Oracle® Communications Service Broker

Social Voice Communicator Implementation Guide

Release 6.1

E29461-01

February 2013

Oracle Communications Service Broker Social Voice Communicator Implementation Guide, Release 6.1

E29461-01

Copyright © 2011, 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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

Contents

3 Configuring Charging for Social Voice Communicator Services

4 Provisioning and Managing Social Voice Communicator Services

5 Monitoring Social Voice Communicator Services

6 SVC Social Voicemail Call Flows

7 About the SVC API Operations

8 Social Voice Communicator API Reference

Preface

This document describes how to install, configure, administer, and use the Oracle Communications Service Broker Social Voice Communicator (SVC) features.

Audience

This document is intended for system administrators, developers, and system integrators who set up or administer SVC.

This documentation assumes that you are familiar with:

- Service Broker concepts. For more information see *Oracle Communications Service Broker Release 6.1 Concepts Guide*
- The operating system on which your system is installed
- Telecom networks and protocols, especially SIP and SS7-based protocols
- Clustering concepts and life cycle management
- Editing Java MBeans
- Web servers and protocols, especially RESTful

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Communications Service Broker Release 6.1 documentation set:

- *Oracle Communications Service Broker Release Notes*
- *Oracle Communications Service Broker Concepts Guide*
- *Oracle Communications Service Broker Installation Guide*

- *Oracle Communications Service Broker System Administrator's Guide*
- *Oracle Communications Service Broker Signaling Server Units Configuration Guide*
- *Oracle Communications Service Broker Modules Configuration Guide*
- *Oracle Communications Service Broker VPN Implementation Guide*

Downloading Oracle Communications Documentation

Oracle Communications Service Broker documentation is available from the Oracle software delivery website:

http://edelivery.oracle.com/

Additional Oracle Communications documentation is available from Oracle Technology Network:

http://www.oracle.com/technetwork/index.html

1

Social Voice Communicator Overview

This chapter introduces Oracle Communications Service Broker Social Voice Communicator (SVC) application.

About SVC

You use SVC to offer these social communication products to your subscribers:

- Oracle Communications Social Voice services that allow your subscribes to set up preconfigured call conferences for public or private groups.
- Oracle Communications Social Voicemail voice-mail services.
- Oracle Communications Social Circle services that allow your subscribers to manage communication to all of their MSISDN numbers through a single contact number.

You can combine and offer some or all of these services to your customers in service tiers as complex or simple as your implementation requires. You can provide these services subscribers over both SIP and SS7 networks.

A click-to-dial feature is included as part of the SVC Social Voice service. It allows your subscribers to place VOIP calls from within the web application that hosts their services.

You implement the SVC Social Voice, Social Circle, and Social Voicemail services as web applications, which gives you the flexibility to integrate them with any other web-based services. For example you can use SVC to create a social networking voice feed.

You give your subscribers access to the web application, or integrate it with other services that you provide. SVC includes an extensive RESTful API that you use to provision your services and provide access to them. Most of the API operations are executable by your subscribers and are intended to be part of a design that allows your subscribers to self-administer their accounts.

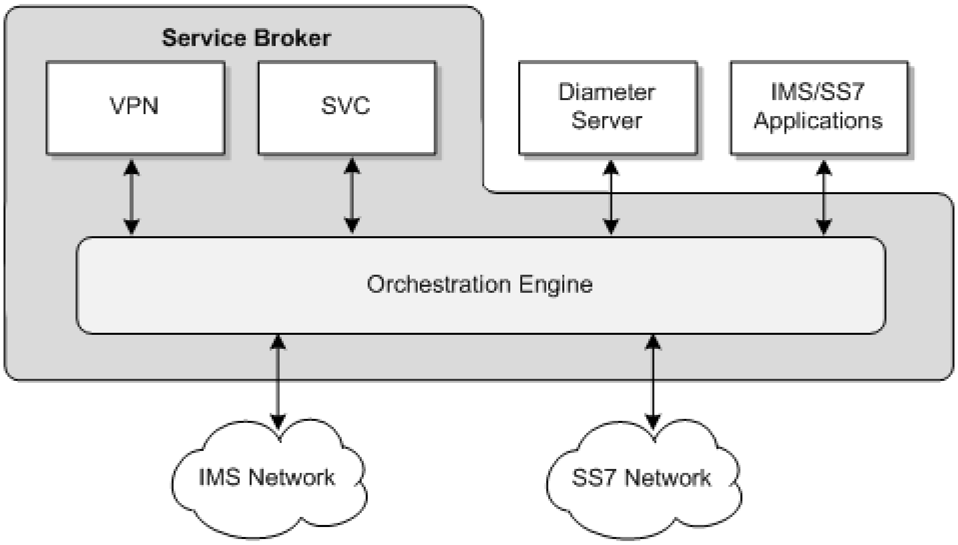
To charge for services, SVC supports both event and session-based Diameter charging. You provide a Diameter charging engine and configure it to work with SVC through an MBean interface. Both Diameter Rf (offline) and Ro (online) charging are supported. However, online charging is limited to calls to subscribers.

SVC is preconfigured to use the Oracle Communications Billing and Revenue Management server for charging, but you can use any Diameter-based charging application for this purpose.

Most of the SVC services require that you obtain a media server to play and record announcements, such as voice-mail greetings and messages, and top-up charging messages.

Figure 1–1 shows how the VPN and SVC features interact with Service Broker and your other network elements.

Figure 1–1 SVC and VPN in Service Broker



Understanding the SVC Features

This section describes the SVC features that your subscribers can use for personal communication. Administrators can perform all of these same actions, but it is more efficient to let your subscribers perform most or all of them themselves. You do this by creating a web application (self-service website) and using the SVC API operations to provide these services. The list of actions that you allow your subscribers to perform depends on your organization’s policies. The following sections provide more detail on these actions.

Service Broker treats the SVC Social Circle, SVC Social Voice, and SVC Social Voicemail services as separate applications, like any other service that you add to Service Broker. And like other Service Broker services, they are available to your subscribers using either SIP or SS7 networks.

Using Social Circle to Create an Address Book of Personal Contacts

Your subscribers can organize their personal or professional contacts into their own address book. These personal contacts can be individual contacts or can be added to SVC Social Circle groups. The subscriber can then create access (policy) rules that apply to the individuals or to the entire group.

The SVC Social Circle service allows your subscribers to use one of their contact phone numbers as a “one number” point of contact for all their other phone numbers. Subscribers have the option to:

- Direct calls to the one number from any one of their other phone numbers.
- Specify the incoming call to ring sequentially through all of their contact numbers in order, or to all contact numbers simultaneously.

- Specify their own access rules based on time of day to limit the connection between their other numbers and the one number.

The SVC Social Circle one number is presented to all incoming calls.

Create Intelligent Routing Rules for Group Access using Social Circle One Numbers

Subscribers use the SVC Social Circle API to determine who has access to their contact numbers and at specific times. Subscribers use this feature to create contact groups with similar access needs, and direct their calls to a single number (one number). The subscriber then creates intelligent routing rules to define how the one number treats calls from the different groups.

For example a subscriber might group their fixed home phone number, fixed work phone number, and cell phone number as a one number. The subscriber then creates access groups for co-workers, family, friends, and unidentified callers, and specifies routing rules like these to control calling access:

- Calls from the co-worker group are accepted during working hours (say, 08:00-17:00) and ring first to the fixed work phone and then to the cell phone number. If neither answers, the call is sent to voice-mail. After working hours, calls are sent to voice-mail
- Calls from family group are accepted during all hours. During working hours, the call firsts rings just on their cell phone. At all other hours, calls from family cause all phones ring simultaneously.
- Calls from the friends groups are accepted during all hours, however, during work hours, the call only rings on their cell phone. After working hours the call first rings at the home phone, then the cell phone.
- Calls from the unidentified group are sent to voice-mail at all hours.

Subscribers might also specify a “do not disturb” period during which no calls are routed to their phones.

Routing rules also allow subscribers to set the amount of time each phone rings in sequence. These rules are based on the time of day and day of week. You can create more intricate routing scenarios by specifying priorities for each of the routing rules.

Using a Social Voice Group Number to Create Conference Calls

SVC Social Voice is a collaborative tool that your subscribers use to create instant conference calls. Any SVC Social Voice group member can initiate a conference call with all other members of the group with a single call to a group number. These groups can be open to all subscribers in your implementation (public) or strictly controlled by the group owner (private).

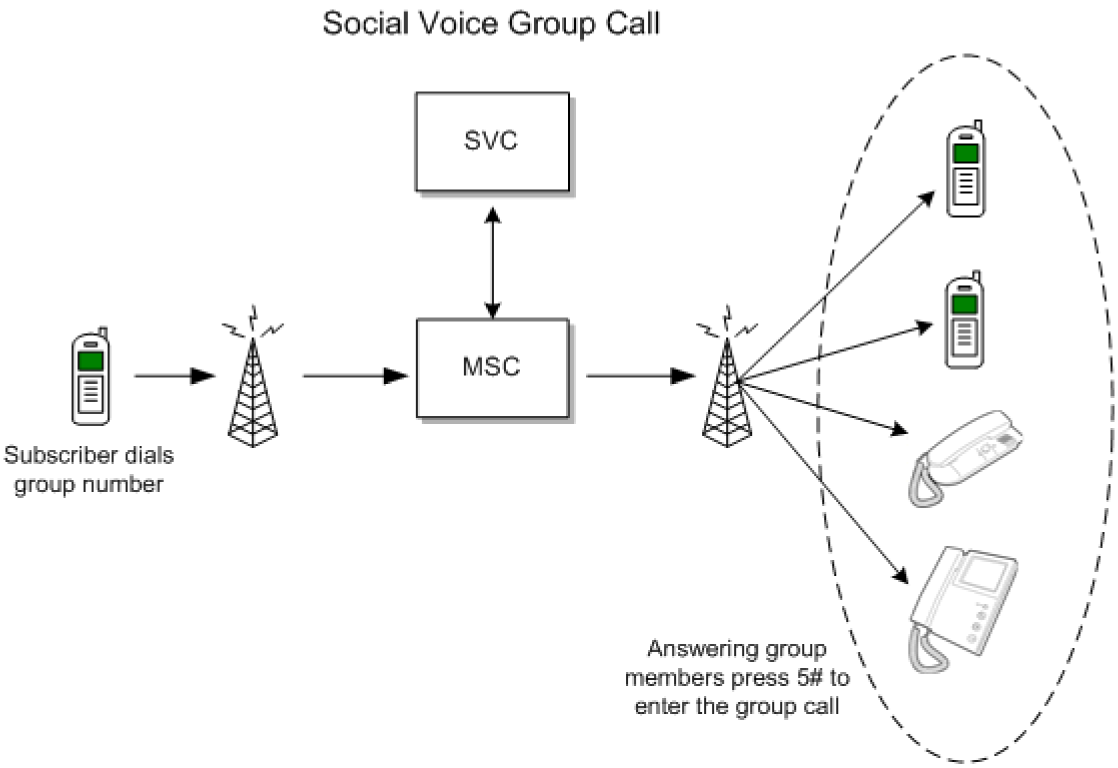
Any subscriber in your implementation can create a public group, and any other subscriber in your implementation can join one. Public groups are intended for less formal communication, such as a company sports team group.

Memberships in SVC public groups is open to anyone in your SVC implementation. Memberships in SVC private groups is strictly controlled, and only the group owner can add members.

Your subscribers use a group number as the single point of contact for all group members. Calls to the SVC Social Voice group number can ring to all of the other group numbers sequentially, or at once. When callers answer, they are prompted to press **5#** to enter the Social Voice conference. The group number is presented on all of the ringing phones. Figure 1–2 illustrates a typical SVC Social Voice call.

The group number for both public and private groups can be a "real" MSISDN number or a virtual web-based number.

Figure 1–2 A Typical Social Voice Call


Social Voice Group Call
SVC
MSC
Subscriber dials group number
Answering group members press 5# to enter the group call

Configuring and Using Web-Based Voice-Mail

The SVC Social Voicemail service offers web-based and phone-based voice-mail services to your subscribers. Using SVC Social Voicemail, you typically set up a website that your subscribers use to create, view, listen to, and share their voice-mail messages as WAV files. You can also offer subscribers the flexibility to manage their own voice-mailbox settings, such as creating their own greeting messages from the web interface. Because it is web-based, the underlying operations allow you to integrate a voicemail service with other products, such as a social media feed.

Subscribers would typically use the web interface to send all calls to voice-mail except for those from a specific group, which ring through. Subscribers can use keypad options to set the number of rings, change their password, and perform other tasks.

The voice-mail operations provided by the SVC API include:

- Activate and deactivate voice-mail access.
- Retrieve a list of all voice-mail accounts.
- Retrieve settings for a voice-mail account.
- Set, change, deactivate, and delete a personal greeting message.
- Retrieve and delete voice-mail messages.
- Retrieve a list of voice-mail messages.

- Retrieve details for a voice-mail message.
- Send a voice-mail message to other subscribers.
- Upload a voice-mail message.

The SVC Social Voicemail services require that you obtain a media server to play voice-mail message files.

See "About SVC Social Voicemail" for illustrations of the SVC Social Voicemail call flows.

Make Calls Over the Web

SVC API includes a Click-to-dial operation that allows subscribers to place calls from a web server. You can use this operation to originate calls from the subscriber user portal.

About SVC Charging

SVC supports the Diameter offline (Rf) and online (Ro) protocols using the Oracle BRM product as the charging engine. Offline charging can be either session or event-based. Online charging can be session based. Both online and offline charging are available for calls and events that originate in either SIP or SS7-based networks.

Using a Media Server to Serve and Record Messages

SVC services are designed to interact with a media server to provide audio messages such as "welcome" messages, recorded greetings, explanations of configuration options, and so on. Any MSCRM (RFC 5022)-compliant media server works.

Configuration Tools and Interfaces Overview

You make SVC services available to your subscribers by creating a web server that uses the SVC API. Typically, subscribers provision their own accounts using these RESTful operations through the web subscriber interface, but you can also use the SVC **batch_load** and **batch_check** command-line tools to provision and administer accounts in bulk.

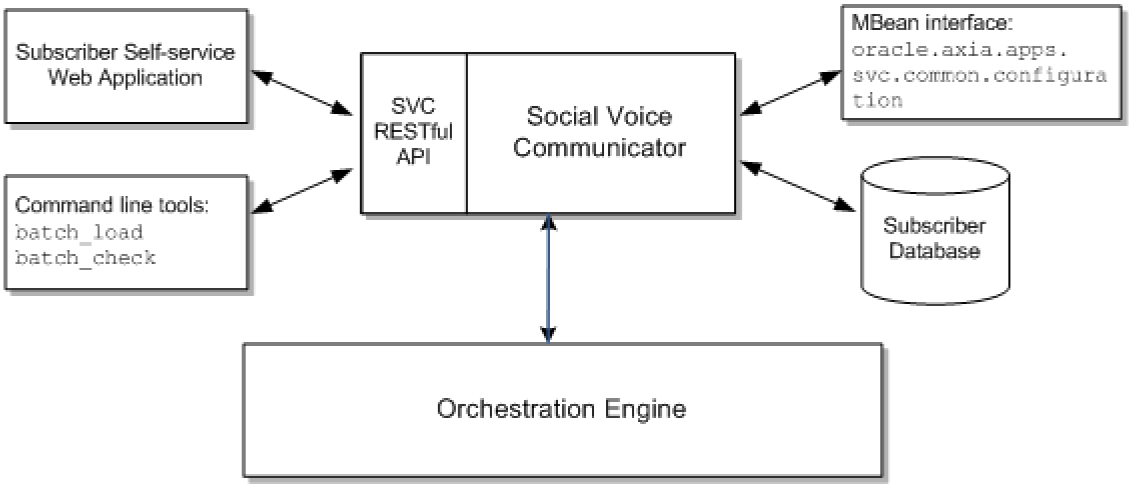
You configure SVC to accept incoming IMS and SS7 traffic by configuring the IMs and SSUs within Service Broker. You configure the SVC services themselves using its MBean interface.

For concepts and configuration information about SSUs, IMs, Diameter charging, and the Orchestration Engine, see these documents:

- *Oracle Communications Service Broker Concepts Guide*
- *Oracle Communications Service Broker Configuration Guide*
- *Oracle Communications Service Broker Release 6.1 Signaling Server Units Configuration Guide*
- *Oracle Communications Service Broker Release 6.1 Modules Configuration Guide*

Figure 1–3 shows the general access points for the configuration tools and other components in a PCM implementation.

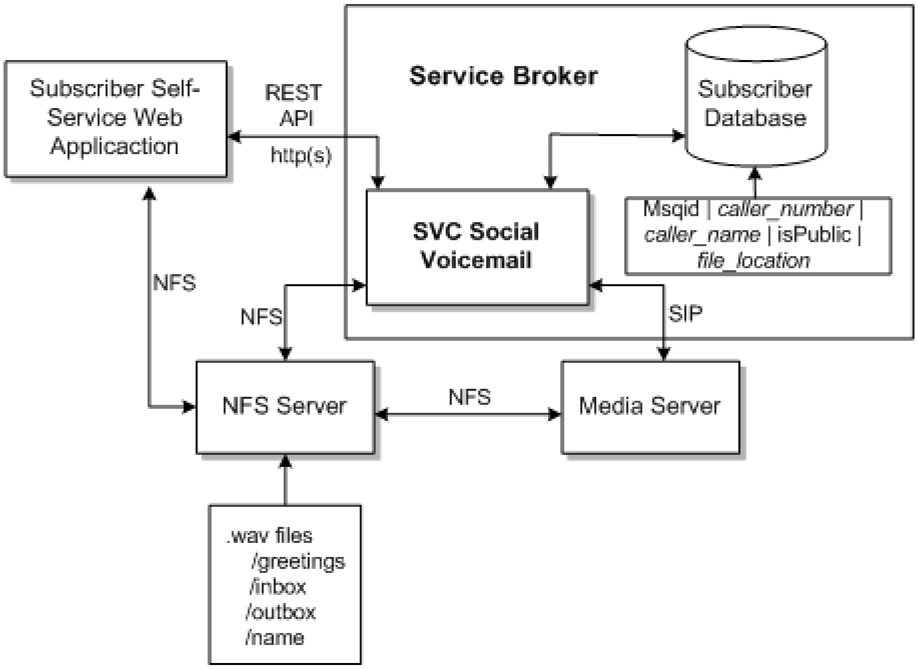
Figure 1–3 Configuration Tools and Clients



About SVC Social Voicemail

The SVC Social Voicemail service offers web-based voice-mail services to your subscribers. Figure 1–4 shows how the various components related to Social Voicemail relate.

Figure 1–4 Social Voicemail Components



Both Oracle Communications Social Voicemail greeting messages and voicemail messages are stored as WAV files on an network file system (NFS) server, which Service Broker, the media server, and the subscriber self-service web portal can all access. The subscriber database stores message records that the web portal uses to find the locations of the WAV files.

Voice-mail *call flows* set the order of the options and messages that callers hear in the voicemail system. For details on the SVC Voicemail call flows see "SVC Social Voicemail Call Flows".

2
Setting Up Social Voice Communicator

This chapter explains how to set up Oracle Communications Service Broker to use Social Voice Communicator (SVC).

