications.

Signature:

```
logoutApplicationAccount(serviceProviderAccountID: String,
applicationAccountID: String)
```

**Table 12-53 logoutApplicationAccount**

| logoutApplicationAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccountID | ID of the application account. |

## Operation: logoutApplicationInstanceGroup

Scope: Cluster

Logs out a given application by ID. De-establish the session between Network Gatekeeper and the application.

Signature:

```
logoutApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-54 logoutApplicationInstanceGroup**

| logoutApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: logoutServiceProviderAccount

Scope: Cluster

Logs out all applications associated with a service provider account. De-establish sessions between Network Gatekeeper and the matching applications.

Signature:

```
logoutServiceProviderAccount(serviceProviderAccountID: String)
```

**Table 12-55 logoutServiceProviderAccount**

| logoutServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account. |

## Operation: setPasswordForApplicationInstanceGroup

Scope: Cluster

Defines a new password for an application instance group ID (the username).

Signature:

```
setPasswordForApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String,
newPassword : String)
```

**Table 12-56 setPasswordForApplicationInstanceGroup**

| setPasswordForApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |
| newPassword | The new password. |

## Operation: setServiceProviderGroupNodeSlaString

Scope: Cluster

Loads a node SLA for a service provider group.

Signature:

```
setServiceProviderGroupNodeSlaString(SLA: String, serviceProviderGroupID:
String)
```

**Table 12-57 setServiceProviderGroupNodeSlaString**

| setServiceProviderGroupNodeSlaString | |
|---|---|
| Parameter | Description |
| SLA | SLA content in string format. |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: setServiceProviderGroupNodeSlaUrl

Scope: Cluster

Loads a node SLA for a service provider group.

Signature:

```
setServiceProviderGroupNodeSlaUrl(url: String, serviceProviderGroupID:
String)
```

**Table 12-58 setServiceProviderGroupNodeSlaUrl**

| setServiceProviderGroupNodeSlaUrl | |
|---|---|
| Parameter | Description |
| url | URL to the SLA file. |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: unlockApplicationInstanceGroup

Scope: Cluster

Unlocks an application instance group ID. The application instance group ID (the username) is unlocked and its state is changed to activated. The application instance group ID may be locked because of too many failed login attempts.

Signature:

```
unlockApplicationInstanceGroup(serviceProviderAccountID: String,
applicationAccountID: String, applicationInstanceGroupID: String)
```

**Table 12-59 unlockApplicationInstanceGroup**

| unlockApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |

**Table 12-59 unlockApplicationInstanceGroup**

| unlockApplicationInstanceGroup | |
|---|---|
| Parameter | Description |
| applicationAccountID | ID of the application account the application instance group belongs to. |
| applicationInstanceGroupID | ID of the application instance group. |

## Operation: updateApplicationAccount

Scope: Cluster

Change the application group associated with an application account.

Signature:

```
updateApplicationAccount(applicationAccount.serviceProviderAccountID:
String, applicationAccount.applicationAccountID: String,
applicationAccount.applicationGroupID: String,
applicationAccount.applicationRef: String, applicationAccount.properties)
```

**Table 12-60 updateApplicationAccount**

| updateApplicationAccount | |
|---|---|
| Parameter | Description |
| applicationAccount.serviceProviderAccountID | ID of the service provider account the application account belongs to. |
| applicationAccount.applicationAccountID | ID of the application account to update. |
| applicationAccount.applicationGroupID | ID of application group to change to. |
| applicationAccount.applicationRef | The new application reference. This is an optional reference used for internal mappings. Must not contain spaces. |
| applicationAccount.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateApplicationGroupSLAString

Scope: Cluster

Updates the SLA for an application group.

Signature:

```
updateApplicationGroupSLAString(applicationGroup.applicationGroupID:
String, applicationGroup.slaContents: String, applicationGroup.properties)
```

**Table 12-61 updateApplicationGroupSLAString**

| updateApplicationGroupSLAString | |
|---|---|
| Parameter | Description |
| applicationGroup.applicationGroupID | ID of the application group to update. |
| applicationGroup.slaContents | The new SLA in string format. |
| applicationGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateApplicationGroupSLAUrl

Scope: Cluster

Updates the SLA for an application group.

Signature:

```
updateApplicationGroupSLAUrl(applicationGroup.applicationGroupID: String,
applicationGroup.slaContents: String, applicationGroup.properties)
```

**Table 12-62 updateApplicationGroupSLAUrl**

| updateApplicationGroupSLAUrl | |
|---|---|
| Parameter | Description |
| applicationGroup.applicationGroupID | ID of the application group to update. |
| applicationGroup.slaContents | URL to the new SLA. |
| applicationGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateApplicationInstanceGroupSLAString

Scope: Cluster

N/A.

Signature:

```
updateApplicationInstanceGroupSLAString(serviceProviderAccountID: String,
applicationInstanceGroup.applicationAccountID: String,
applicationInstanceGroup.applicationInstanceGroupID: String,
applicationInstanceGroup.applicationInstanceGroupRef: String,
applicationInstanceGroup.properties)
```

**Table 12-63 updateApplicationInstanceGroupSLAString**

| updateApplicationInstanceGroupSLAString | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | ID of the service provider account the application instance group belongs to. |
| applicationInstanceGroup.applicationAccountID | ID of the application account the application instance group belongs to. |

**Table 12-63 updateApplicationInstanceGroupSLAString**

| updateApplicationInstanceGroupSLAString | |
|---|---|
| Parameter | Description |
| applicationInstanceGroup.applicationInstanceGroupID | ID of the application instance group to update. |
| applicationInstanceGroup.applicationInstanceGroupRef | This is an optional reference used for internal mappings.<br>Ignored if left empty. Must not contain spaces. |
| applicationInstanceGroup.slaContents | The new SLA.<br>Always use 1. |
| applicationInstanceGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateServiceProviderAccount

Scope: Cluster

Changes the service provider group associated with an service provider account.

Signature:

```
updateServiceProviderAccount(serviceProviderAccount.serviceProviderAccount
ID: String, serviceProviderAccount.serviceProviderGroupID: String,
serviceProviderAccount.serviceProviderRef: String,
serviceProviderAccount.properties)
```

**Table 12-64 updateServiceProviderAccount**

| updateServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccount.serviceProviderAccountID | ID of service provider account to update. |
| serviceProviderAccount.serviceProviderGroupID | ID of the service provider group to change to. |

**Table 12-64 updateServiceProviderAccount**

| updateServiceProviderAccount | |
|---|---|
| Parameter | Description |
| serviceProviderAccount.serviceProviderRef | This is an optional reference used for internal mappings. Ignored if left empty. Must not contain spaces. |
| serviceProviderAccount.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateServiceProviderGroupNodeSlaString

Scope: Cluster

Updates the node SLA for a service provider group.

Signature:

```
updateServiceProviderGroupNodeSlaString(SLA: String,
serviceProviderGroupID: String)
```

**Table 12-65 updateServiceProviderGroupNodeSlaString**

| updateServiceProviderGroupNodeSlaString | |
|---|---|
| Parameter | Description |
| SLA | Content of the new SLA in string format. |
| serviceProviderGroupID | ID of service provider group. |

## Operation: updateServiceProviderGroupNodeSlaUrl

Scope: Cluster

Updates the node SLA for a service provider group.

Signature:

```
updateServiceProviderGroupNodeSlaUrl(url: String, serviceProviderGroupID:
String)
```

**Table 12-66 updateServiceProviderGroupNodeSlaString**

| updateServiceProviderGroupNodeSlaString | |
|---|---|
| Parameter | Description |
| url | URL to the new SLA file. |
| serviceProviderGroupID | ID of the service provider group. |

## Operation: updateServiceProviderGroupSLAString

Scope: Cluster

Updates the SLA for a service provider group.

Signature:

```
updateServiceProviderGroupSLAString(serviceProviderGroup.serviceProviderGr
oupID: String, serviceProviderGroup.slaContents: String,
serviceProviderGroup.properties)
```

**Table 12-67 updateServiceProviderGroupSLAString**

| updateServiceProviderGroupSLAString | |
|---|---|
| Parameter | Description |
| serviceProviderGroup.serviceProviderGroupID | ID of the service provider group to update. |
| serviceProviderGroup.slaContents | The new SLA content in string format. |
| serviceProviderGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

## Operation: updateServiceProviderGroupSLAUrl

Scope: Cluster

Updates the SLA for a service provider group.

Signature:

```
updateServiceProviderGroupSLAUrl(serviceProviderGroup.serviceProviderGroup
ID: String, serviceProviderGroup.slaContents: String,
serviceProviderGroup.properties)
```

**Table 12-68 updateServiceProviderGroupSLAUrl**

| updateServiceProviderGroupSLAUrl | |
|---|---|
| Parameter | Description |
| serviceProviderGroup.serviceProviderGroupID | ID of the service provider group to update. |
| serviceProviderGroup.slaContents | URL to the new SLA file. |
| serviceProviderGroup.properties | These properties can only be set from an integrated PRM/CRM system. |

CHAPTER 13

# Reference: Provisioning operations for SESPA_access

The following sections contain a reference for the management attributes operations and attributes for service provider and application administration. All attributes and operations are available in the managed object SESPA_access in the Network Gatekeeper management console.


- Settings related to overall application characteristics
- Settings related to applications and traffic paths
- Reference: attributes and operations for SESPA_access


## Settings related to overall application characteristics

Below is a list of settings and operational data related to overall characteristics:


- Attribute: LoginTicketLifetimeCheckEnabled (r)
- Attribute: LoginTicketResourceLifetime
- Operation: disableLoginTicketLifetimeCheck
- Operation: enableLoginTicketLifetimeCheck
- Operation: getLoginInfo
- Operation: logoutSession

# Settings related to applications and traffic paths

Below is a list of settings related to applications and traffic paths:

- Operation: getAppProperties
- Operation: listAppProperties
- Operation: setAppProperties

# Reference: attributes and operations for SESPA_access

Below is a list of attributes and operations for configuration and maintenance:

- Attribute: CorbaSettings (r)
- Attribute: LoginTicketLifetime
- Attribute: LoginTicketResourceLifetime
- Attribute: WSSettings (r)
- Attribute: LoginTicketLifetimeCheckEnabled (r)
- Operation: addloadDistribution
- Operation: deleteAppProperties
- Operation: disableLoginTicketLifetimeCheck
- Operation: enableLoginTicketLifetimeCheck
- Operation: getAppProperties
- Operation: getLoginInfo
- Operation: listAppProperties
- Operation: listActiveLoginTickets
- Operation: listAllLoginTickets
- Operation: listExpiredLoginTickets
- Operation: logoutSession
- Operation: removeLoadDistribution

- Operation: setAppProperties
- Operation: setCorbaSettings
- Operation: setWSSettings

## Attribute: CorbaSettings (r)

Scope: Cluster

Unit: milliseconds

Displays CORBA time-out settings.

## Attribute: LoginTicketLifetime

Scope: Cluster

Unit: minutes

Format: boolean

The session ID, also called the login ticket, is used to identify whether a login session is valid based on a configurable time-interval. When an application logs in, a login timer is started. Once the login timer expires, all subsequent requests from the application will be rejected, except requests to logout, login or to refresh the login ticket.

The application can request to refresh the login ticket during a second configurable time interval, see Attribute: LoginTicketResourceLifetime.

## Attribute: LoginTicketResourceLifetime

Scope: Cluster

Unit: minutes

Format: int

Specifies Timeout value until login ticket resource cleanup timers expire. Given in minutes relative to the timeout of a login ticket. When this timer expires, all resources allocated for an application are deallocated. Valid only if Login ticket expiration usage is enabled.

## Attribute: WSSettings (r)

Read-only.

Scope: Cluster

Unit: n/a

Deprecated. Not applicable.

## Attribute: LoginTicketLifetimeCheckEnabled (r)

Scope: Cluster

Read-only.

Unit: n/a

Format: boolean

Displays if loginticket check is enabled or not.

The session ID, also called the login ticket, is used to identify whether a login session is valid based on a configurable time-interval. When an application logs in, a login timer is started. Once the login timer expires, all subsequent requests from the application will be rejected, except requests to logout, login or to refresh the login ticket.

The application can request to refresh the login ticket during a second configurable time interval. When the login ticket timer expires, a resource cleanup timer is started. When the resource cleanup timer expires, all data about resources assigned to the login, such as opened mailboxes, are cleaned up.

## Operation: addloadDistribution

Scope: Server

Deprecated.

Specifies how the load will be distributed among the stateless adapters for backwards compatible traffic paths.

Signature:

```
addloadDistribution(distribution.host:String, distribution.share: int)
```

**Table 13-1 addloadDistribution**

| addloadDistribution | |
|---|---|
| Parameter | Description |
| distribution.host | Specifies how the load will be distributed internally in backwards-compatible traffic paths. |
| distribution.share | The host's load share in relation to the other hosts.<br>Shall be set to 1. |

## Operation: deleteAppProperties

Scope: Cluster

Deletes application properties.

Signature:

```
deleteAppProperties(serviceProviderAccountID:String, applicationAccountID:
String, applicationInstanceGroupID: String)
```

**Table 13-2 deleteAppProperties**

| deleteAppProperties | |
|---|---|
| Parameter | Description |
| serviceProviderAccountID | Service provider account ID. |
| applicationAccountID | Application account ID. |
| applicationInstanceGroupID | Application instance group ID. |

## Operation: disableLoginTicketLifetimeCheck

Scope: Cluster

Disables checks for session (loginticket) lifetime.

Signature:

```
disableLoginTicketLifetimeCheck()
```

**Table 13-3 disableLoginTicketLifetimeCheck**

| disableLoginTicketLifetimeCheck | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: enableLoginTicketLifetimeCheck

Scope: Cluster

Enables checks for session (loginticket) lifetime.

Enables the login ticket lifetime check. This is typically done when configuring the system. If the lifetime check is enabled when the system has been up and running for a while this may cause sessions to be immediately logged out.

Signature:

```
enableLoginTicketLifetimeCheck()
```

**Table 13-4 enableLoginTicketLifetimeCheck**

| enableLoginTicketLifetimeCheck | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: getAppProperties

Scope: Cluster

Displays the application properties defined for an application.

Signature:

```
getAppProperties(serviceProviderID:String, applicationID: String,
applicationInstanceGroupID: String)
```

**Table 13-5 getAppProperties**

| getAppProperties | |
|---|---|
| Parameter | Description |
| serviceProviderID | Service provider account ID.<br>Key. |
| applicationID | Application account ID.<br>Key. |
| applicationInstanceGroupID | Application instance group ID.<br>Key. |

## Operation: getLoginInfo

Scope: Cluster

Displays info about a given session ID (login ticket).

Signature:

```
getLoginInfo(loginTicket: String)
```

**Table 13-6 getLoginInfo**

| getLoginInfo | |
|---|---|
| Parameter | Description |
| loginTicket | ID for the session ID to get information about. |

## Operation: listAppProperties

Scope: Cluster

Displays a list of application accounts that have application account properties defined.

Signature:

```
listAppProperties(serviceProviderID: String)
```

**Table 13-7 listAppProperties**

| listAppProperties | |
|---|---|
| Parameter | Description |
| serviceProviderID | Service provider account ID to filter on. |

## Operation: listActiveLoginTickets

Scope: Cluster

Displays a list of active, non-expired, login tickets.

Signature:

```
listActiveLoginTickets()
```

**Table 13-8 listActiveLoginTickets**

| listActiveLoginTickets | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: listAllLoginTickets

Scope: Cluster

Displays a list of all login tickets, both non-expired and expired that has associated resources.

Signature:

```
listAllLoginTickets()
```

**Table 13-9 listAllLoginTickets**

| listAllLoginTickets | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: listExpiredLoginTickets

Scope: Cluster

Displays a list of all expired login tickets that has associated resources.

Signature:

```
listExpiredLoginTickets()
```

**Table 13-10 listExpiredLoginTickets**

| listExpiredLoginTickets | |
|---|---|
| Parameter | Description |
| - | - |

## Operation: logoutSession

Scope: Cluster

Logs out, or closes, a session.

Signature:

```
logoutSession(loginTicket : String)
```

**Table 13-11 logoutSession**

| logoutSession | |
|---|---|
| Parameter | Description |
| loginTicket | ID of the session to close. |

## Operation: removeLoadDistribution

Scope: Cluster

Deprecated method. Do not use.

Removes a load distribution setting for a given server.

Signature:

```
removeLoadDistribution(host : String)
```

**Table 13-12 removeLoadDistribution**

| removeLoadDistribution | |
|---|---|
| Parameter | Description |
| host | IP-address of the host to remove load distribution for. |

## Operation: setAppProperties

Scope: Cluster

Defines properties for backwards compatible traffic paths.

Signature:

```
setAppProperties(serviceProviderAccount.serviceProviderAccountID:: String,
serviceProviderAccount.serviceProviderGroupID: String,
serviceProviderAccount.serviceProviderRef: String,
serviceProviderAccount.properties)
```

**Table 13-13 setAppProperties**

| **setAppProperties** | |
|---|---|
| Parameter | Description |
| serviceProviderID | Service provider account ID to set properties for. |
| applicationID | Application account ID to set properties for. |
| applicationInstanceGroupID | Application instance group ID to set properties for. |
| appPassword | Password for the application instance group. |
| mailbox | The mailbox ID to be used for the application instance group.<br>Mandatory when the application uses the backwards compatible Short Messaging or Multimedia Messaging Parlay X 2.1 interfaces services and the format of the senderName parameter in the application request does not contain information on mailbox and mailbox password. |
| mailboxPassword | Password associated with the mailbox. |
| merchantId | The merchant ID for the application instance group.<br>Mandatory when Parlay X 2.1 Payment is used. |
| accountId | Account ID for the application instance group, to be used in payment/charging sessions.<br>Manadatory when Parlay X 2.1 Payment is used. |

**Table 13-13 setAppProperties**

| setAppProperties | |
|---|---|
| Parameter | Description |
| currency | Currency to be used for amount charging in payment/charging sessions.<br>Manadatory when Parlay X 2.1 Payment is used. |
| chargeVolumeType | The unit to be used for volume charging in payment/charging sessions.<br>A unit is defined as one of the following:<br>0 - undefined<br>1 - number of times or events<br>2 - octets<br>3 - seconds<br>4 - minutes<br>5 - hours<br>6 - days<br>Manadatory when Parlay X 2.1 Payment is used. |

## Operation: setCorbaSettings

Scope: Cluster

Defines CORBA timeout settings.

Signature:

```
setCorbaSettings(corbaConnectTimeoutMs: int, corbaRequestTimeoutMs: int)
```

**Table 13-14 setCorbaSettings**

| setCorbaSettings | |
|---|---|
| Parameter | Description |
| corbaConnectTimeoutMs | Connect timeout.<br>Given in ms. |
| corbaRequestTimeoutMs | corbaRequestTimeoutMs<br>Given in milliseconds. |

## Operation: setWSSettings

Scope: Cluster

Deprecated method.