Set Up Tasks Overview

Configuring the SVC Social Voice, SVC Social Circle, and SVC Social Voicemail services requires these general steps:

- Configure Service Broker features to work with SVC services. See "Configuring Service Broker for SVC Services" for more information.
- Configuring the SVC services themselves. These configuration tasks are covered in "About Configuring SVC Services" and "Configuring SVC Services".
- Configuration tasks that vary by SVC service. Table 2–1 lists the SVC services that require these additional configuration tasks:
  - Configuring SVC Services to Place Calls
  - Configuring Voice-Mail for SVC Services
  - Configuring the Social Circle Click-to-Dial Service
- Creating a user portal that your subscribers will use to self-administer their accounts. See "Configuring the Social Circle Click-to-Dial Service" for more information.
- (Optional) Configure charging. See "Configuring Charging for Social Voice Communicator Services" for more information.

Table 2–1 lists the configuration tasks required by each SVC service. The following sections explain the configuration tasks.

Table 2–1 Required SVC Service Configuration Tasks

Configuration Task	SVC Social Voice	SVC Social Circle	SVC Social Voicemail	SVC Social Circle Click-to-Dial
Configuring Service Broker for SVC Services	Yes	Yes	Yes	Yes
Configuring SVC Services	Yes	Yes	Yes	Yes
Configuring SVC Services to Place Calls	No	Yes	Yes	No

Table 2–1 (Cont.) Required SVC Service Configuration Tasks

Configuration Task	SVC Social Voice	SVC Social Circle	SVC Social Voicemail	SVC Social Circle Click-to-Dial
Configuring Voice-Mail for SVC Services	No	No	Yes	No
Configuring the Social Circle Click-to-Dial Service	No	No	No	Yes

Configuring Service Broker for SVC Services

This section explains how to configure Service Broker to use SVC services. The configuration tasks include:

- Installing the SVC Software
- Securing the SVC RESTful Interface
- Configuring Data Persistence for SVC Services
- Securing Subscriber Credentials for SVC Services
- Opening an HTTP Listening Port for SVC Clients

The following sections describe these tasks.

Installing the SVC Software

You install the SVC software by selecting it as an included component of the SVC Service Broker installation in either with the graphical user interface, or as part of a silent installation.

In Graphical Installation Mode, you include the SVC option by selecting it in the **Available Product Components** page of the installation sequence.

When using the silent installer script, include the SVC software in the installation by adding **socialvoice_communicator** to the **DEPENDENCY_LIST** list of values.

For more information on installing Service Broker software, see the *Oracle Communications ServiceService Broker Installation Guide.*

Deinstalling the SVC Software

Once installed, you can remove the SVC with the Service Controller software by starting the Oracle Universal Installer and clicking the **Deinistall Product** button. In the product list, choose the **Service Controller** as the component to remove under the Service Broker product item. To remove the SVC application alone, use the **runinstaller** program with the **-deinstall socialvoice_communicator** options. See the *Oracle Communications Service Broker Installation Guide* or details on using the **runinstaller** program.

Securing the SVC RESTful Interface

SVC exposes a RESTful API that administrators use to configure and provision SVC services. In test and evaluation implementations, it may not be necessary to secure the connection to the RESTful facade. However, the API connection must be secured for production implementations.

The security features available for securing the API connection include Secure HTTP (HTTPS) and SSL-based client certification authentication.

Note: to SSL-based client certificate authentication, SVC authenticates client requests based on user name and password values passed in HTTP headers of API requests. See "Authentication and Authorization" for more information.

API connection security is governed by the common security settings for the Service Broker domain. Therefore, to secure the connections to the SVC RESTful API, follow the instructions for applying security to Service Broker as described in *Oracle Communications Service Broker Security Guide*.

By default, the security settings in a new installation of the Service Broker software are enabled.

The following Service Broker properties control the connectivity between the RESTful API clients and the SVC API facade:

- **org.eclipse.equinox.http.jetty.https.enabled**: Set to **true** to use HTTPS for the REST API connection.
- **org.eclipse.equinox.http.jetty.https.ports**: Specify a port number to use for the secured HTTP connection.

Set these properties in the properties files located in *Oracle_home***/ocsb61/admin_server/properties**:

- For a web domain, edit the **admin.properties** file
- For a hosted domain edit the **hosting.properties** file:

See the *Oracle Communications Service Broker Security Guide* and *Oracle Communications Service Broker System Administrator's Guide* for complete information on securing the Service Broker installation using the properties files.

Creating the SVC Domain

The Service Broker SVC services operate as components of the Service Broker Processing Domain, so create a processing domain or unified domain before configuring and using SVC services.

When prompted for a domain type to install, select **Social Voice Communicator**. You choose the domain type by entering the index number that corresponds to your selection, as indicated in the script prompts.

Note: SVC services must be run in a domain that uses the service availability service mode. Ensure that this option is selected when you create the domain.

For information about creating Service Broker domains and detail about the service modes, see the *Oracle Communications Service Broker Installation Guide*.

Configuring Data Persistence for SVC Services

Before using the SVC API to configure SVC services, set up the database that SVC uses to store its application data. The application data includes subscriber information, Social and SVC Social Voice contacts, and statistical data generated by the operations of the SVC services.

SVC works with these Oracle Database versions:

- Oracle Database 10g Release 2
- Oracle Database 11g Release 1 or Release 2

Note that SVC services do not support Oracle Coherence in-memory storage or Oracle Berkeley DB file-based storage. Use an Oracle Database for persistence.

To configure data persistence, you first set up the database. You then configure the connection to the database in the Service Broker configuration, as described in the following sections.

Preparing the Database

SVC includes a database configuration script that sets up the tables and schema required for SVC services. The script invokes other SQL scripts, which perform the actual configuration of the database.

A SQL script is available for each type of SVC service: SVC Social Voice, SVC Social Circle, and SVC Social Voicemail, a SQL script exists for creating the statistical data tables and schema that are common to all services.

Only run the SQL scripts that are applicable to your implementation. For example, if using SVC services only, you only need to run the SVC SQL script and statistical data script.

Before configuring data persistence, make sure the database is available by network to the Service Broker. Oracle recommends that you create a user account on the database specifically for the Service Broker’s use, and another for SVC.

To set up the database:

1. Open the following properties file for editing:

   Oracle_home/**ocsb61/admin_server/properties/create_db_table.properties**

2. Configure the profile database settings, as follows:

   - **profile.db.server** - The IP address of the database server.
   - **profile.db.port** - The port on which the database listens for TCP client connections.
   - **profile.db.dbname** - The name of a database to store data. This is the database in which the script creates and formats tables for the service and statistical data.
   - **profile.db.user** - The name of a database user with privileges to access this database.

   For example:

   ```
   profile.db.server=10.10.1.125
   profile.db.port=1521
   profile.db.dbname=orcl
   profile.db.user=ocsb
   ```

3. Save and close **create_db_tables.properties**.

4. Move any SQL scripts not relevant to your implementation from the SQL script directory to a backup location. The scripts are located in:

   Oracle_home/**ocsb60/admin_server/scripts/ddl**

   The database configuration script invokes any SQL script it finds in this directory.

   The scripts are:

   - **er_ons.sql**: Configures tables in the database for the SVC Social Circle service.
   - **er_stats.sql**: Configures tables in the database for statistical information generated by activities of the services.
   - **er_voicemail.sql**: Configures tables in the database for the SVC Social Voicemail service.
   - **er_vpn.sql**: Configures tables in the database for the VPN service.

5. Run the database configuration script:

   Oracle_home/**ocsb61/admin_server/create_db_table.sh**

   The script connects to the database using the setting you specified in **create_db_table.properties**.

   At the following prompt, enter the password associated with the database user you identified in the **create_db_table.properties** file.

   ```
   Please input password of profile database user
   ```

The database configuration script prints status or error messages to the screen. If successful, you can now use the SVC API and batch operation tools.

See "Understanding the SVC Data Model" for more information on the data objects.

Configuring the Database Connection

After preparing the database, create the connection to the database in the Service Broker domain using the Administration Console.

You must know at least the following values for your database:

- **CredentialKey**: Use **KEY**, the default credential key name assigned to the profile database user by the database configuration script.
- **name**: Use **oracle_driver** for the connection name, unless you have modified the default driver reference attribute for the SVC application.
- **connectionFactory**: In most cases, this should be the built-in driver, **oracle.jdbc.pool.OracleDataSource**.

For complete information on configuring the database connection, see the discussion on post installation tasks in *Oracle Communications Service Broker Installation Guide*.

Securing Subscriber Credentials for SVC Services

If your SVC implementation includes storing user credentials, such as passwords for voice-mail accounts, configure the Service Broker credential store feature to store those credentials. For more information see the discussion on securing credentials with credential store in *Oracle Communications Service Broker Security Guide*.

Opening an HTTP Listening Port for SVC Clients

To make the SVC API accessible to clients, create an HTTP access point in the Service Broker Signaling Tier:

1. Open the Administration Console.
2. Navigate to **OCSB**, **Signaling Tier**, **SSU Web Services**, then **General**.

   The General page appears
3. Click the **HTTP** tab, then **Server**, then **Network Access**.
4. Click the Edit icon.
5. Click **New**.

   The **New** HTTP window appears.
6. Fill in details for the new HTTP access point:

   For more information about the Credential Store settings see the discussion on securing credentials with Credential Store in *Oracle Communications Service Broker System Administrator's Guide*.

   - **Server Address** - The host name or IP address of the server to bind to. Leave blank if using a local host.
   - **Server Port** - A server port number to use.
   - **Protocol** - Most production implementations use HTTPS.
   - **SSL Client Auth** (if using HTTPS) - Boolean. Whether to enable SSL client authentication for this port.
   - **Keystore Id** (if using HTTPS) - The Credential Store Keystore ID.
   - **Truststore Id** - The Credential Store Truststore ID.
   - **Target** - The target server. If blank, applies to all servers.
7. Click **OK**.
8. Select the **Security Context** subtab.
9. Click the **Edit** icon.
10. Select the **Keyid 0 and Context URI /** line.
11. Click **Update**.

    The **Update** window appears.
12. Change the value for **Auth Method** to **NONE**.
13. Click **OK**.
14. Click Confirm.

About Configuring SVC Services

This section explains how to set up the SVC services to work with Service Broker, and your networks. All SVC services require these configuration tasks.

This section assumes that you have followed the instructions in the "Configuring Service Broker for SVC Services" and installed and connected Service Broker for SVC services.

Configuring SVC Services

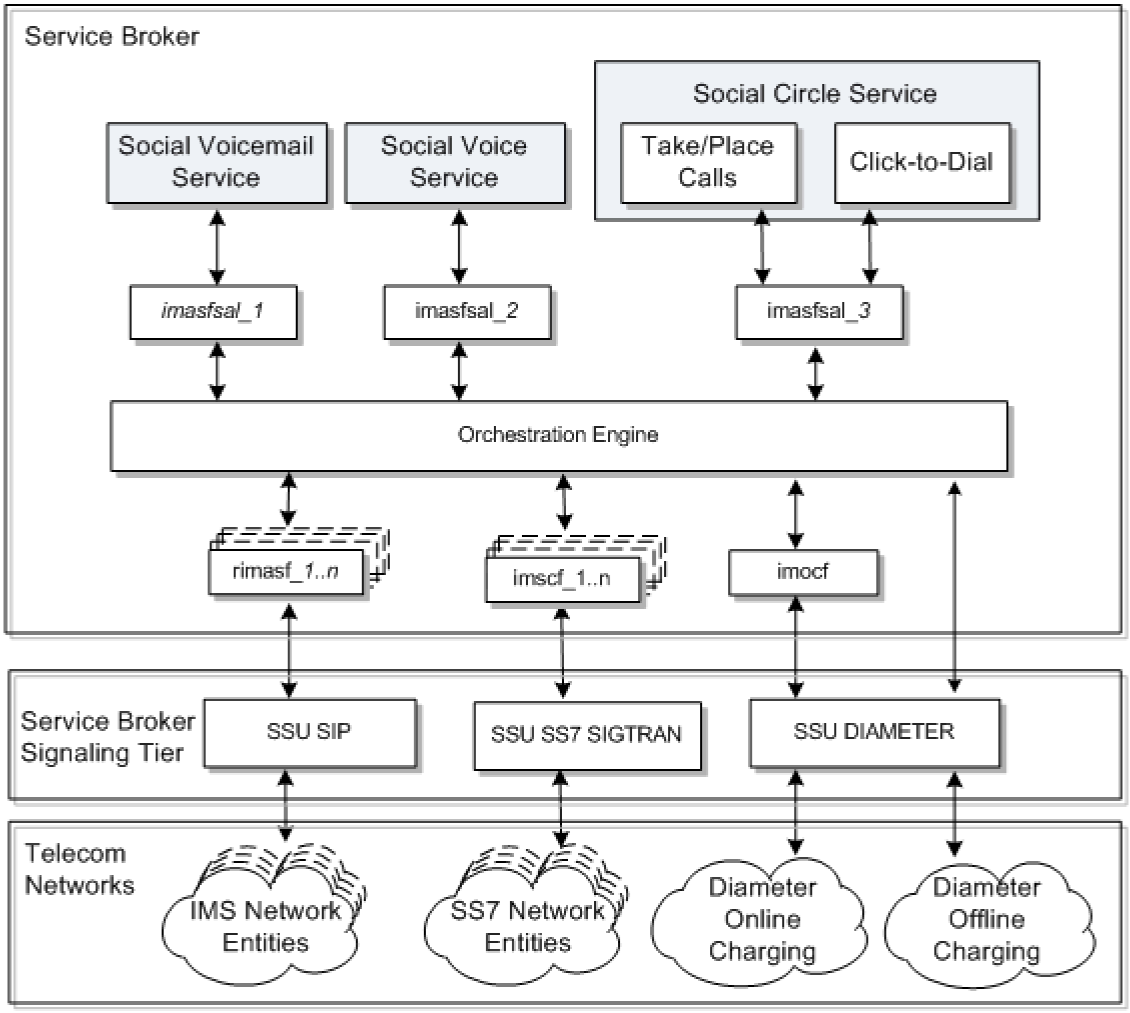
This section lists the configuration tasks required by all SVC services. Table 2–1 lists the additional configuration tasks that each SVC service requires.

Configuring SVC service include the following tasks described in this section:

- Connecting SVC Services to Service Broker
- Connecting SVC Services to Telecommunications Networks
- Configuring a Media Server for SVC Services
- Configuring SVC Services to Take Calls

Figure 2–1 shows an overview of the various Service Broker components of a working SVC implementation and the various components required.

Figure 2–1 SVC Configuration Overview



Connecting SVC Services to Service Broker

All SVC services require application-facing IM-ASF-SAL IMs to connect to the Orchestration Engine, and network-facing IMs to connect them to your telecom network. Follow these steps for each SVC service your implementation requires within a domain:

1. Open the Administration Console and click **Edit**.
2. In the OCSB navigation tree, expand **Processing Tier**, then **Interworking Modules**, then **IM Management**.
3. Click **New**.
4. Create a new IM-ASF-SAL to route SIP messages from the SIP SSU to the Orchestration Engine. Each IM-ASF-SAL requires a unique name. For more information, see the discussion on managing Interworking Modules in *Oracle Communications Service Broker Modules Configuration Guide*.
5. Configure the new IM-ASF-SAL for the SVC Service.
6. Select the new IM-ASF-SAL from the MBean navigation tree. Select **OCSB**, then **Processing Tier**, then **Interworking Modules**, then *IM-ASF-SAL_name*.

   The **Configuration** pane appears.
7. Click **Application Server**.
8. Enter the SVC service address in the **SAL Application Address** box using this syntax:

   ```
   sip:SVC_application_name@oracle.com
   ```

   Where:

   - *SVC_application_name* - Is the name of your SVC application to connect to. You get this name from the appropriate MBean. For example, SVC Social Circles application name, **sip:socialcircles@oracle.com**, is available from the **applicationName** attribute of the **personalsvc** configuration MBean:

     oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.personalsvc.configuration,version=6.1.0,name0=personalsvcConfig,name1=general,name2=applicatonName
9. Set the **SAL Mode** to **External**.
10. Click **Apply**.

    For more information on the remaining configuration settings see the discussion on configuring IM-ASF-SAL in *Oracle Communications Service Broker Modules Configuration Guide*.
11. Commit the new IM-ASF SAL.

Connecting SVC Services to Telecommunications Networks

The services provided by SVC work over both SS7-based and IMS-based networks. To expose SVC services to those networks, connect SVC to the network-facing IMs which in turn, connect to external networks as described in the following sections. Figure 2–1 shows an overview of the required components; the sections below provide instructions for creating and configuring them.

Connecting SVC Services to SIP Networks

You use R-IM-ASF, SSU SIP, and Orchestration Engine routing rules connect SVC services to a SIP-based telecom network.

Each network node requires SSU SIP and R-IM-ASF.

Configure SVC services with a SIP network:

- For each network node in your implementation, create, activate, and configure an R-IM-ASF to route SIP traffic between the Orchestration Engine and the SIP SSU.

  For more information, see the discussion on managing Interworking Modules, and configuring R-IM-ASF in *Oracle Communications Service Broker Modules Configuration Guide*.

- Add iFC routing rules to the Orchestration Engine to route traffic to and from the SIP network nodes.

  For more information on creating the iFC, see "Sample iFC Routing Rules" and *Oracle Communications Service Broker Orchestration Studio User's Guide*.

- Configure the SIP SSU to accept calls from your SIP network nodes and route them to the R-IM-ASF IMs that you created. Route all traffic from your network notes through the default SIP SSU.

  For more information, see the discussion on configuring SIP Signaling Units in *Oracle Communications Service Broker Signaling Server Units Configuration Guide*.

In addition, you must also change the default **SessionIDOverride** setting from **false** to **true**. You can set this attribute using the Service Broker JConsole or any JMX client, or even run scripts on the MBeans directly.

Set the **SessionIDOverride** attribute of the following MBean to **true**.

MBean: **com.convergin:Type=SipSsu,Version=6.1.0,Location=AdminServer,Name=ssusip.ssusip**

Connecting SVC Services to SS7 Networks

This section explains how to set up Oracle Communications Service Broker to make SVC services available to SS7 telecom networks.

Your SVC services require network-facing IMs and Orchestration Engine routing rules to connect your SVC service to a SS7-based telecom network.

Both the Oracle-supplied SVC services and other services that you connect them with, require connections from your SS7 network nodes to the Service Broker Orchestration Engine. You create these connections by configuring the SSU SS7 and creating IM-SCF and RIM-SCF IMs. You configure IM-SCF to route traffic from SIP network entities to the SSU SS7 and then to the Orchestration Engine. To route traffic the from the Orchestration Engine to the SIP network notes, configure an RIM-SCF. Figure 2–1 shows these relationships.

Each network node requires an individual IM-SCF, RIM-SCF, or both.

To connect SVC services to a SIP network:

- For each network node in your implementation, create, activate, and configure an IM-SCF to route SIP traffic from SS7 network nodes to the Orchestration Engine. At a minimum, specify these **Configuration** settings:

  - **General: Alias**
  - **Call Handling**: **Body Encoding Format**
  - **Media Resources**: Add the media resources that your implementation requires.

    For more information on the configuration settings, see the discussion on configuring IM-SCF in *Oracle Communications Service Broker Modules Configuration Guide*.

- For each network node in your implementation, create, activate, and configure an RIM_SCF to route SS7 traffic from the Orchestration Engine to the SSU SS7 SIGTRAN.

  For more information, see the discussion on managing Interworking Modules and configuring RIM_SCF in *Oracle Communications Service Broker Modules Configuration Guide*.

- Configure the **SS7 SSU** to accept traffic from your SS7 network nodes and route it to the RIM_SCF IMs that you created. The SSU SS7 SIGTRAN settings are configured on the Administration Console by opening **OCSB**, then **Signaling Tier**, then **SSU Web Services**, then **SS7 SSU**. These configuration settings are the minimum required for an SVC Service; your implementation may require others:

  – **SSU SS7**: **General**: Configure the SSU to connect to your SS7 stack.

  – **M3UA: Local Point Code**

  – **M3UA: Connectivity:** Add at least one **Local System** entry and one **Remote System** entry.

  – **M3UA: Connectivity: Network Mapping**

  – **M3UA: Connectivity: Network Routing**: Configure the routes to your network entities.

  – **SCCP: Local SSNs**

  – **SCCP: Remote PC and SSN Addresses**

  – **Routing: Incoming Routing Rules**: Add a rule that accepts traffic from the IM-SCF.

  – **Routing: Incoming Routing Criteria**: Add a rule that routes calls by the **Alias** entry you created in the IM-SCF.

  For more information on the configuration settings, see the discussion on configuring SS7 Signaling Server Units for SIGTRAM in *Oracle Communications Service Broker Signaling Server Units Configuration Guide.*

- Add routing rules to the Orchestration Engine to route traffic to and from SIP network nodes. For each SVC Service, enter the IM-ASF-SCF that you created in the **Default Routing Targets** box.

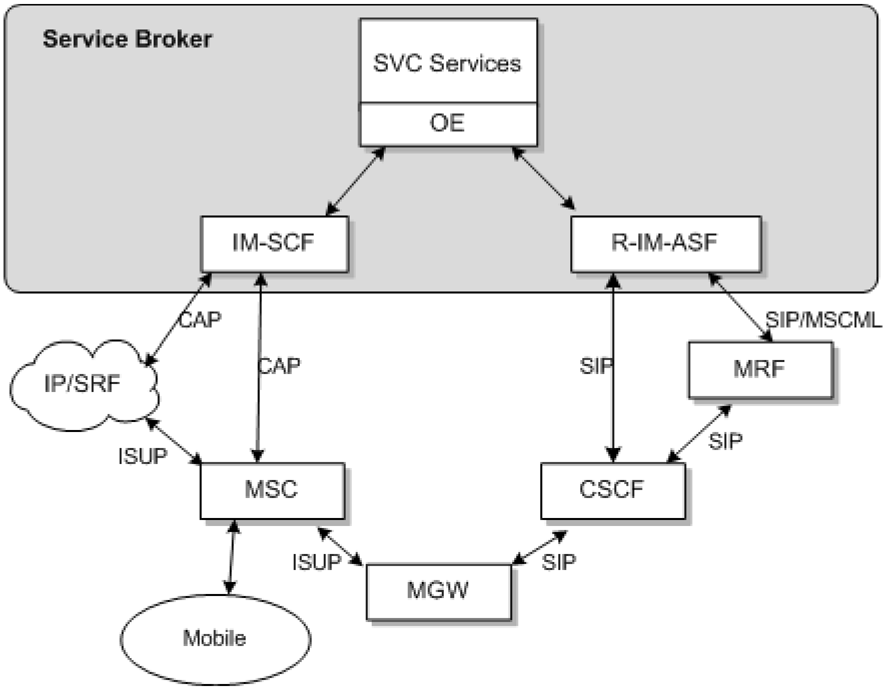
  For more information see the discussion on configuring SS7 Signaling Server Units for SIGTRAN *Oracle Communications Service Broker Signaling Server Units Configuration Guide*.

Configuring a Media Server for SVC Services

Your media server configuration depends entirely on your individual implementation. Figure 2–2 shows a possible SVC deployment that illustrates the flexibility you have in working with media services over telecom networks. This SVC implementation works on both IMS and SS7 networks. This implementation assumes that all subscribers share the same MRF.

> **Note:** Netann media servers are not supported.

Figure 2–2 Example Media Server Configuration



To configure a media server to work with your SVC implementation:

1. If not running, start the Administration Console.

   For details, see the *Oracle Communications Service Broker Installation Guide*.

2. Using a JMX MBean browser, such as the Java Monitoring and Management console (JConsole) tool, connect to the Administration Console process.

   See the *Oracle Communications Service Broker System Administrators's Guide* for information on connecting to the Administration Console process to access configuration MBeans.

3. In the MBean browser, navigate to the voicemail configuration

   MBean: oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0,name0=svcConfig,name1=callControl,name2=mediaServer

4. Specify an email address to receive voice-mail and email notifications in the **mediaServer:type** parameter.

Configuring SVC Services to Take Calls

This section describes the MBean attribute that you use to enable SVC services to take telephone calls. Table 2–1 lists the SVC services that require this configuration task.

You can set this attribute using the Service Broker JConsole or any JMX client, or even run scripts on the MBeans directly.

In the **hostedDomain** attribute of the following MBean, add the SVC service domain home you created.

MBean:
oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.commo

n.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=hostedDomain[0]

Configuring SVC Services to Place Calls

This section describes the MBean attributes that enable SVC services to place telephone calls and access voice-mail features. Table 2–1 lists the SVC services that require this configuration task.

This sections assumes that you have acquired a media server to use and know the media server's listening port and IP address.

You can set these attributes using the Service Broker JConsole or any JMX client, or even run scripts on the MBeans directly.

To configure SVC services to process calls and use voice-mail services:

1. In the **hostedDomain** attribute of the following MBean, add the SVC service domain home you created.

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=hostedDomain[0]

2. In the **mrfAlias** attribute of the following MBean, change **mrfAlias** to the Media Resource Function alias, usually with this syntax *mrf_identifier***@ocsb.com.**

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=mrfAlias

3. In the **ip** attribute of the following MBean, specify the IP address of your media server

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=mediaServer,name3=ip

4. In the **port** attribute of the following MBean, specify the port number to use on the media server:

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=mediaServer,name3=port

5. In the **type** attribute of the following MBean, specify the Media Server Type.

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.common.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=mediaServer,name3=type

6. In the **filePath** attribute of the following MBean, specify the root path to the media files. The syntax is: **file://***file_path*

   MBean:
   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.com

mon.configuration,version=6.1.0.SNAPSHOT,name0=svcConfig,name1=callControl,name2=mediaServer,name3=filePath

Configuring Voice-Mail for SVC Services

This section explains how to configure voice-mail for SVC services. Table 2–1 lists the SVC services that require this configuration task.

Before you configure voice-mail for SVC services with the following sections, you must complete these tasks:

- Acquire a media server and know the media server's listening port and IP address.
- Acquire an email server.
- Follow the steps in "Connecting SVC Services to SIP Networks" and "Connecting SVC Services to SS7 Networks".

Mounting the Voice-Mail Message Location and Media Server

This section assumes that you use a separate server for storing voice-mail messages, and know that server's IP address and message file location. To set up a server to store voice-mail messages:

1. On your SVC services server, select a local NFS location to mount audio files.
2. On your SVC services server open a command shell.
3. Mount the remote voice file location to a local location by using the following command syntax:

   Linux:

   ```
   mount -t nfs -o rw Audio_File_IP_Adress:/Audio_File_Path Local_Location
   ```

   Solaris:

   ```
   mount -F nfs rw Audio_File_IP_Adress:/Audio_File_Path Local_Location
   ```

   Where:

   - *Audio_File_IP_Address* is the audio file server identifier.
   - *Audio_File_Path* is the location of the audio files on the audio file server.
   - *Local_Location* is the SVC server location to mount the audio files.

   Example Linux command:

   ```
   mount -t nfs -o rw 239.255.0.0:/opt/data/voicedata /opt/oracle/svc/voicedata
   ```

   Example Solaris command:

   ```
   mount -F nfs 239.255.0.0:/opt/oracle/data/voicemail /opt/oracle/data/voicemail
   ```

   Then use `df -k` to confirm that the remote file location is mounted correctly.

4. Mount the media server to the same location with this command syntax:

   Linux:

   ```
   mount -t nfs -o rw SVC_Server_IP_Address:/Audio_File_Location Media_Server_
   location
   ```

Solaris:

```
mount -F nfs rw SVC_Server_IP_Address:/Audio_File_Location Media_Server_
location
```

Where:

- *Audio_File_IP_Address* is the audio file server location.
- *Audio_File_Location* is the path to the audio files on the audio file server.
- *Media_Server_Location* is the location to mount the media server.

Configuring Voice-Mail for Services

This section assumes that you have followed the instructions in "Configuring a Media Server for SVC Services" and added a media server to store voice-mail messages.

To configure voice-mail capability for an SVC service:

1. Set the SVC Server **umask** to **002**.

   This ensures that your SVC Server has permission to write files to the voice-mail directory.

2. On a command line as superuser, add the media server user to the root group with this command:

   ```
   #useradd root User_Name
   ```

   Where:

   - *User_Name* is the name of the media server user.

3. Add DNS server names and IP addresses to the **/etc/resolve.conf** file using this syntax:

   ```
   nameserver DNS_Server_IP_Address
   nameserver ISP_Server_IP_Address
   ```

   Where:

   - *DNS_Server_IP_Address* is the location of an DNS server to use.
   - *ISP_Server_IP_Address* is the location of an ISP server to use.

   For example:

   ```
   nameserver 208.67.222.222
   nameserver 208.67.222.220
   ```

4. Open the Administration Console and navigate to **OCSB**, **Processing Tier**, **Interworking Modules,** then **IM Management**.

5. If you have not created a new IM-ASF-SAL to route traffic from the Orchestration Engine to the SVC Social Voicemail server, do so now. For more information, see "Connecting SVC Services to Service Broker".

   For more information see the discussion on configuring R-IM-ASF in *Oracle Communications Service Broker Modules Configuration Guide*.

6. Select the new **IM-ASF-SAL** in the IM Management list.

7. Select **Configuration** then **Application Server**.

8. Enter the application alias of the SVC voicemail server in the **SAL Application Address** box using this syntax:

```
sip:Voicemail_Sever_Name@Domain_Name.com
```

Where:

- *Voicemail_Server_Name* is the name of the voicemail server you are adding.
- *Domain_Name* is the name of the domain where the server can be found.

9. Click **Commit**.
10. In the navigation pane select **OCSB**, then **Signaling Tier**, and then the SSU for your protocol.
11. Configure an incoming routing rule to the **IM-ASF** that you created. For more information see *Oracle Communications Service Broker Modules Configuration Guide*.
12. Navigate to **OCSB** then **Processing Tier** then **Orchestration Engine**.
13. Click **Configuration** then **Static Route OLP**.
14. Enter the URI of the SVC Social Voicemail service using this syntax:

    ```
    sip:IMASFSAL_Name.IMASFSAL@ocsb
    ```

    Where:

    - *IMASFSAL_Name* is the name of the IM-ASF-SAL you created for voice-mail.
15. Click **Commit**.
16. Configure these parameters using the Service Broker JConsole, or any JMX client or create scripts to run on the MBeans directly:

    a. Set the **SessionIDOverride** attribute to **true** in this MBean:

    com.convergin:Type=SipSsu,Version=6.1.0,Location=AdminServer,Name=ssusip.ssusip

    b. Set a location to store voice-mail files using the **localVoiceRootPath** parameter in this MBean. The syntax is /*parent_directory*/*child_directory*/..:

    localVoiceRootPath MBean in **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=voicemailsvcConfig,name1=general,name2=localVoiceRootPath**

    c. Specify a root location on the media server to store voice-mail messages using the **mediaServerVoiceRootPath** attribute of this MBean. Use this syntax: *filename*:///*parent_directory*/*child_directory*/...

    oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=voicemailsvcConfig,name1=general,name2=localVoiceRootPath.

    d. Specify an email server login account name to use with the **accountName** attribute of this MBean:

    accountName MBean in **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=voicemailsvcConfig,name1=emailNotification,name2=accountName**.

    e. Specify the email server login account password using the **setpassword** operation of this MBean. Set the **key** and **password** attributes to the key and password that the account uses, and set **one-way** to **False**:

Specify a **password** in the **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=CredentialStore**.

f. Specify an email address to send voice-mail and email notifications to the email server using the **emailAddress** attribute of this MBean:

   emailAddress MBean in **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0.name0=voicemailsvcConfig,name1=emailNotification,name2=emailAddress**.

g. Specify the IP address or domain name of the email server using the **mailServerHost** attribute of this MBean:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=voicemailsvcConfig,name1=emailNotification,name2=mailServerHost.

h. Specify the port to listen for email messages from the email server using the **mailServerPort** attribute of this MBean:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.voicemailsvc,version=6.1.0,name0=voicemailsvcConfig,name1=emailNotification,name2=mailServerPort.

Configuring Security Settings for Sending Voicemail Messages

By default, the SVC Social Voicemail service is configured to automatically send voice-mail messages that arrive in their voice-mail account to subscribers' email accounts in **.wav** format. To enable this capability, configure the security settings on both the email server and email clients, and set several Service Broker configurations.

You use the **keytool** program to generate and import keystore and server certificates; this program is available as part of the JDK that you installed with Service Broker.

For more information on keytool and keytool parameters, see the discussions on setting up the public key infrastructure between Administration clients and Processing Servers, and Signaling Servers, and about Keytool and X.500 distinguished names in *Oracle Communications Service Broker Security Guide*.

Perform these steps to configure email notification for SVC Social Voicemail service:

1. Generate and export HTTPS certificates and keystores using the **keytool** program as follows:

   a. Generate a public or private key for the server. This is an example command:

   ```
   keytool -genkeypair -dname "cn=Servername, ou=Organizational_Unit,
   o=Organization, c=US" -alias server -keypass Password -keystore Server_
   Keystore -storepass Password
   ```

   b. Export the public key using a self-signed certificate. This is an example command:

   ```
   keytool -exportcert -alias Server_Name -keystore Server_Keystore -storepass
   Password -rfc -file Server_Key_Filename.cer
   ```

   c. Import the server certificate into the client truststore. This is an example command:

   ```
   keytool -importcert -file Server_Key_Filename.cer -keystore Client_
   Truststore -storepass Password -noprompt
   ```

d. Generate the public or private key for the client. This is an example command:

   ```
   keytool -genkeypair -dname "cn=Axia Client_Name, ou=Organizational_Unit,
   o=Organization, c=US" -alias Client -keypass Password -keystore Client_
   Keystore -storepass
   ```

e. Export the public key to the client using a self-signed certificate. This is an example command:

   ```
   keytool -exportcert -alias Client_Name -keystore Client_Keystore -storepass
   Password -rfc -file Client_Certificat_Name.cer
   ```

f. Import the client certificate into the server truststore. This is an example command:

   ```
   keytool -importcert -file Client_Certificat_Name.cer -keystore Server_
   Truststore -storepass Password -noprompt
   ```

g. Copy the client keystore and client truststore files (*Public_Key_Filename***.cer** and *Client_Certificate_Name***.cer**) to these Service Controller locations:

   ```
   Oracle_home/install/admin_server
   Oracle_home/install/managed_server
   ```

h. Copy the Server_keystore and Server Truststore files to this Service Controller location:

   ```
   Oracle_home/install/managed_server
   ```

2. In your *Oracle_home***/ocsb61/admin_server/properties/common.properties** file: ensure that **axia.ssl=** is set to **true**, and that these entries name the correct files:

   ```
   javax.net.ssl.keystore=servsrkeystore
   javax.net.ssl.trustStore=servertruststore
   ```

3. Confirm that your *Oracle_home***/ocsb61/managed_server/server.properties** file includes these entries set to the correct file names:

   ```
   javax.net.ssl.keystore=servsrkeystore
   javax.net.ssl.trustStore=servertruststore
   ```

4. Copy the server certificate file (*Server_Key_Filename***.cer**) from the email server to the Service Broker keystore.

5. Import the server certificate file to the client truststore and server truststore with these commands:

   ```
   keytool -importcert -alias mail -file Server_Key_Filename.cer -keystore Server_
   Truststore -storepass Password -noprompt
   keytool -importcert -alias mail -file Server_Key_Filename.cer -keystore Client_
   Truststore -storepass Password -noprompt
   ```

6. In the JConsole, expand **oracle.axia.api.management.ds.DomainServiceMBean** then **setDomainProperty**.

7. Enter these name-value pairs:

   ```
   Name: axia.ssl; Value: true
   Name: systemprop.axia.netio.ssl; Value: true
   Name: systemprop.axia.netio.client_auth; Value: false
   ```

Your SVC Social Voicemail service now automatically sends voice-mail messages as WAV files in email messages to subscribers.

Configuring SVC Social Voice Group Calling Timers

The SVC Social Voice group calling features allow you to customize the response time that callees have when they answer a group call. When a callee first receives a group call they hear a recorded message instructing them to press **5** then **#** to accept the call. The default settings for pressing these digits are:

- Time allowed to press 5 after the recording is played: 20000 ms
- Time allowed to press # after pressing 5: 1000 ms. If the callee does not press **#**, they are automatically added to the call after this timer has expires.
- Time allowed for a user to press another digit after pressing 5: 20000 ms. This timer currently has no effect and is reserved for future use.

To change these time limits:

1. Use the Service Broker JConsole, or any JMX client to expand this MBean: **oracle.axia.apps.svc.groupsvc.framework groupsvcConfig**
   1. To enter a different time limit a callee is allowed after listening to the recorded message before pressing **5**, set the **firstDigit** attribute to a new value (milliseconds).
   2. To set a different time limit a callee is allowed after pressing **5** to press **#**, set **extraDigit** to a new value (milliseconds).

Configuring the Social Circle Click-to-Dial Service

This section lists the configuration tasks necessary to configure the SVC Social Circle Click-to-Dial service. The SVC Social Circle Click-to-Dial service is the only SVC service that requires these configuration tasks.

The Social Circle Click-to-Dial service is unique among the SVC services in that it is a session-initiating application. Traffic originates from it and travels through Service Broker to the telecom network. You route traffic from the click-to-dial application using IM-ASF-SAL and from this IM to the Orchestration Engine, and then to network nodes.

Complete these tasks to configure the Social Circle Click-to-Dial service:

1. Select and configure an IP Multimedia Subsystem (IMS) to store WAV files for the click-to-dial messages.
2. Select and configure a SIP server to accept click-to-dial calls.
3. Create and activate an application-facing IM-ASF SAL.

   For more information on creating IMs, see the discussion on managing Interworking Modules in *Oracle Communications Service Broker Modules Configuration Guide*.
4. Create and activate an R-IM-ASF to connect the SIP server to the Orchestration Engine. Set the **Network Entity Email Address** field to:

   ```
   sip:cscf@IP_address:Port_number
   ```

   Where:

   IP_Address - The IP address of your SIP server.

Port_number - The SIP server port to use. 5060 is the default.

For more information on creating IMs, see the discussion on managing Interworking Modules in *Oracle Communications Service Broker Modules Configuration Guide*.

5. Configure the SIP SSU.

   For more information on configuring the SIP SSU see the discussion on configuring SIP Signaling Server Units in *Oracle Communications Service Broker Signaling Server Units Configuration Guide.* Be sure to add your SIP server IP address to the **GloballyRoutable User agent URI:** box.

6. Configure the Orchestration Engine.

   Navigate to **OCSB**, **Processing Tier**, **Orchestration Engine**, **General**. In the **Subscriber Profile Receiver** field, select **OlpLSSInfoReceiver** to route traffic to the IM-ASF that you created.

   For more information on configuring the Orchestration Engine, see the discussion on configuring the Orchestration Engine in *Oracle Communications Service Broker Modules Configuration Guide.*

7. Configure the Service Broker orchestration Profiles:

   a. In the Administration Console navigate to **OCSB**, **Processing Tier**, **Supplementary Modules**, **SM-LSS**.

   b. Click the **Configuration**, **Orchestration Profiles**.

   c. Select the default orchestration profile and click **Update** to update.

      The **Update** dialog box appears. It contains the local initial filter criteria (iFC) logic that controls the application chain applied by Service Broker. The **OLP Data** field contains the actual iFC text that specifies the orchestration logic

   d. Fill in the iFC criteria to use. This example **clicktodial.ifc** file will help guide you through the process. Also see "Sample iFC Routing Rules" for an iFC example that includes the other SVC services.

```
<?xml version="1.0" encoding="UTF-8"?>
<Sh-Data>
<Sh-IMS-Data>
  <IFCs>
     <InitialFilterCriteria>
      <Priority>0</Priority>
      <TriggerPoint>
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <RequestURI>sip:User_Nmae@IP_Address:Port_Number</RequestURI>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasf_Name.IMASF@ocsb</ServerName>
       <DefaultHandling>0</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>
     <InitialFilterCriteria>
      <Priority>1</Priority>
      <TriggerPoint>
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <RequestURI>sip:User_Nmae@IP_Address:Port_Number</RequestURI>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasf_Name.IMASF@ocsb</ServerName>
       <DefaultHandling>0</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>
  </IFCs>
</Sh-IMS-Data>
</Sh-Data>
```

e. Configure these parameters using the Service Broker JConsole, or any JMX client or create scripts to run on the MBeans directly:

   – Set the application-facing module name in this MBean: **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.personalsvc.configuration,version=6.1.0,name0=personalsvcConfig,name1=click2dial,name2=applicationFacingModuleName**.

     Use this syntax: **sip:***imrasfsal_name***.IMRASFSAL@***domain_name***.com**.

   – (Optional) Replace the default **click_2_dial.wav** announcement files in these MBeans. These files are played to a subscriber when they initiate a click-to-dial call. Replace these default files with versions that meets your implementation's needs. **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.personalsvc.configuration,version=6.1.0.name0=personalsvcConfig,name1=click2dial,name2=click2DialVoiceIMS**

     And

     oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.personalsvc.configuration,version=6.1.0,name0=personalsvcConfig,name1=click2dial,name2=click2DialVoiceIN

   – Set the default network type in this MBean: **oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.personalsvc.configuration,version=6.1.0,name0=personalsvcConfig,name1=click2dial,name2=defaultNetworkType**.

     The default value is **IMS**. Change this to **IN** to for SS7 networks.

For more information on configuring the SM-LSS see the discussion on configuring SM-LSS in *Oracle Communications Service Broker Modules Configuration Guide*.

Sample iFC Routing Rules

This section shows a sample set of iFC routing rules used to test traffic for SVC services. Example data has been filled in.

```
<?xml version="1.0" encoding="UTF-8"?>
<Sh-Data>
<Sh-IMS-Data>
  <IFCs>

     <!-- Social Voicemail service -->
     <InitialFilterCriteria>
      <Priority>0</Priority>
      <TriggerPoint>
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <RequestURI>sip:29009@open-ims.test</RequestURI>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasfsal_vm.IMASFSAL@ocsb</ServerName>
       <DefaultHandling>1</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>

     <!-- Social Circle one number service-->
     <InitialFilterCriteria>
      <Priority>1</Priority>
      <TriggerPoint>
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <SIPHeader>
          <Header>To</Header>
          <Content>.*sip:456.*@open-ims.test.*</Content>
        </SIPHeader>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasfsal_gons.IMASFSAL@ocsb</ServerName>
       <DefaultHandling>1</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>

     <!-- Social Voice one number service-->
     <InitialFilterCriteria>
      <Priority>2</Priority>
      <TriggerPoint>
```

```
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <SIPHeader>
          <Header>To</Header>
          <Content>.*sip:123.*@open-ims.test.*</Content>
        </SIPHeader>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasfsal_1.IMASFSAL@ocsb</ServerName>
       <DefaultHandling>1</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>

     <InitialFilterCriteria>
      <Priority>3</Priority>
      <TriggerPoint>
       <ConditionTypeCNF>0</ConditionTypeCNF>
       <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
       </SPT>
       </TriggerPoint>
      <ApplicationServer>
       <ServerName>sip:imasf_cscf.IMASF@ocsb</ServerName>
       <DefaultHandling>1</DefaultHandling>
      </ApplicationServer>
     </InitialFilterCriteria>

  </IFCs>
</Sh-IMS-Data>
</Sh-Data>
```

About Creating a User Portal for Your Subscribers

The first step in providing SVC services to your subscribers is to create a web-based user portal that your subscribers use to access those services. SVC includes an API that you use for this purpose. You can design the user portal in any way that meets your requirements. For more information on the SVC API see "About the SVC API Operations" and "Social Voice Communicator API Reference".

3

Configuring Charging for Social Voice Communicator Services

This chapter explains how to integrate Oracle Communications Service Broker Social Voice Communicator (SVC) with external charging applications.

About Charging

SVC supports offline and online Diameter-based charging applications. Specifically, it acts as a Diameter client or Charging Trigger Function (CTF) that reports session and event information related to SVC activities to external Diameter servers.

SVC supports Diameter offline (Rf) charging for all of its services and online (Ro) charging for incoming calls to SVC subscribers. Charging can be event-based, session-based, or both.

Diameter messages are mainly composed of attribute-value pairs (AVPs) containing charging data. SVC supports AVPs from both the RFC 3855 and 3GPP TS 32.299 standards (standards-based AVPs) and additional AVPs supplied for SVC (non-standards based AVPs).

Configuring charging to work with SVC includes these general steps:

- Configuring communication between your charging application and Service Broker. You do this by configuring the SSU Diameter and an IM-OCF.
- Configuring traffic routing within Service Broker by configuring the Orchestration Engine.

Configuring Online Charging for SVC Services

SVC supports Diameter Ro (online) charging of incoming calls only. This section assumes that you have:

- Obtained a Diameter server and know its IP address and port number.
- Installed Service Broker.
- Started the domain and managed server.

SVC supports the Diameter Ro AVPs listed in the following sections:

- SVC Diameter Ro AVPs from Diameter Base and Credit-Control Protocols
- Standards-based SVC Diameter Ro AVPs
- Non-Standards Based Diameter Ro AVPs

To set up Diameter Ro charging for your SVC service:

1. Open the Administration Console.
2. In the navigation pane navigate to **OCSB**, **Signaling Tier**, then **SSU Diameter**.
3. Configure **SSU Diameter**. Specifically, define Service Broker as the Diameter node, and define at least one Diameter peer. You must specify the IP address, host name, listening port number, and supported protocol for each node and peer. Ensure that the **Application ID** entry is set to 4 for Ro charging.

   For more information, see the discussion on configuring Diameter Signalling Server Units in *Oracle Communications Service Broker Signaling Server Units Configuration Guide*.
4. Expand OCSB, **Processing Tier** and **Interworking Modules**.
5. Create and configure an IM_OCF Interworking Module.

   For more information on the configuration settings, see the discussion on configuring IM-OCF in *Oracle Communications Service Broker Modules Configuration Guide*.
6. Configure the Service Broker Orchestration Engine.

   a. In the navigation pane, navigate to **OCSB** then **Processing Tier**.

   b. Click **Orchestration Engine**.

   c. Configure the Orchestration Engine to route traffic between your Service Broker and Oracle Communications Billing and Revenue Management (BRM). You do this by creating an Initial Filter Criteria (IFC) XML file that specifies the routing criteria.

   For more information on Orchestration Engine configuration, see the discussion on configuring the Orchestration Engine and the discussion on initial filter criteria in *Oracle Communications Service Broker Modules Configuration Guide*.

Be sure your Diameter charging server and media servers are running before you start generating traffic.

Configuring Offline Charging for SVC Services

The Service Broker SVC feature uses connections in the Signalling Tier to exchange messages with external Diameter servers. Therefore, to enable offline charging, you must configure the Diameter server connection settings in the Signaling Tier. You also configure the network-facing IMs and orchestration rules in the Processing Tier that integrate the VPN application with the external Diameter server.

The Diameter AVPs that are generated by the SVC application are defined by MBean attributes. The supported AVPs are listed in:

- SVC Diameter Rf AVPs
- SVC Diameter Rf AVPs

This section assumes that you have:

- Configured a Diameter-based charging server (such as Oracle Communications BRM) and know its IP address and listening port number.

  For more information on setting up BRM see the Oracle Communications BRM documentation set.
- Installed Service Broker.

- Created and started a domain and managed server.
- Followed the instructions in "Setting Up Social Voice Communicator" and set up the processing and signalling tiers for SVC services.

To set up Diameter Rf charging for SVC services:

1. Configure **SSU Diameter**.

   Specifically, define Service Broker as the Diameter node, and define at least one Diameter peer. You need to specify the IP address, host name, listening port number, and supported protocol for each node and peer.

   For more information on the **SSU Diameter** configuration settings, see the discussion on configuring Diameter Signaling Server units in *Oracle Communications Service Broker Signaling Domain Configuration Guide.*

   Ensure that the **SSU Diameter**, **Diameter Nodes**, **Routes**, **Application ID** field is set to**3** for Rf charging.
2. Start and connect to the Administration Console process with a JMX MBean client.
3. Configure event-based charging by navigating to the following MBean:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.diametersvc,version=6.1.0,name0=diameterConfig,name1=enableChargingEventRf
4. Set the **enableChargingEvent** Rf attribute to **true**. The default setting is **false**.
5. Configure session-based charging by navigating to this MBean:

   oracle:type=oracle.axia.cm.ConfigurationMBean,name=oracle.axia.apps.svc.diametersvc,version=6.1.0,name0=diameterConfig,name1=enableChargingSessionRf
6. Set the **eventChargingSessionRf** attribute to **true**. The default setting is **false**.
7. In the Service Broker Signaling Domain, configure connectivity to the external Diameter charging server to which you want to send accounting reports.

   For instructions on how to perform this configuration, see the information about configuring Diameter Signaling Server Units in *Oracle Communications Service Broker Signaling Server Units Configuration Guide*.

The Service Broker SVC services now generates accounting request messages based on the service activities, which are transmitted to the charging server specified in the Service Broker Signalling Tier configuration.

SVC Diameter AVP Reference

The following sections list the AVPs generated by the SVC services.

SVC Diameter Ro AVPs from Diameter Base and Credit-Control Protocols

The AVPs defined by the Diameter base protocol (RFC3588) and Diameter Credit-Control (RFC 4006) that SVC supports are listed below.

The definitions of the attributes as generated by SVC conform to their definitions as stipulated by the public specification.

For complete information on the definition or purpose of a specific AVP, see the RFC 3588 Diameter Base Protocol.

- **Session-Id**

- **Origin-Host**
- **Origin-Realm**
- **Destination-Host**
- **Destination-Realm**
- **Auth-Application-Id**
- **Service-Context-Id**
- **CC-Request-Type**
- **CC-Request-Number**
- **User-Name**

Standards-based SVC Diameter Ro AVPs

The AVPs defined by 3GPP TS 32.299 Diameter Base Protocol that SVC supports are listed below.

The definitions of the attributes as generated by SVC conform to their definitions as stipulated by public specification.

For complete information on the definition or purpose of a specific AVP, see the 3GPP TS 32.299 specification.

- **Origin-State-Id**
- **Event-Timestamp**
- **Service-Information**
- **Service-Information.Subscription-Id**
- **Service-Information.Subscriber-Id.Subscription-Id-Type**
- **Service-Information.Subscriber-Id.Subscription-Id-Data**
- **Service-Information.Subscription-Id**
- **Termination-Cause**
- **Requested-Action**
- **Multiple-Services-Indicator**
- **Multiple-Services-Credit-Control**
- **Used-Service-Unit**
- **Reporting-Reason**
- **CC-Time**
- **Service-Identifier**

Non-Standards Based Diameter Ro AVPs

Table 3–1 lists the non-standards based Diameter Ro AVPs that Service Broker uses for passing SVC service information. See "Configuring Online Charging for SVC Services" for instructions on how to implement these AVPs.

Table 3–1 SVC Diameter Ro Optional AVPs

AVP Name	AVP Code	Type	Description
Service-Information.IMS-Information	**876**	Grouped	Transmits additional 3GPP service-specific information elements not covered by existing AVPs.
IMS-Information.Role-of-node	**829**	Enumerated	Specifies the role of the Application Server. Can be one of the following values: **0**: ORIGINATING_ROLE The Application Server serves the calling party. **1**: TERMINATING_ROLE The Application Server serves the callee.
IMS-Information.Event-Type	**823**	Grouped	Includes information about the type of chargeable telecommunication service or event for which the accounting-request and/or credit control request messages are generated.
Event-TypeSIP-Method	**824**	UTF8String	The SIP method name (for example, **INVITE** or **UPDATE**) that generated the accounting request. If the request is a one number Service request, one of the following values is also returned: **1**: Join - interim **REINVITE** message **2**: Leaving - Interim **BYE** message
IMS-Information.Node-Functionality	**862**	Enumerated	The node functionality identifier.
IMS-InformationUser-Session-ID	**830**	UTF8String	Contains the session identifier (the caller ID of the initial request).
I**MS-Information.Time-Stamps**	**833**	Grouped	Contains the times of the initial SIP request and the response.
Time-Stamps. SIP-Request-Timestamp	**834**	Timestamp	The SIP request timestamp in UTC format.
Time-Stamps. SIP-Response-Timestamp	**835**	Timestamp	The SIP request response timestamp in UTC format.
I**MS-Information Service-ID**	**855**	UTF8String	Identifies the service that the MRFC is hosting. For conferences, the conference ID is used.
IMS-Charging-Identifier	**841**	UTF8String	The IMS charging identifier (ICID) generated by the IMS node for the SIP session.
IMS-Information Calling-Party-Address	**831**	UTF8String	The address (SIP URI or TEL URI) of the calling party (Public User Identifier or Public Service Identity).
IMS-Information Called-Party-Address	**832**	UTF8String	The address of the callee. This is either the SIP URI or TEL URI from the initial request.
IMS-InformationRequested-Party-Address	**1251**	UTF8String	Contains the address (SIP URI or TEL URI) of the original target of the SIP request (Public User Identifier or Public Service Identity). Only present if different from the Called-Party-Address parameter (832).
IMS-InformationService-Specific-Info	**1249**	Grouped	Contains the service-specific data if provided by the Application Server or a PCEF (for pre-defined PCC rules).

Table 3–1 (Cont.) SVC Diameter Ro Optional AVPs

AVP Name	AVP Code	Type	Description
Service-Specific-InfoService-Specific-Data	**863**	UTF8String	Can be one of the following values: **PLACE_CALL** **TAKING_CALL** **CLIDK_TO_DIAL** **VOICEMAIL** **GROUP_SERVICE** **ONE-THE-FLY_CONFERENCING**
Service-Specific-InfoService-Specific-Type	**1257**	Unsigned32	Contains the type of service-specific data: **0**: SVC
IMS-InformationSDP-Media-Component	**843**	Grouped	Contains the media (SDP) used for a session. This is taken from the **START** message of the initial ACR. If the media changes, it is taken from an interim ACR.
SDP-MediaComponentSDP-Media-Name	**844**	UTF8String	Contains the value for **m=** line in the SDP data of the **200 OK** message.
SDP-Media-ComponentSDP-Media-Description	**845**	UTF8String	Contains the value for the **attribute-line** for a media component. Specifies the media described in the **SDP-Media-Name** AVP.
IMS-InformationCause-Code	**861**	Integer32	Contains the code (from 3PP TS 32.299) for the **Accounting-request [stop]** and/or **Account-request [event]**. Values less than **0** are used for success codes; values greater than **1** are used for failure codes.

SVC Diameter Rf AVPs

The AVPs defined by RFC 3588 Diameter Base Protocol that SVC uses are listed below.

The definitions of the attributes as generated by SVC conform to their definitions as stipulated by the public specification.

For complete information on the definition or purpose of a specific AVP, see the RFC 3588 Diameter Base Protocol.

- **Session-Id**
- **Origin-Host**
- **Origin-Realm**
- **Destination-Host**
- **Destination-Realm**
- **Accounting-Record-Type**
- **Accounting-Record-Number**
- **Acct-Application-Id**
- **Acct-Interim-Interval**
- **User-Name**
- **Event-Timestamp**

SVC Diameter Rf AVPs

The AVPs defined by 3GPP TS 32.299 Diameter Base Protocol that SVC uses are listed below.

The definitions of the attributes as generated by SVC conform to their definitions as stipulated by public specification.

For complete information on the definition or purpose of a specific AVP, see the 3GPP TS 32.299 specification.

- **Service-Context-Id**
- **Service-Information**
- **Service-Information.Subscription-Id**
- **Service-Information.Subscriber-Id.Subscription-Id-Type**
- **Service-Information.Subscriber-Id.Subscription-Id-Data**
- **Service-Information.IMS-Information**
- **IMS-Information.Role-of-node**
- **IMS-Information.Event-Type**
- **Event-Type.SIP-Method**
- **IMS-Information.Node-Functionality**
- **IMS-Information.User-Session-ID**
- **IMS-Information.Time-Stamps**
- **Time-Stamps.SIP-Request-Timestamp**
- **Time-Stamps.SIP-Response-Timestamp**
- **IMS-Information.Service-ID**
- **IMS-Charging-Identifier**
- **IMS-Information.Calling-Party-Address**
- **IMS-Information.Called-Party-Address**
- **IMS-Information.Requested-Party-Address**
- **IMS-Information.Service-Specific-Info**
- **Service-Specific-Info.Service-Specific-Data**
- **Service-Specific-Info.Service-Specific-Type**
- **IMS-Information.SDP-Media-Component**
- **SDP-Media-Component.SDP-Media-Name**
- **SDP-Media-Component.SDP-Media-Description**
- **IMS-Information.Cause-Code**

4

Provisioning and Managing Social Voice Communicator Services

This chapter explains how to provision and manage Oracle Communications Service Broker Social Voice Communicator (SVC) subscriber services data.

About the SVC API

You use the SVC RESTful API to create, provision, configure, and manage the SVC services that you offer to your subscribers.

You typically create a self-administered web portal (or add to an existing portal) that you subscribers use to self-administer their accounts. You can design the user portal any way that meets your requirements, using any tools at your disposal, if they can use the RESTful API.

About Creating Subscriber Accounts with the SVC API

Before subscribes can access their accounts and access SVC services, an administrator must create their accounts. The SVC API operations, including the "Creating Accounts and Services by Using the SVC API" operation, operate on one record at a time.

Your administrator may need to create or alter multiple accounts at the same time, for example to:

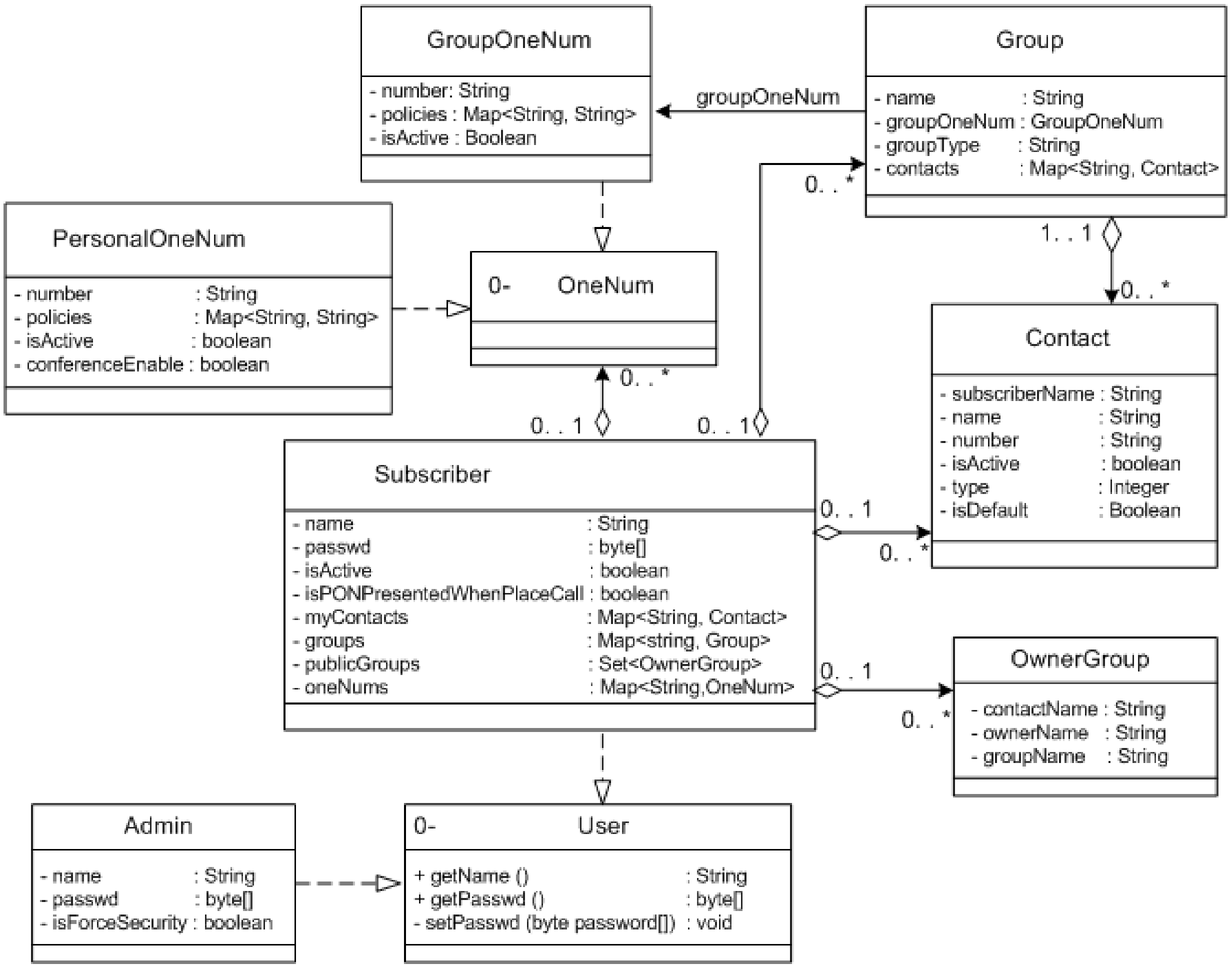
- Create accounts to have ready for your subscribers to configure.
- Make large-scale changes to your SVC subscriber base for changes in policy.
- Load a test database full of accounts to test new functionality.

Administrators use the Service Broker Batch Loader program to create or alter groups of SVC subscriber accounts. See "Provisioning Multiple Accounts by Using the Batch Loader Tools" for more information.

Understanding the SVC Data Model

Figure 4–1 shows the SVC data objects and their relationships.

Figure 4–1 SVC Data Model



Creating Accounts and Services by Using the SVC API

SVC is designed primarily to allow subscribers to administer their own accounts individually once you set up a user portal with the API operations. Your administration staff can also use these operations on individual subscriber accounts. The following sections explains the available API operations.

Managing Subscriber Accounts for SVC Services

Perform operations in Table 4–1 to create, change, or delete SVC subscriber accounts for all SVC services.

Table 4–1 Subscriber Account Management Operations

Operation	Description
Create Subscriber Accounts	Creates subscriber accounts. Administrator-only.
Modify a Subscriber Password or Social Circle One Number Preference	Changes subscriber account details.
Delete a Subscriber Account	Removes a subscriber account from the database.

Table 4–1 (Cont.) Subscriber Account Management Operations

Operation	Description
Get Subscriber Accounts	Returns a list of all subscriber account names and locations. Administrator-only.
Get Subscriber Account Details	Returns all details for a subscriber account.

Managing Subscriber Account Personal Contacts

Perform the operations in Table 4–2 to add, change, delete, or obtain information for personal contacts.

Table 4–2 Managing Personal Contact Operations

Operation	Description
Create Personal Contacts in a Subscriber Account	Populates a subscriber's account with the personal contacts contained in the array submitted with this operation.
Modify a Personal Contact Details and/or Default Status	Modifies a personal contact with the new status (toggles default on or off) and number that you submit with this operation.
Get a Subscriber's Personal Contact List	Returns an array of personal contact names and access URIs for a subscriber.
Get a Personal Contact and its Default Status	Returns the personal contact number and status (**default** or not) for a single personal contact.
Delete a Social Circle Contact	Deletes the personal contact that you submit with this operation.
Delete All Personal Contacts	Removes all personal contacts for a subscriber.

Managing Subscriber Personal Social Circle Accounts

Perform the operations in Table 4–3 to manage the SVC Social Circle one numbers and the routing policy rules for those numbers.

Table 4–3 Social Voice Service Management API Operations

Operation	Description
Create a Social Circle One Number	Adds an SVC Social Voice service to a subscriber account. Administrator-only.
Get All Social Circle One Number Contacts for a Subscriber	Retrieves the list of SVC Social Voice contacts for a subscriber's SVC Social Circle.
Get Details for a Social Circle Contact	Retrieves the SVC Social Voice name, type (**group** or **personal**), and any policies applied to a SVC Social Voice contact.
Delete a Social Circle Contact	Removes a SVC Social Voice service from a subscriber's account.
Delete All Social Circle Contacts	Removes all SVC Social Voice contacts for a subscriber's account.
Create a Policy Rule for a Social Circle Contact	Applies call routing policies to a single SVC Social Voice contact, that are based on the time of day and day of week.
Get a Social Circle Policy Rule Details	Retrieves details of a policy object.

Table 4–3 (Cont.) Social Voice Service Management API Operations

Operation	Description
Modify a Social Circle Policy Rule	Makes changes to a subscriber's SVC Social Voice policy, by replacing the policy object in the database.
Delete a Social Circle Policy Rule	Removes a SVC Social Circle policy rule.

Managing Public and Private Groups

Each SVC Social Voice account can have any number of group and group members. There are two types of groups: public, which any subscriber can join, and private, which can be more exclusive. Typical private groups include family and friends; typical public groups include sports teams or hobby clubs. Two default groups are supplied: **personal** and **business**. Use the operations Table 4–4 to administer SVC Social Voice private groups, and the operations in Table 4–5 to administer SVC Social Voice public groups.

Table 4–4 Social Voice Private Group Management Operations

Operation	Description
Create a Private Group	Adds a SVC Social Voice private group to a subscriber's account.
Add a Private Group Contact	Adds a new member to a private group.
Get All Private Group Memberships for a Subscriber	Returns details for all private groups that a subscriber owns or is a member of.
Get Private Group Details	Sets the name, SVC Social Voice group number, and URI of a subscriber's contact group.
Modify a Private Group's One Number	Changes a private group's one number value.
Get All Contacts for a Private Group	Returns an array of private group member.
Get Details for a Private Group Contact	Returns the name, number, and active status for a private group member.
Modify a Private Group Contact	Changes a private group contact details.
Delete a Private Group Contact	Removes a subscriber from a private group.
Delete a Private Group	Deletes a contact group and all its members from a subscriber's account.

Table 4–5 Social Voice Public Group Management

Operation	Description
Create a Public Group	Creates an SVC Social Voice public group.
Add a Public Group Member	Adds a member to a Social Voice public group.
Delete a Public Group Contact	Removes a member from a public group.

Table 4–5 (Cont.) Social Voice Public Group Management

Operation	Description
Get All Public Groups for a Subscriber	Returns an array of the types, URIs, and names of all groups that a subscriber owns.
Get All Public Group Memberships for a Subscriber	Returns an array of all group names and group owners that a subscriber is a member of.
Get All Contacts for a Public Group	Returns an array of the active status, name, contact name, contact number, and group type for each contact in a public group.
Delete a Public Group	Removes an SVC Social Voice public group.

Making a Click-to-Dial Call

Perform the "Make a Click-to-dial Call" operation to initiate a call from a web interface.

Note that ACR messages are sent after the call is connected, rather than before.

Managing Social Voicemail Access

Each SVC subscriber is eligible for access to a voice-mail system. Perform the operations in Table 4–6 to manage voice-mail access.

See "Configuring Voice-Mail for SVC Services" for details on configuring SVC voice-mail services.

Table 4–6 Voice-Mail Subscriber Management Operations

Operation	Description
Activate Social Voicemail for Subscribers	Grants subscribers access to SVC Social Voicemail features.
Activate/Deactivate Social Voicemail Access	Adds or removes SVC Social Voicemail access for a subscriber, but does not affect the subscriber's data. This operation is intended for temporary service suspension. Administrator-only. To permanently remove a voice-mail access perform the "Delete Social Voicemail Access and Data for a Social Circle" operation.
Delete Social Voicemail Access and Data for a Social Circle	Deactivates a subscriber's SVC Social Voicemail access and removes all data. This operation permanently removes voice-mail access and data. Administrator-only. To temporarily suspend voice-mail access, perform the "Activate/Deactivate Social Voicemail Access" operation.
Get All Social Voicemail Subscriber Accounts	Returns a list of all SVC Social Voicemail accounts and their status (active/inactive). Administrator-only.

Managing Social Voicemail Service

After granting access to SVC Social Voicemail services, perform the operations in Table 4–7 to manage subscriber's SVC Social Voicemail services.

Table 4–7 Voice-Mail Box Management Operations

Operation	Description
Get Social Voicemail Settings	Changes the SVC Social Voicemail settings.

Table 4–7 (Cont.) Voice-Mail Box Management Operations

Operation	Description
Delete a Social Voicemail Personal Greeting Message	Deletes the personal greeting WAV file. The default greeting file is used instead.
Modify Social Voicemail Message Public/Private Setting	Sets a voice-mail message as either public or private.
Deactivate a Social Voicemail Name Message	Deactivates a name message WAV file.
Get Social Voicemail Message Settings	Returns the voice-mailbox settings, such as password, maximum number of allowed messages, and so on.
Get Social Voicemail Name Message Settings	Returns the location and status of a a subscriber's voice name message WAV file.
Create Social Voicemail Greeting Message	Sets a WAV file as the **personalgreeting.wav** file in a subscriber's greeting folder.
Create Social Voicemail Box Name Message	Sets a WAV file as a voice-mailbox name message.

Managing Social Voicemail Messages

Perform the operations in Table 4–8 to manage SVC Social Voicemail messages.

Table 4–8 Voice-Mail Message Management Operations

Operation	Description
Delete a Social Voicemail Message	Removes a voice-mail message.
Get a List of Social Voicemail Messages	Retrieves a list of voice-mail messages.
Get a Social Voicemail Message File	Retrieves the WAV file containing a voice-mail message.
Get Details for a Social Voicemail Message	Retrieves the message type, date and time created, caller name and number, file location, and public/private setting for a voice-mail message.
Get Personal Greeting Message Settings	Retrieves the location and status of a voice-mailbox greeting message.
Send a Social Voicemail Message to Other Subscribers	Sends a voice-mail message from one subscriber to another.
Create sentMessage Voice Message	Uploads a WAV file.
Get a List of the Public Social Voicemail Messages in a Mailbox	Retrieves a list of the publicly-accessible voice-mail message in a subscriber's mailbox.

Provisioning Multiple Accounts by Using the Batch Loader Tools

The SVC API creates or alters records in one account at a time, often by the subscribers themselves. During the course of creating or administering your SVC implementation, you will probably need to create or alter multiple accounts at the same time.

You use the Service Broker batch loader tools to make bulk changes to multiple SVC accounts in the same database at the same time. For more information, see the discussion on using the batch operation tools in *Oracle Communications Service Broker VPN Implementation Guide*.

The batch loader tools require system administration-level permission to use, and require you to specify the user name and password of an API administrator. The program is located in *Oracle_home*/**ocsb61/admin_server/utils/bulkloader**.

The batch loader tools accept input in the form of comma-separated value files.

See "Batch Loader SVC Data Files" for more information on the input data files.

5

Monitoring Social Voice Communicator Services

This chapter describes the MBean operations you use to collect and view statistics generated by the Oracle Communications Service Broker Social Voice Communicator (SVC) services.

Monitoring SVC Services

This section describes the MBean operations that collect statistics for SVC services. For more information on collecting Service Broker statistics, see the discussion on monitoring Service Broker in *Oracle Communications Service Broker System Administrator's Guide*.

Table 5–1 describes attributes that enable you to monitor SVC. These operations are provided by this MBean: **oracle.axia.svc.common.statistic.management.SVCRuntimeStatisticsMBean**

Table 5–1 SVC Monitoring Attributes

Operation	Description	Type	Attribute
retrieveTotalCallCounts	Returns the total number of call attempts (including successful and failed calls).	Counter	Call
retrieveSuccessfulCallcounts	Returns the total number of successful calls for all SVC services.	Counter	SuccessfulCall
retrieveTotalPersonalCallCounts	Returns the total Number of SVC Social Circle calls.	Counter	PersonalCall
retrieveSuccessfulPersonalCallCounts	Returns the total number of successful SVC Social Circle calls.	Counter	SuccessfulPersonalCall
retrieveTotalGroupCallCountsCallCounts	Returns the total number of SVC Social Voice group calls.	Counter	GroupCall
retrieveSuccessfulGroupCallcCountsCallCounts	Returns the total number of successful SVC Social Voice group calls.	Counter	SuccessfulGroupCall
retrieveTotalConferencingCallCounts	Returns the total number number of conference calls.	Counter	ConferencingCall
retrieveCallCounts	Returns the total number of successful on-the-fly conference calls.	Counter	SuccessfulConferencingCall

Table 5–2 describes operations that you use to monitor the SVC Social Voicemail service. These operations are provided by this MBean:

oracle.axia.svc.common.statistic.management.SVCVmRuntimeStatisticsMBean.

Table 5–2 SVC Social Voicemail Monitoring Attributes

Operation	Description	Type	Attribute
retrieveTotalAccessingVMCounts	Returns the total number of voice-mail call attempts (includes successful and failed calls).	Counter	AccessingVoiceMail
retrieveCallCountsSuccessfulAccessingVMCounts	Returns the total number of successful voice-mail calls.	Counter	SuccessfulAccessingVoiceMail
retrieveTotalLeavingVMCounts	Returns the total number of calls attempting to leave voice-mail messages.	Counter	LeavingVoiceMail
retrieveSuccessfulLeavingVMCountsCall	Returns the total number of calls in which voice-mail messages were left.	Counter	SuccessfulLeavingVoice Mail

Table 5–3 describes attributes that enable you to monitor the CPS audit logs that SVC creates. These attributes are provided by this MBean:

oracle.axia.svc.common.statistic.management.SVCCPSAudioLogMBean

Table 5–3 SVC CPS Monitoring Attributes

Operation	Description	Type	Attribute
genCPSAuditLog	Returns a list of the CPS audit log names and each log's start and end dates.	Counter	CPSauditlogreport.

6

SVC Social Voicemail Call Flows

This chapter describes the Oracle Communications Service Broker Social Voice Communicator (SVC) Social Voicemail call flows. These flows cannot be changed.

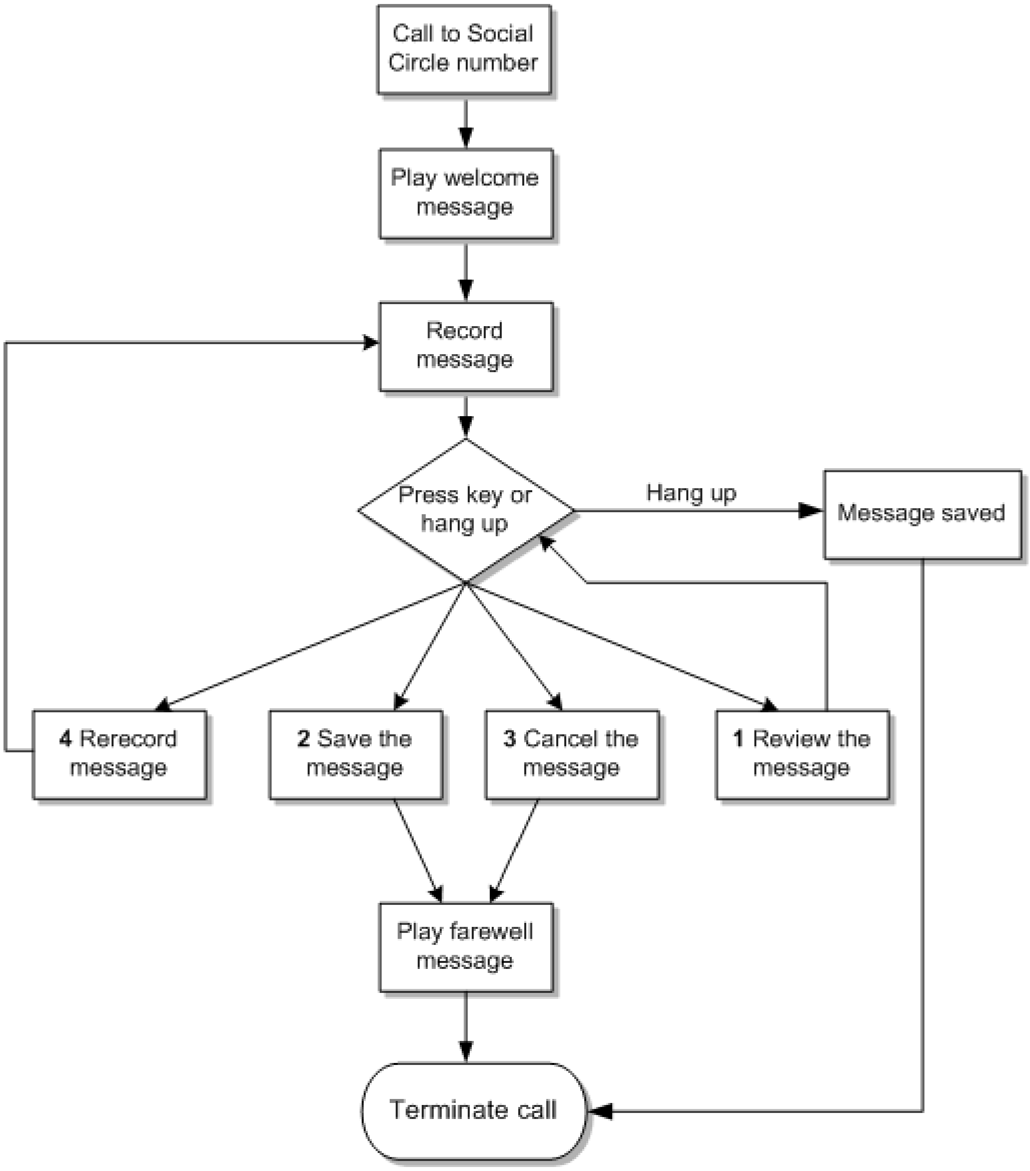
Social VoiceMail Call Flows

The following sections explain the order of messages and options that callers entering the SVC Voicemail system hear.

Social Circle Call Flow for Leaving a Message

Figure 6–1 shows the call flow when a person calls a SVC Social Circle one number and leaves a message.

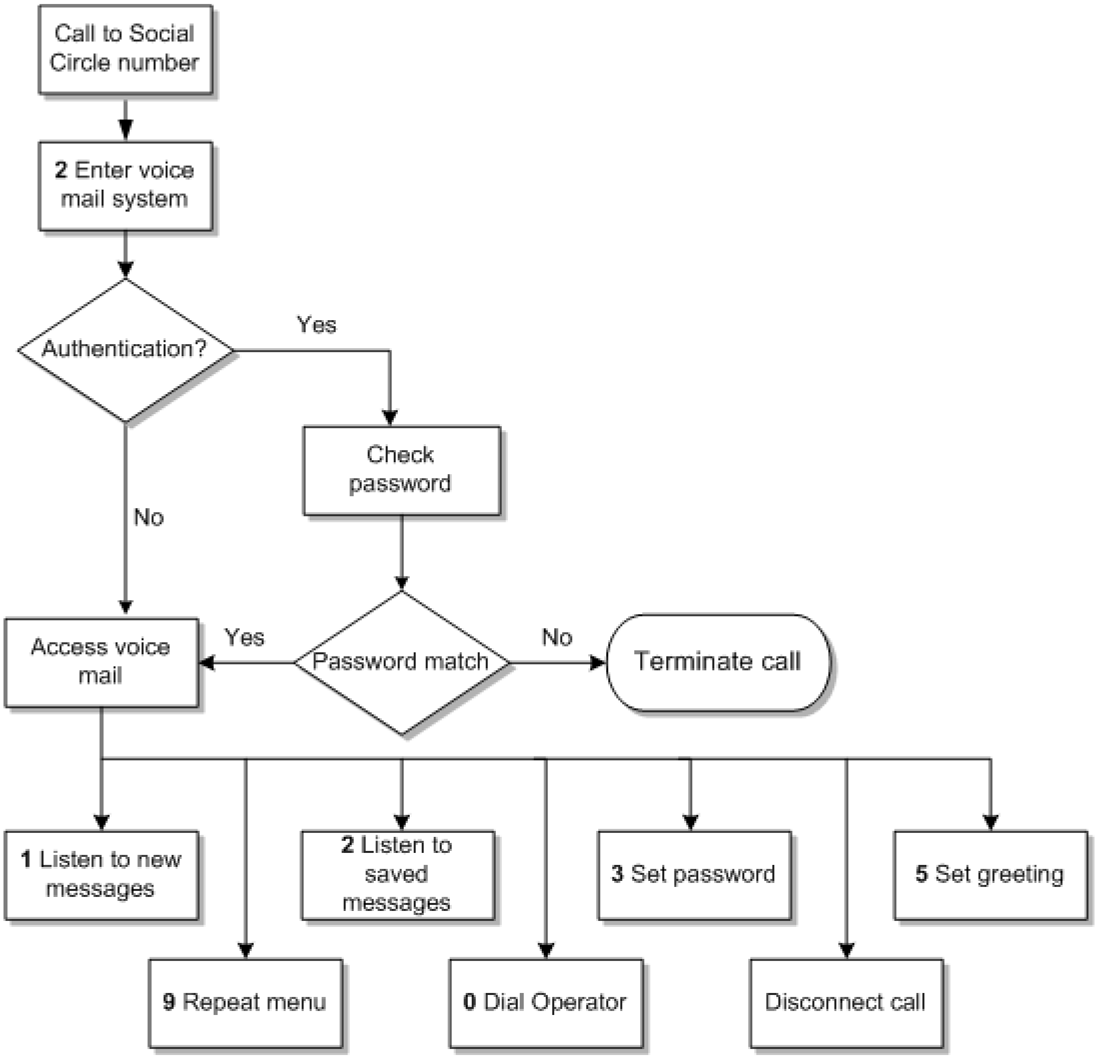
Figure 6–1 Call Flow for Leaving a Message



Call Flow for Accessing Social Voicemail

Figure 6–2 shows the call flow that a SVC Social Circle subscriber uses to access their SVC Social Voicemail service and select an option for service.

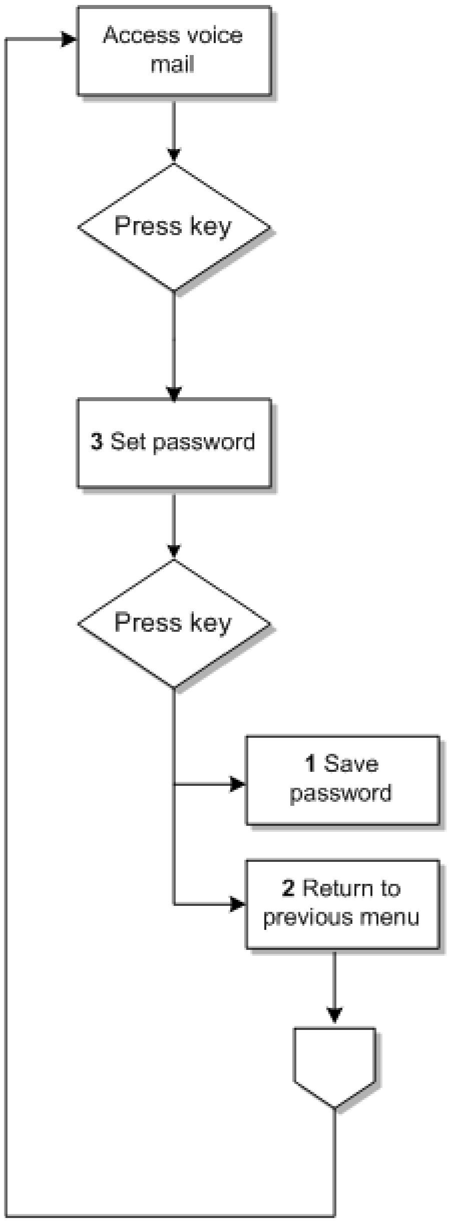
Figure 6–2 Access Voice-Mail Call Flow



Call Flow for Setting a Password

Figure 6–3 shows the call flow for setting SVC Social Voicemail password. This call flow starts after a user has been authenticated and has access to their voice-mail account.

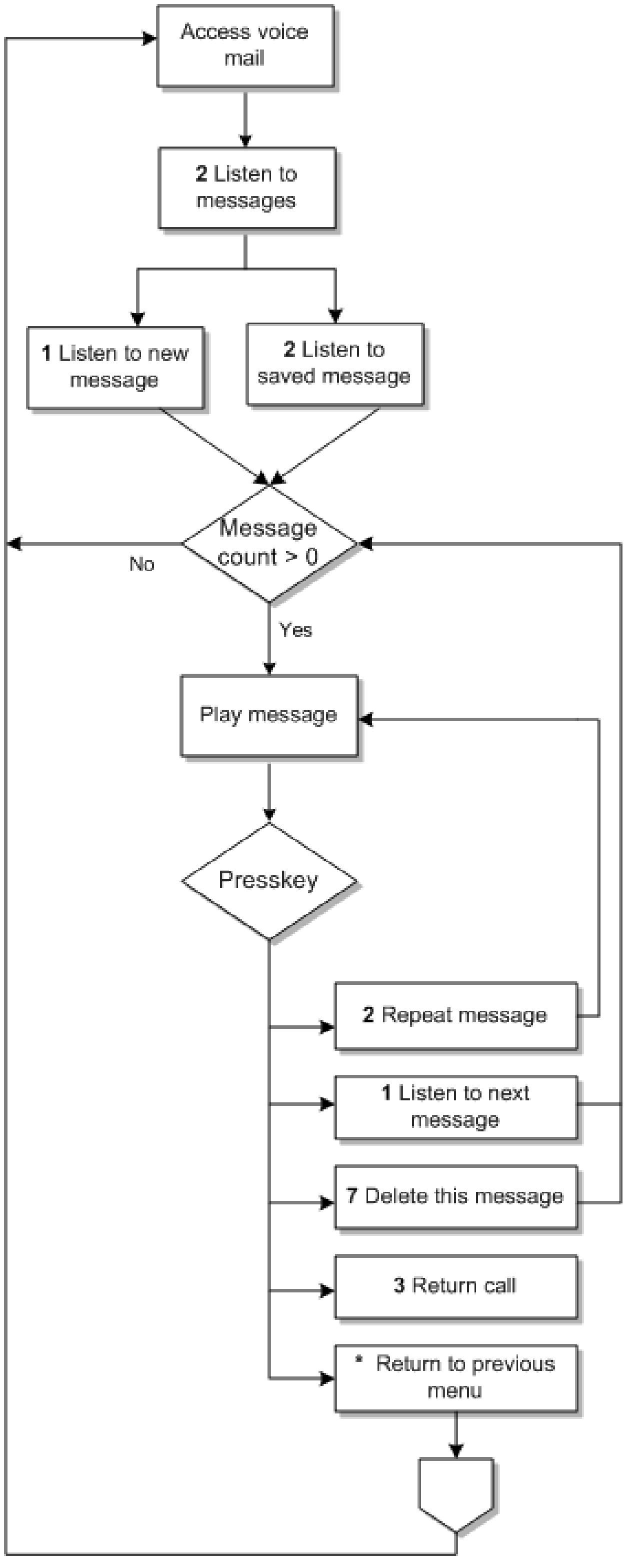
Figure 6–3 Setting Password Call Flow



Call Flow for Retrieving a Message

Figure 6–4 shows the call flow for retrieving a SVC Social Voicemail message. This call flow starts after a user has been authenticated and has access to their voice-mail account.

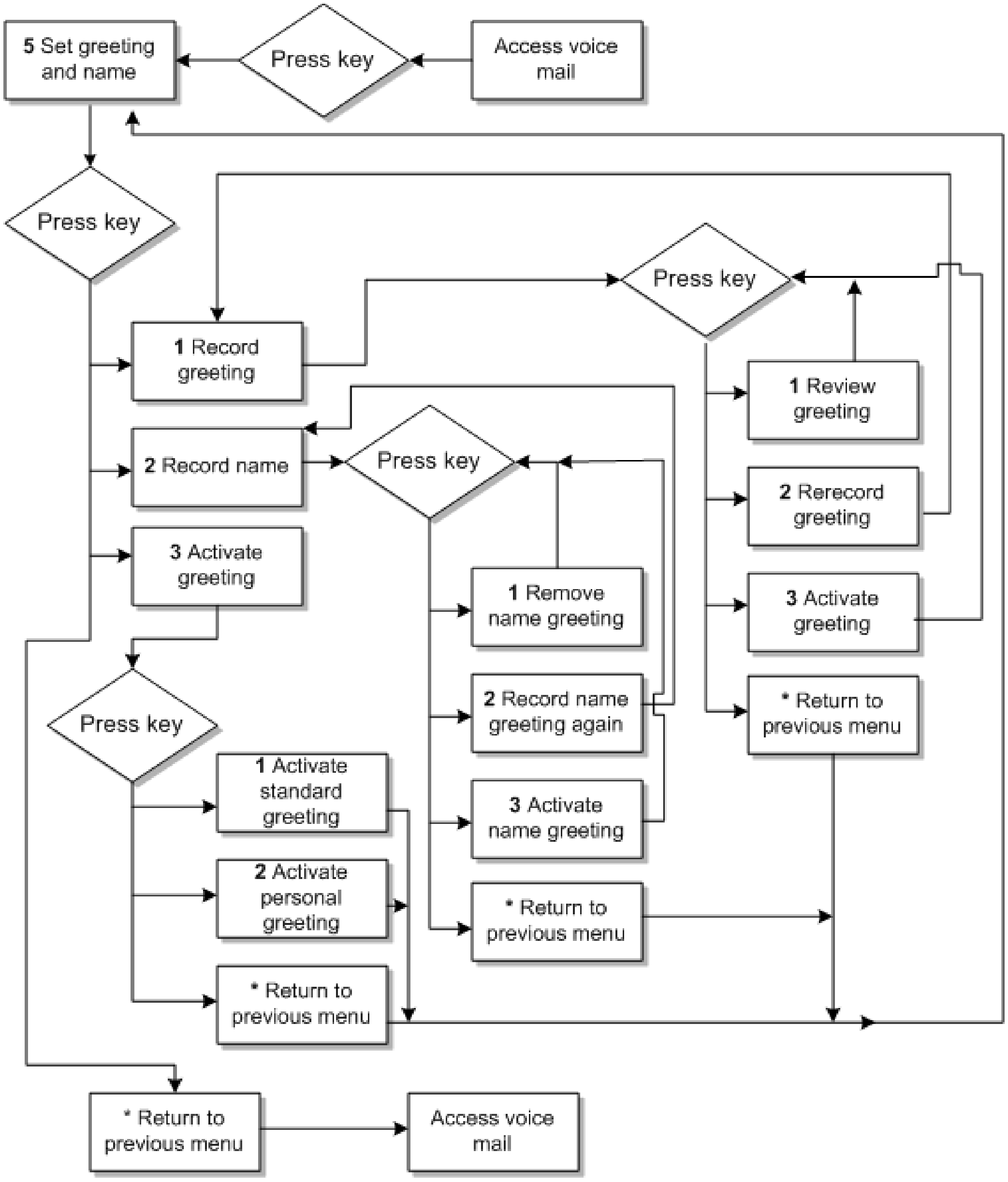
Figure 6–4 Retrieve Message Call Flow



Call Flow for Setting Up a Greeting

Figure 6–5 shows the remaining Voicemail Call flow, which mainly includes setting the name and greeting messages. This call flow starts after a user has been authenticated and has access to their voice-mail account.

Figure 6–5 Call Flow for Setting Up a Greeting



7

About the SVC API Operations

This chapter describes information that you need to know to use the Oracle Communications Service Broker Social Voice Communicator (SVC) API.

About the SVC API

You use the SVC API to create, change, and delete subscriber accounts and data for SVC services. Like any RESTful API, the SVC API makes its services available to client applications through simple HTTP requests.

All API operations are accessible by your administrators, and a large majority are accessible by your subscribers to use for managing their accounts. Your subscribers typically access these API operations through a self-managed web portal that you set up for them. This may be a dedicated portal that you create for your subscribers, or may more typically involve integrating these operations into an existing customer-facing web application.

Your system administration staff will probably also use these portals to perform day-to-day provisioning and configuring tasks for SVC services. These API operations are also available to other services or programs that can access the web server.

The batch operation tools are also clients of this API. You use the batch operation tools to populate application data for SVC services in bulk.

About Exposing the API Operations to Your Subscribers

The specific API-based tasks that you allow your subscribers to access depends on your business requirements. For example you can expose operations that allow your subscribers to:

- Change their account details.
- Add or modify SVC Social Circle contacts and contact details.
- Add or modify SVC Social Voice contacts and contact details.
- Administer their own SVC Social Voicemail messages, including retrieving and deleting voice-mail messages, adding or changing their voice-mail greeting, or deactivating their voice-mail access.

About REST

The SVC API follows the style of a (REpresentational State Transfer) interface.

In a RESTful API, functions are distinguished by the combination of a particular URI and the HTTP method used to access it. In general, the URI identifies the resource on which to act, and the HTTP method identifies the type of action to perform.

The resource can be a generic resource or an instance-specific resource. A generic resource is equivalent to a factory creation function in a traditional API. The client application would use the resource, for example, to create a new SVC subscriber SVC Social Voice contact. After the resource is created, it can be modified or inspected at its unique URI.

For example, to create objects that represent subscribers, the client application would send the following request to the API:

`POST /svc/subscribers/`*subscriber_name*

After the subscriber objects are created, the client application can inspect, modify, or delete the objects by sending a GET, PUT, or DELETE request to the address associated with that object instance. For a user with the user name of alice, for example, the URI to access the user's object would be:

```
/svc/subscribers/alice
```

The methods in the HTTP protocol used in the SVC RESTful API - POST, GET, PUT, and DELETE - correspond to the programming operations commonly known as CRUD operations. CRUD, which stands for create, read, update, and delete, represent the common operations applicable in data-oriented APIs. The equivalent function calls in a traditional API may be similar to `createUser()`, `getUser()`, `setUser()`, and `deleteUser()`. In this case, the instance on which the function operates is typically identified through an input parameter.

Accessing the SVC API

In a Service Broker deployment, the SVC API is exposed by the managed server. Therefore, at least one managed server must be deployed and running in your environment.

Several types of clients connect to the managed server instance, including the Administration Console client process and JMX clients. The managed server listens for SVC API at the port configured for client HTTP access. Before using this API, ensure that the managed server is configured to accept web requests.

In a managed server cluster, all servers access the same data store. Therefore, it is not important which managed server the API client addresses.

Request URI Format

Clients access the services of the SVC RESTful API by sending HTTP requests to the managed server instance in the Service Broker managed domain. To use the API, the managed server must be configured and running.

The format of the URL to invoke operations in the API is:

https://*host***:***port***/svc/***component***/***resourceId*

The URL is made up of these parts:

- *host*:*port*: are the hostname and port of the managed server in the SVC domain. In the **Domain Management**, **Servers** page of the Administrator Console, these values correspond to the Host and Port values for a specific managed server instance. Note that the Port field value should be used as the port number, not the Admin Port or JMX-related port.

- *component*: is the data system component that you want to modify, create or query.
- *resourceId*: The identifier of the server instance on which the application operates.

JSON Body Parameters

Operations that are performed by using the GET or DELETE HTTP methods do not require input values other than what is provided in the URL and headers of the client request. That is, they do not require HTTP body content to be supplied in the invocation request.

However, SVC RESTful API operations are performed by using the POST and PUT methods require additional input data. The API takes input parameters in the form of JSON (JavaScript Object Notation) data in the body of the request.

JSON is a data exchange format based on JavaScript that is commonly used to pass information between web clients and servers over HTTP. In the body of the request, JSON data appears as one or more name-value pairs. This example shows two name-value pairs user name for an object that represents a person. The data type of both parameters is string:

```
{adminUsername:"alice",password:"password"}
```

JSON defines relatively few data types, including:

- number
- string
- boolean
- object

Certain parameters are composed of a value array. For example, you create a list of subscriber personal contacts using an array of name-value pairs:

```
[ { "mobile":"13811110000","default":true},
{"home":"01082311212"},
{"office":"alice@oracle.com"}]
```

to simple types, a parameter can be a complex type. An object type is a parameter that contains a nested set of subparameters. For example a policy rule contain all parameters required for the rule as shown in the following example:

```
[{
 "froms": [],
 "sequencialCall": true,
 "sequentialRingInterval": 30,
 "toAddresses": [
  "tom@oracle.com",
  "0118613611046326"
 ],
 "schedule": {
   "startDate":{
     "year":2009,
     "month":10,
     "day":24
   },
   "startTime":{
     "hour":1,
     "minute":00,
     "second":0
```

```
      },
      "endTime":{
        "hour":13,
        "minute":0,
        "second":0
      },
      "recurrentPeriod":1,
      "timezone":+8,
      "until":{
        "year":2010,
        "month":12,
        "day":24
      }
    },
    "priority":10
  }]
```

In addition to validating the basic data type of a submitted parameter, SVC may validate the content of the parameter. For example, it may ensure that a number falls within an expected range or that dates meet the required format.

The values of certain parameters are limited to what are called *word characters*. Expressed in regular expression notation, the word character set is: [A-Za-z0-9_]. In other words, a value may contain upper or lower-case letters, numbers, and underscore characters.

Numeric values are expressed as digits. In some cases, a parameter may contain a range of numbers, such as the extension value for a PBX user. To specify a range of numbers, use the common numeric prefix for the number range followed by a wildcard, for example **898***. In this example, the range applies to all numbers that begin with the numbers 898, such as 89823 or 8981.

Data format requirements for each parameter, if applicable, are listed in "Provisioning and Managing Social Voice Communicator Services".

Creating and Updating Objects

To add subscribers and their contacts to SVC services, you use the SVC API to create the data objects that represent the entities you want to add.

The body of the POST request should contain the JSON-formatted data that describes the entity, such as the subscriber’s name or contact numbers.

Each type of object can have mandatory parameters and optional parameters. A mandatary parameter must be included in the create object request with a value that meets the content requirements for that parameter.

When updating an object, you supply a new value for a modified setting in the body of a PUT request. In the body, you only need to supply the values for those parameters you want to modify, not for the entire object.

Too Many Subscribers Message

If a GET operation is required to return a set of more than 40,000 results, the operation fails and this message is returned:

```
Too many subscribers, please contact the admin to get subscribers data
```

In this case the return set is too large for a message to contain and you should have a database administrator retrieve this data from the database itself.

Errors and Exceptions

When an operation succeeds, the SVC API returns a response with HTTP status code 200.

There are a variety of reasons that an operation can fail. For instance, an object you are creating may reference an object that does not yet exist; the JSON body of a request may be malformed; or field data may not match the expected from.

In general however, if an error occurs, SVC returns a response that has an HTTP code reflecting the nature of the error. These error codes include:

- 400: The request body value is not correct. The response body contains detailed information about the error.
- 401: username or password is not correct.
- 403: the administrator does not have permission to access this object.
- 404: the URL style is unsupported or the resource does not exist in the system.
- 405: method is not allowed.
- 406: request method is not supported for this URL.
- 500: internal system error occurred.
- 503: service is unavailable.

SVC Data Types

The SVC API uses the following data types:

- The standard JSON data types listed in RFC 4627.
- Array of digit strings: A comma-separated list usually used to pass a list of MSISDN numbers.
- String: A Java J2SE standard string.
- Digit string: A string that can include of zero and positive integers 1-9.
- String (address): An RFC 2396 standard address.
- String (email): An RFC 822 standard email address.
- Null value: An empty pair of double quotes: “”
- UNIX timestamp: Milliseconds after January 1, 1970, 0:00:00 GMT.

API operation that transfer WAV files do so as attachments to HTTP requests.

Accessing the SVC API

The API is exposed by a Service Broker managed server. You cannot use the API to provision and manage SVC service before configuring at least one managed server in the Service Broker installation, and the managed server must be running.

The managed server listens for SVC API requests at the host address and port configured for the managed server. Specifically, the API is served on the port of the Port attribute, the Admin Port, or JMX-related ports for the server.

By default, security for the API is off. Turn on security to have basic authentication performed for the server.

Authentication and Authorization

By default, security is disabled for the SVC API. That is, clients can submit requests to the API without providing user credentials. While this may be suitable for test and evaluation systems, for production-level systems, access to the SVC RESTful API should be secured.

There are two types of SVC users:

- **Administrators** have full access to the functions of the API. These users can modify data objects associated with any SVC subscriber. Administrators are the only users allowed to perform these actions:
  - Create a subscriber account
  - Get a list of subscribers
  - Modify SVC Social Voicemail access
  - Delete SVC Social Voicemail access
  - Get a list of all accounts with SVC Social Voicemail access
- **Subscribers** can access the vast majority of the API operations, including adding or changing personal or SVC Social Voice contacts, and changing their voice-mail access.

SVC includes a default account administrator named **admin**. You set this user’s default password during Service Broker installation. You can use this account to create additional administrator and subscriber accounts.

If security is enabled, client requests to the API must include the credentials of an administrator. The credentials should be supplied in the form of HTTP headers named **name** and **password**.

For example, a GET request submitted with user credentials would appear as follows:

```
GET /svc/subscriber HTTP/1.1
name: admin001
password: password
Host: 10.148.126.226
Accept: */*
```

A request that contains valid credentials but for a resource for which the user does not have privileges returns a **403 Forbidden** error.

Note that the headers appear in HTTP requests in clear text. This may be acceptable in a test or evaluation system. However, in a production system, the connection should use Secure HTTP (HTTPS).

Batch Loader SVC Data Files

The following sections describe the data file format used by the batch loading tools to provision SVC accounts for each type of object used by the SVC services. For each data file, the sections show the name, schema, and sample data. You can find sample data files in these Batch Loader tools directories:

```
Oracle_home/admin_server/utils/bulkloader/onsdata
Oracle_home/admin_server/utils/bulkloader/vmdata
```

The first line in each sample data file shows the data type syntax and is not loaded by the Batch Loader or validation tools. This syntax line is optional in the file.

See "Social Voice Communicator API Reference" for detailed descriptions of the fields for each data type, and the corresponding parameter descriptions for the object creation operations.

All parameters are required unless indicated otherwise.

Table 7–1 lists the data files and their data types.

Table 7–1 Batch Load and Batch Check Data Types

Data Type	Used for	Can be Used for	Resource File in Oracle_home/ocsb61/utils/bukloader/onsdata
contact	Social Circle Contact	Batch loading	**3.contact.csv**
group	Personal Group (public or private)	Batch loading	**5.group.csv**
groupcontact	Private Group Contact	Batch loading	**6.groupcontact.csv**
onenumber	Social Circle One Number	Batch loading	**4.onenumber.csv**
policy	Social Voicemail Policy	Batch loading	**7.policy.csv**
publicgroupcontact	Public Group Contact	Batch loading	**8.publicgroupcontact.csv**
subscriber	Subscriber User	Batch loading and batch checking	**2.subscriber.csv**

Social Circle Contact

The **contact** data type represents an SVC Social Circle contact.

Data Type

contact

Data Format

Subscriber_Name,key,value,default

Example

```
alice,office,alliceoffice@oracle.com,true
```

Notes

See the "Create a Social Circle One Number" operation for more information on the parameters.

Personal Group

The **group** data type represents SVC Social Voice public or private groups. The only difference between public and private data types is that public groups include a **group_type**.

Data Type

group

Data Format - Private Group

Subscriber_Name,Group_Name,Group_Number

Data Format - Public Group

*Subscriber_Name,Group_Name,Group_Number,***public**

Example - Private Group

```
alice,family,456
```

Example - Public Group

```
alice,family,456,public
```

Notes

See "Create a Public Group" or "Delete a Private Group" for more information on the parameters.

Private Group Contact

The **groupcontact** data type represents subscriber's SVC Social Voice private group contacts.

Data Type

groupcontact

Data Format

Subscriber_Name,Group_Name,Contact_Name,number,isActive

Example

```
alice,family,carol,carol@oracle.com,true
```

Notes

See "Create Personal Contacts in a Subscriber Account" for more information on the parameters.

Social Circle One Number

The **onenumber** data type represents SVC Social Circle one numbers.

Data Type

onenumber

Data Format

Subscriber_Name,One_Number,type

Example

```
"alice","123","personal"
```

Notes

See "Create a Social Circle One Number" for more information on the parameters.

Social Voicemail Policy

The **policy** data type represents SVC Social Voicemail policies. You send or retrieve policies as parameters when performing these operations:

- Create a Policy Rule for a Social Circle Contact
- Get a Social Circle Policy Rule Details
- Modify a Social Circle Policy Rule
- Delete a Social Circle Policy Rule

- Get All Social Circle One Number Contacts for a Subscriber
- Get Details for a Social Circle Contact

Data Type
list

Data Format
Subscriber_Name,One_Number,Policy_Name,froms,sequentialCall,sequentialRingInterval, toAddress,priority,schedule

Example

```
alice,123,policy1,address[23456@asfddd.com;group|family,true,,aliceoffice@oracle.c
omalicehome@oracle.com,1,2009|11|24;9|30|0;2012|12|24|;18|0|0;2;-8
```

Notes
See "Create a Policy Rule for a Social Circle Contact" for more information on the parameters.

Public Group Contact

The **publicgroupcontact** data type represents subscriber's SVC Social Voice public group contacts.

Data Type
publicgroupcontact

Data Format
*Owner_Name,Group_Name,Contact_Name,number,isActive,***public**

Example

```
alice,family,carol,carol@oracle.com,true,public
```

Notes
See "Create Personal Contacts in a Subscriber Account" for more information on the parameters.

Subscriber User

This **subscriber** data type represents SVC subscriber administrator accounts.

> **Note:** Administrator accounts cannot be created with the Batch Loader.

Data Type
subscriber

Data Format
username,password,isPONPresented

Example

```
bob,12345,TRUE
```

Notes
See the "Create Subscriber Accounts" operation for parameter details.

Voice-Mail Service

Represents SVC Social Voicemail access.

Data Type

vmbox

Data Format

Subscriber_Name,One_Number

Example

```
alice,123
```

Notes

See "Activate Social Voicemail for Subscribers" for more information on the parameters.

8

Social Voice Communicator API Reference

This chapter describes the Oracle Communications Service Broker Social Voice Communicator (SVC) subscriber management RESTful API, an application programming interface that creates, modifies, and deletes subscriber accounts and data for SVC services.

About the SVC API

This chapter lists RESTful operations for this API, including the parameters accepted and returned by each operation and examples of HTTP requests and responses.

All JSON body parameters for POST (create) operations are required unless indicated as optional. For PUT (modify) operations, the request only needs to contain the attributes to be changed.

See "About the SVC API Operations" for information on using individual operations in this API.

> **Note:** In the request and response message examples in this chapter, line breaks and spaces have been added to the JSON data in the body of the message to improve readability.

SVC Subscriber Account Management Operations

These operations create, update, or delete SVC subscriber accounts.

Create Subscriber Accounts

Creates SVC subscriber accounts for a set of users given their user names and their initial passwords.

An **HTTP response 200** message is returned on success. If one member of the list cannot be created, no accounts are created and a **400 - Subscriber** *subscriber* **already exists** error message is returned.

Authorization

Administrators

HTTP Method

POST

URI

/svc/subscribers

Request Body

The request body parameters accepted by the operation are arrays of:

- **username**: (String) Subscriber's user name.
- **password**: (String) Subscriber's initial password. This is usually supplied by a service provider.
- **isPONPresented**: (Boolean) Optional. If **true**, the subscriber's personal SVC Social Voice group number is used as the caller number when the subscriber initiates a call. If **false**, the subscriber's calling phone number is used. The default value is **false**.

Examples

Example 8–1 Request

```
POST /svc/subscribers HTTP/1.1
Accept: application/json
Authentication: user admin:D@3K4L@F$5S5OIKLS
Content-Length: ??
Content-Type: application/json
Host: ocsb host
[{"username":"bob345","password":"password", true}
{"username":"ming_yu","password":"password"}
{"username":"George_Michaels","password":"password"}
{"username":"555_813_8910","password":"password"}]
```

Get Subscriber Accounts

Returns a list of all subscriber user names and URIs.

Authorization

Administrators

HTTP Method

GET

URI

/svc/subscribers

Request Body

Empty

Response Body

This operation returns a comma-separated list of user names and their access URIs.

Examples

Example 8–2 Request

```
GET http://10.182.98.78:9001/svc/subscribers HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–3 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 105
Server: Lab1(8.0.1.0)
[{"tom":"http://10.182.98.78:9001/svc/subscriber/tom"},
{"alice":"http://10.182.98.78:9001/svc/subscriber/alice"}]
```

Get Subscriber Account Details

Returns details for the subscriber account username with the request.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name*

Request Body

Empty

Response Body

This operation returns these subscriber account details:

- **username**: (String) The subscriber's user name.
- **groups**: (String) The subscriber's group access URI.
- **one_numbers**: (String) The subscriber's SVC Social Circle one number access URI
- **contacts**: (String) The subscriber's contact list URI.
- **isPONPresented**: (Boolean) If **true**, the subscriber's personal SVC Social Circle one number is used as the caller number when the subscriber initiates a call. If **false**, the subscriber's originating phone number is used.

Examples

Example 8–4 Request

```
GET http://10.182.98.78:9001/svc/subscriber/bob HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–5 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 105Server: Lab1(8.0.1.0)
{ "username":"bob",
"groups":"http://localhost:9001/svc/subscriber/bob/groups"
"one_numbers":"http://localhost:9001/svc/subscriber/bob/onenumber",
"contacts":"http://localhost:9001/svc/subscriber/bob/contacts",
"isPONPresented":true}
```

Modify a Subscriber Password or Social Circle One Number Preference

Changes a subscribers's password, SVC Social Circle one number preference, or both.

Authorization

Administrator or subscriber

HTTP Method

PUT

URI

/svc/subscriber/*Subscriber_Name*

Request Body

This operation submits the following parameters:

- **password**: (String) A new password for the subscriber to use.
- **isPONPresented**: (Boolean) If **true**, the subscriber's personal SVC Social Circle one number is used as the caller number when the subscriber initiates a call. If **false**, the MSISDN number used to create the subscriber's account is used.

Examples

Example 8–6 Request

```
PUT http://10.182.98.78:9001/svc/subscriber/bob HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
{"password":"password", "isPONPresented":"true"}
```

Delete a Subscriber Account

Deletes a subscriber's SVC account.

Authorization

Administrator or subscriber.

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name*

Request Body

Empty

Response Body

Empty

Examples

Example 8–7 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/bob HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

SVC Social Circle Personal Contact Management Operations

These operations add, change, delete, or obtain information for SVC Social Circle personal contacts.

Create Personal Contacts in a Subscriber Account

Populates a subscriber's list of contacts from an array that you submit with this operation. Exactly one of the contacts must be given the default value of **true** making it the default contact.

Authorization

Administrator or subscriber.

HTTP Method

POST

URI

/svc/subscribers/*Subscriber_Name***/ contacts**

Request Body

This operation submits the following parameters:

- *Contact_name*:*Contact_number*: (String) or *Contact_name*:*SIP_address*: (String) The name to use for the contacts and their identifying MSISDN number or SIP address in URI format.
- **default**: (Boolean) Set the default contact to **true** and all others to **false**.

Examples

Example 8–8 Request

```
POST http://10.182.98.78:9001/svc/subscribers/bob/contacts HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"mobile":"13811110000","default":true},{"home":"01082311212"},{"office":"alice@o
racle.com"}]
```

Get a Subscriber's Personal Contact List

Returns all personal contact names and access URIs for a subscriber.

Authorization

Administrator or subscriber.

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/contacts**

Request Body

Empty

Response Body

This operation returns an array contact names and their access URIs.

Examples

Example 8–9 Request

```
GET http://10.182.98.78:9001/svc/subscriber/bob/contacts HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–10 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 137
Server: Jetty(8.0.1.0)
[{"office":"http://10.182.98.78:9001/svc/subscriber/alice/contact/office"},
{"home":"http://10.182.98.78:9001/svc/subscriber/alice/contact/home"}]
```

Delete All Personal Contacts

Deletes all members of a subscriber's SVC Social Circle personal contact list.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/contacts**

Request Body

Empty

Response Body

Empty

Examples

Example 8–11 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/bob/contacts HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Get a Personal Contact and its Default Status

Retrieves a subscriber's SVC Social Circle personal contact information and whether the contact number is the default.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/contacts/***Contact_name*

Request Body

Empty

Response Body

This operation returns the following parameters:

- **default**: (Boolean) Specifies whether the contact number is the default number.
- **number**: (String) the contact string. For example a telephone number or email address.

Examples

Example 8–12 Request

```
GET http://10.182.98.78:9001/svc/subscriber/bob/contact/home HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–13 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0)
[{"default":false,"number":"bob@oracle.com"}]
```

Modify a Personal Contact Details and/or Default Status

Modifies an SVC Social Circle personal contact default status or contact information.

Authorization

Administrator or subscriber

HTTP Method

PUT

URI

/svc/subscriber/*Subscriber_Name***/contact/***Contact_name*

Request Body

This operation submits the following parameters:

- **default**: (Boolean) Optional. **true** makes the contact the subscriber's default contact. Non-default numbers do not need an argument.
- **number**: (String) The contact's MSISDN number to add.

Examples

Example 8–14 Request

```
PUT http://10.182.98.78:9001/svc/subscriber/bob/contact/home HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
URI: http://10.182.100.43:9001/svc/subscriber/bob/contact/home
[{"default":true,"number":"222@129.148.75.22"}]
```

Delete a Personal Contact

Deletes an SVC Social Circle personal contact.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/contact/***Contact_name*

Request Body

This operation submits the *Contact_Name* parameter of the contact to delete.

Response Body

Empty

Examples

Example 8–15 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/alice/contact/home HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–16 Response

```
HTTP/1.1 200 OK
[Edit] 2.2
URI: http://10.182.100.43:9001/svc/subscriber/alice/contact
```

SVC Social Circle Number and Contact Management Operations

These operations manage SVC Social Circle one numbers and contacts.

Create a Social Circle One Number

Adds an SVC Social Circle contact to a subscriber account. A subscriber can only have one **personal** number, but any number of **group** numbers.

Authorization

Administrator

HTTP Method

POST

URI

/svc/subscriber/*Subscriber_Name***/onenumbers**

Request Body

This operation submits the following parameters:

- **onenumber**: (String) The SVC Social Circle service number to add.
- **type**: (String) The SVC Social Voice category; default values are **personal** or **group**.

Response Body

Empty

Examples

Example 8–17 Request

```
POST http://10.182.98.78:9001/svc/subscriber/bob/onenumber HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"onenumber":"+123", "type":"personal"},
{"onenumber":"+456", "type":"group"}]
```

Get All Social Circle One Number Contacts for a Subscriber

Returns details for all SVC Social Circle contacts for a subscriber account.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/onenumbers**

Request Body

Empty

Response Body

This operation returns an array of the following parameters for each contact:

- **onenumber**: (String) The name of the SVC Social Circle contact.
- **onenumber_url**: (String) The access URL for the SVC Social Circle contact.
- **type**: (String) The SVC Social Circle type. The default values are **group** and **personal**.

Examples

Example 8–18 Request

```
GET http://10.182.98.78:9001/svc/subscriber/alice/onenumbers HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–19 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Server: Jetty(8.0.1.0)
[{"onenumber":"+123","onenumber_
url":"http://localhost:9001/svc/subscriber/alice/onenumber/+123",
"onenumber_policy":[],"type":"personal"},{"onenumber":"+456",
"onenumber_url":"http://localhost:9001/svc/subscriber/alice/onenumber/+456",
"onenumber_policy":[],"type":"group"}]
```

Delete All Social Circle Contacts

Deletes all SVC Social Circle contacts for a subscriber account.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/onenumbers**

Request Body

Empty

Response Body

Empty

Examples

Example 8–20 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/alice/onenumbers HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Get Details for a Social Circle Contact

Returns details for an SVC Social Circle contact.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/onenumber/***One_Number*

Request Body

Empty

Response Body

This operation returns the following parameters:

- **onenumber**: (String) The SVC Social Circle contact number.
- **type**: (String) The SVC Social Circle type. The default values are **group** and **personal**
- **onenumber_policy**: (Array) A comma-separated list of policy names and access URIs.

Examples

Example 8–21 Request

```
GET http://10.182.98.78:9001/svc/subscriber/alice/onenumber/123 HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–22 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0Server: Jetty(8.0.1.0)
[{"onenumber":"123","type":"personal",
"onenumber_policy":[{"policy1":
"http://localhost:9001/svc/subscriber/bob/onenumber/123/policy/policy1}]
```

Delete a Social Circle Contact

Deletes a SVC Social Circle contact for a subscriber account.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/onenumber/***One_Number*

Request Body

Empty

Response Body

Empty

Examples

Example 8–23 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/alice/onenumber/789 HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Create a Policy Rule for a Social Circle Contact

Creates a behavior policy for an SVC Social Circle contact.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/subscriber/*Subscriber_Name***/onenumber/One_Number/policy/***Policy_name*

Request Body

This operation accepts a policy rule array of these values:

- **froms**: Identifies the numbers based on their address and group values. An empty list matches all calls.
  - **address**: Uses this syntax: *number @domain*.
  - **group**: A group name
- **sequenialCall**: Boolean. If **true**, the value for **sequentialRingInterval** sets the ring interval.
- **sequentialRingInterval**: The number of seconds to allow the call to ring.
- **toAddresses**: The target SIP or Tel URI.
- **schedule**: Specifies the time based routing parameters.
  - **startDate** (**year**, **month day**): The routing policy start day.
  - **startTime** (**hour**, **minute**, **second**): The routing policy start time.
  - **endTime** (**hour**, **minute**, **second**): The routing policy end time.
  - **recurrentPeriod**: The recurring mode. 1 to repeat daily; 2 for Monday through Friday.
  - **timezone**: The routing policy time zone.
  - **until** (**year**, **month**, **day**): A routing policy end date.
  - **priority**: **1** is the highest priority. When a call can be mapped to multiple policies, the highest priority policy is used.

Response Body

Empty

Examples

Example 8–24 Request

```
POST http://10.182.98.78:9001/svc/subscriber/alice/onenumber/123/policy/policy21
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
```

```
Host: 10.182.98.78:9001
[{
 "froms": [],
 "sequencialCall": true,
 "sequentialRingInterval": 30,
 "toAddresses": [
  "tom@oracle.com",
  "0118613611046326"
 ],
 "schedule": {
   "startDate":{
     "year":2009,
     "month":10,
     "day":24
   },
   "startTime":{
     "hour":1,
     "minute":00,
     "second":0
   },
   "endTime":{
     "hour":13,
     "minute":0,
     "second":0
   },
   "recurrentPeriod":1,
   "timezone":+8,
   "until":{
     "year":2010,
     "month":12,
     "day":24
   }
 },
 "priority":10
}]
```

Get a Social Circle Policy Rule Details

Retrieves the details of an SVC Social Circle policy rule.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/onenumber/***One_Number***/policy/***Policy_name*

Request Body

Empty

Response Body

The response includes a policy rule array. For more information on the array syntax, see the "Create a Social Circle One Number" operation.

Examples

Example 8–25 Request

```
GET http://10.182.98.78:9001/svc/subscriber/bob/onenumber/789/policy/policy21
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–26 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Server: Jetty(8.0.1.0)
[{"froms":

[{"address":

  "sip:1234@asfddd.com"}],

 "sequencialCall": false,
   "sequentialRingInterval": 30,

"toAddresses": [

"tom@oracle.com",
   "0118613611046326" ] ,

"schedule":{
   "startDate":{

"year":2009,
"month":11,

 "day":24},
"startTime":{

 "hour":9,

 "minute":30,

 "second":0},

 "endTime":{

 "hour":18,

 "minute":0,

"second":0},

"recurrentPeriod":2,

 "timezone":-8,

"until":{

"year":2010,

 "month":12,

"day":24}},

 "priority":10}]
```

Modify a Social Circle Policy Rule

Makes changes to an SVC Social Circle policy rule object.

Authorization

Administrator or subscriber

HTTP Method

PUT

URI

/svc/subscriber/*Subscriber_Name***/onenumber/***One_Number***/policy/***Policy_Name*

Request Body

This operation submits a policy rule array that replaces a policy rule object. For more information on the policy object, see the "Create a Social Circle One Number" operation.

Examples

Example 8–27 Request

```
PUT http://10.182.98.78:9001/svc/subscriber/alice/onenumber/123/policy/policy21
HTTP/1.1

Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{
 "froms": [],
 "sequencialCall": true,
 "sequentialRingInterval": 30,
 "toAddresses": [
  "tom@oracle.com",
  "0118613611046326"
 ],
 "schedule": {
   "startDate":{
     "year":2009,
     "month":10,
     "day":24
   },
   "startTime":{
     "hour":1,
     "minute":00,
     "second":0
   },
   "endTime":{
     "hour":13,
     "minute":0,
     "second":0
   },
   "recurrentPeriod":1,
   "timezone":+8,
   "until":{
     "year":2010,
     "month":12,
     "day":24
   }
 },
 "priority":10
}]
```

Delete a Social Circle Policy Rule

Removes a policy from a subscriber’s SVC Social Circle contact.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/onenumber/***One_Number***/policy/***Policy_Name*

Request Body

Empty

Response Body

Empty

Examples

Example 8–28 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/alice/onenumber/789/policy/policy21
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

SVC Social Voice Private Group Management Operations

These operations administer SVC Social Voice personal groups.

Create a Private Group

Adds a SVC Social Voice private group to a subscriber's account.

Authorization

Administrator or subscriber

HTTP Method

POST or PUT

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name*

Request Body

This operation submits the following parameters:

- **grp_onenumber**: (String) The group access number to add. Can be any string except **Others** and **All Contacts** which are reserved for Oracle use.

Response Body

Empty. Returns an **HTTP 200** message on success and an **HTTP 400** message if the group number does not exist.

Examples

Example 8–29 Request

```
POST http://10.182.98.78:9001/svc/subscriber/alice/group/family HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_onenumber":"5556700"}]
```

Delete a Private Group

Deletes a subscriber's SVC Social Voice private group and all of its members.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name*

Request Body

Empty

Response Body

Empty. Returns an **HTTP 200** message on success.

Examples

Example 8–30 Request

```
DELETE http://10.182.98.78:9001/svc/subscriber/alice/group/family HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–31 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Server: Jetty(8.0.1.0)
```

Add a Private Group Contact

Adds a SVC Social Voice contact name, number, and status to a private group. The **grp_onenumber** parameter is required to add the contact to a public group.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name* / **contact/***Contact_Name*

Request Body

This operation submits the following parameters:

- **grp_contact**: (Array) An array containing the contact name, contact number, and a boolean value indicating active/inactive. **grp_contact** uses this syntax:

  ```
  ["grp_contact":["contact_name","contact_number","true | false"]}
  ```

Example 8–32 shows a sample **grp_contact** array.

Response Body

Empty

Examples

Example 8–32 Request

```
POST hhttp://10.182.98.78:9001/svc/subscriber/alice/group/family/contact/bob
HTTP/1.1

Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_contact":["bob", "3869", "false"]}]
```

Get All Private Group Memberships for a Subscriber

Returns the type, URI, and name of all private Social Voice groups that a subscriber is a member of.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/groups**

Request Body

Empty

Response Body

This operation returns an array of these parameters for each of the group:

- **groupType**: (String) Can be **public** or **private**.
- **groupURL** (String) - The URL of the group with the syntax: *IP_Address*:/svc/subscriber/subscriber_name/group/*group_name*.
- **groupName** (String) - The name of the group.

The array uses this syntax:

```
[{"groups":"[{groupType":" "public | private" "groupUrl":"URL","groupName":"Group_
Name"}
```

Examples

Example 8–33 Request

```
GET http://10.182.98.78:9001/svc/subscriber/bob/groups HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–34 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 104
Server: Jetty(8.0.1.0)
[{groups:": {"groupType""Private",
     "groupURL":"http://10.182.98.78:8080/svc/subscriber/sunzhen/group/default_
group",
"groupName":"default_group" }{"groupType""Private",
"groupURL":"http://10.182.98.78:8080/svc/subscriber/sunzhen/group/public_group",
groupName":"public_group" }
"]}
```

Modify a Private Group’s One Number

Changes a subscriber’s SVC Social Voice private one number value.

Authorization

Administrator or subscriber

HTTP Method

PUT

URI

/svc/subscriber/*Subscriber_Name***/group/***Group_name*

Request Body

This operation submits any of the follows parameters:

- **grp_onenumber**: (String) The group’s SVC Social Voice access number.

Response Body

Empty. Returns an **HTTP 200** message on success and an **HTTP 400** message if the group number does not exist.

Examples

Example 8–35 Request

```
PUT http://10.182.98.78:9001/svc/subscriber/alice/group/family HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_onenumber":"family"]
```

Example 8–36 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Server: Jetty(8.0.1.0)
```

Get All Contacts for a Private Group

Returns an array of a private group's SVC Social Voice contact group names, URIs, and types.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name***/contacts**

Request Body

Empty

Response Body

This operation returns an array of the group contact URIs. Example 8–38 shows an example array.

Examples

Example 8–37 Request

```
GET http://10.182.98.78:9001/svc/subscriber/cindy/group/friends/contacts HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–38 Response

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Jetty(8.0.1.0)
[{

 "contacts":
   [
       {
           "contactUrl":
"http://239.255.255.255:8080/svc/subscriber/fred/group/publicGroup/contact/c133735
3829328",
           "isactive": true,
           "name": "c1337353829328",
           "subscriberName": "alice",
           "number": "7777",
           "type": 3
       },
       {
           "contactUrl":
"http://239.255.255.255:8080/svc/subscriber/fred/group/publicGroup/contact/c133735
2487076",
           "isactive": false,
           "name": "c1337352487076",
           "subscriberName": "wangqian",
           "number": "5554321178",
           "type": 2
       }
   ]
}
]
```

Get Details for a Private Group Contact

Returns an array with the SVC Social Voice private group contact name, number, and active/inactive status.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name***/contact/***Contact_Name*

Request Body

Empty

Response Body

This operation submits the following parameters:

- **grp_contact**: (Array) An array containing the contact name, contact number, and a boolean value indicating active/inactive. **grp_contact** uses this syntax:

```
["grp_contact":["contact_name","contact_number","true | false"]}
```

Example 8–40 shows a sample **grp_contact** array.

Examples

Example 8–39 Request

```
GET http://10.182.98.78:9001/svc/subscriber/alice/group/family/contact/henry
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–40 Response

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Jetty(8.0.1.0)
[{
"contacts":
[{"grp_
contact":"http://239.255.255.255:8080/svc/subscriber/fred/group/publicGroup/contac
t/c1337353829328","isactive":
true,"name":"c1337353839327","subscriberName":"alice",number": "7777","type": 3},{
"grp_contact":
"http://239.255.255.255:8080/svc/subscriber/fred/group/publicGroup/contact/c133735
2487076",
"isactive": false,"name": "c1337352487076",subscriberName": "wangqian",number":
"wangqian@mydomain.com","type": 2}]} ]
```

Get Private Group Details

Retrieves the name, SVC Social Voice group number, and/or URI of a subscriber's Social Voice contact private group.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name*

Request Body

This operation submits the follows group parameters:

- **grp_onenumber**: (String) The group's SVC Social Voice group number access number.
- **grp_name**: (String) The name of the group.
- **grp_contacts**: (String) The URI to access group contacts.

Response Body

Empty

Examples

Example 8–41 Request

```
POST http://10.182.98.78:9001/svc/subscriber/alice/group/family HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_onenumber":"123456", "grp_name":"friends","grp_
contacts":"http://10.182.98.78:9001/svc/subscriber/cindy/group/friends/contacts"}]
```

Modify a Private Group Contact

Replaces a SVC Social Voice private contact group URI with a new one. All changes in the URI are accepted.

Authorization

Administrator or subscriber

HTTP Method

PUT

URI

/svc/subscriber/*Subscriber_Name***/group/***Private_Group_Name***/contact/***Contact_Name*

Request Body

This operation submits the following parameters:

- **grp_contact**: (Array) An array containing the contact name, contact number, and a boolean value indicating active/inactive. **grp_contact** uses this syntax:

  ```
  ["grp_contact":["contact_name","contact_number","true | false"]}]
  For example:
  ["grp_contact":["bob","bob499@domain.com","true"]}]
  ```

-

Example 8–42 shows a sample **grp_contact**array.

Examples

Example 8–42 Request

```
PUT
http://10.182.98.78:9001/svc/subscriber/alice/group/family/contact/fred HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_contact":["fred", "3869", "false"]}]
```

Delete a Private Group Contact

Removes a contact from a subscriber's SVC Social Voice private group.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/group/***Group_name***/contact/***Contact_name*

Request Body

Empty

Response Body

Empty

Examples

Example 8–43   Request

```
DELETE
http://10.182.98.78:9001/svc/subscriber/alice/group/family/contact/cindy HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

SVC Social Voice Public Group Management Operations

These operations administer SVC Social Voice public groups.

Create a Public Group

Adds a SVC Social Voice group to a subscriber's account. If just the **grp_onenumber** parameter is used, a private group for the subscriber is created. If both **grp_onenumber** and a **group_type** of **public** are sent, a public group is created.

Authorization

Administrator or subscriber

HTTP Method

POST or PUT

URI

/svc/subscriber/*Subscriber_Name***/group/***Public_Group_Name*

Request Body

This operation submits the following parameters:

- **grp_onenumber**: (String) The group identifier (usually an informal name) to add. Can be any string except **Others** and **All Contacts** which are reserved for Oracle use.
- **group_type**: (String): (Optional) Creates a public group. A value of **public** is required if this parameter is used. The parameter itself is optional, and a private group is created if it is not present.

Response Body

Empty. Returns an **HTTP 200** message on success and an **HTTP 400** message if the group number does not exist.

Examples

Example 8–44 Request

```
POST
http://10.182.98.78:9001/svc/subscriber/alice/group/fballteam
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"grp_onenumber":"8888","group_type":"Public"}]
```

Delete a Public Group

Deletes a Social Voice public group. Only empty groups can be deleted. Use the "Delete a Public Group Contact" operation to remove the public group members first, then use this operation to delete the public group itself.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/group/***Public_Group_Name*

Request Body

Empty.

Response Body

Empty

Examples

Example 8–45 Request

```
POST
http://10.182.98.78:9001/svc/subscriber/alice/group/bballteam
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Add a Public Group Member

Adds a member to a Social Voice public group. The added entity can be one of these:

- A **grp_contact** array including the subscriber's name, one-number contact number, and a status of either **active** or **inactive**.
- Just the subscriber's personal one number.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/subscriber/*Subscriber_Name***/group/***Public_Group_name***/contact/newcontact/***Group_Owner_Name*

Request Body - Array

This operation submits the following parameters:

- **grp_contact**: (Array) An array containing the contact name, contact number, and a boolean value indicating active/inactive. **grp_contact** uses this syntax:

  ```
  ["grp_contact":[{"personal_onenumber", "is_active"]
  ```

  For example:

  ```
  ["grp_contact":"bob499@domain.com","true"]
  ```

- Just a subscriber's one number with this syntax:

  ```
  {"grp_onenumber": ["personal_onenumber"]}
  ```

For example:

```
["grp_onenumber":"41555876"]
```

Where:

personal_onenumber: (String) The identifier of the subscriber to add to the group.

is_active: (Boolean) Specified whether the member is active.

Response Body

Empty

Examples

Example 8–46 Request

```
POST
http://10.182.98.78:9001/svc/subscriber/alice/group/bballteam/contact/newcontact/s
unzhen HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

```
["grp_contact":[{"personal_onenumber":"4155551233","false"]]
```

Delete a Public Group Contact

Deletes a member of a public group.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/subscriber/*Subscriber_Name***/group/***Public_Group_name***/contact/newcontact/***Public_Group_Owner*.

Request Body

Empty

Response Body

Empty

Examples

Example 8–47 Request

```
DELETE
http://10.182.98.78:9001/svc/subscriber/alice/group/family/contact/c1337181093/ali
ce HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Get All Public Groups for a Subscriber

Returns the type, URI, and name of all Social Voice groups that a subscriber owns.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/inpublic/groups**

Request Body

Empty

Response Body

This operation returns an array of these parameters for each of the group:

- **groupType**: (String) Can be **public** or **private**.
- **groupURL** (String) - The URL of the group.
- **groupName** (String) - The name of the group.

The array uses this syntax:

```
[{"groups":"[{groupType":" "public" "groupUrl":"URL","groupName":"Group_Name"}]
```

Examples

Example 8–48 Request

```
GET http://239.255.255.255:8080/svc/subscriber/sunzhen/public/groups HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–49 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 104
Server: Jetty(8.0.1.0)
{"groups":[
{"groupType":"Public","groupUrl":"http://10.182.98.78:8080/svc/subscriber/bob/grou
p/default_group","groupName":"default_group"},
{"groupType":"Public","groupUrl":"http://10.182.98.78:8080/svc/subscriber/bob/grou
p/privateGroup","groupName":"privateGroup"}]}
```

Get All Public Group Memberships for a Subscriber

Retrieves a list of all the public groups that a subscriber is a member of.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/inpublic/groups**

Request Body

Empty

Response Body

An array of the **groups** that the subscriber is a member of, with these fields:

- **ownerName**: (String) - The name of the group owner.
- **groupName**: (String) - The name of the group.

Examples

Example 8–50 Request

```
GET http://10.182.98.78:9001/svc/subscriber/alice/inpublic/groups HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–51 Return

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Jetty(8.0.1.0)
{"groups":[{"ownerName":"alice","groupName":"team2"}]
[{"ownerName":"alice","groupName":"team3}]]
```

Get All Contacts for a Public Group

Retrieves a list of all members of a public group.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/group/***Public_Group_Name***/contacts/***Group_Owner*

Request Body

Empty

Response Body

This operation returns an array of member with these parameters:

- **isactive**: (Boolean) - Returns true if the group is currently active.
- **name**: (String) - The name of the group.
- **subscriberName**: (String) - The name of the group contact.
- **number**: (String) - The group member's identifying number.
- **type**: (String) - Can be one of:
  - **1**: A **grp_contact** array including the member's
  - **3**: The subscriber's *One_Number*.

Examples

Example 8–52 Request

```
GET http://10.182.98.78:9001/svc/subscriber/alice/group/team1/contacts/cindy
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–53 Return

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Jetty(8.0.1.0)
[{"contacts":
{"isactive": true,"name": "c1337353829328","subscriberName": "alice","number":
"7777","type": 3},

{"isactive": false,"name": "team1","subscriberName": "wangqian","number":
"wangqian@open-ims.test","type": 1}]}
```

SVC Click-to-Dial Make Call Operation

This operation makes a Click-to-Dial call.

Make a Click-to-dial Call

Connects the subscriber's SVC Social Circle default contact number on a website or email server to a callee on a SIP or SS7 telecom network.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/subscriber/*Subscriber_Name***/makecall?to=***callee*

Request Body

Empty

Response Body

Empty

Examples

Example 8–54 Request

```
GET http://10.182.98.78:9001/svc/subscriber/george/makecall?to=5551234567 HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

SVC Social Voicemail Access Management Operations

These operations manage SVC Social Voicemail access.

Activate Social Voicemail for Subscribers

Activates SVC Social Voicemail features for existing subscribers.You submit an array of subscriber user names and passwords with the request to activate accounts for those subscribers. If any invalid or pre-existing subscribers are submitted, the entire operation fails and none of the subscribers are granted access to SVC Social Voicemail.

If a pre-existing voicemail box is sent with the request, an **HTTP_BAD_REQUEST (400)** message is returned.

Authorization

Administrator

HTTP Method

POST

URI

/svc/admin/vmboxes

Request Body

This submits an array of user name/password combinations. Voice-mail is activated for all of the users in the array.

- **username**: (String) The subscriber's user name.
- **one_number**: (String) The subscriber's MSISDN SVC Social Circle one number

Examples - check these

Example 8–55 Request

```
POST http://10.182.98.78:9001/svc/admin/vmboxes HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001{"username":"bob345","onenumber":"123456"},
[{"username":"ming_yu","onenumber","5555551234"},
{"username":"George_Michaels","onenumber":"4155550999"},
{"username":"555_813_8910","onenumber":"2135550444"}]
```

Activate/Deactivate Social Voicemail Access

Changes SVC Social Voicemail access for a subscriber. You submit the subscriber's user name and SVC Social Circle one number and an **isActive** setting of **true** or **false** to activate or deactivate the subscriber's account. No resources or data are removed when the account is deactivated.

Use this operation to deactivate a subscriber's SVC Social Voicemail account temporarily.

To delete a subscriber's account resources and data permanently, see "Delete Social Voicemail Access and Data for a Social Circle".

Authorization

Administrator

HTTP Method

PUT

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/admin**

Request Body

This operation submits the following parameters:

- **isActive**: (Boolean) A value of **true** activates subscriber voice-mail access and **false** deactivates it.

Response Body

Empty

Examples

Example 8–56 Request

```
PUT https://10.182.98.78:9001/svc/vmsubscriber/bob345/onenumber/123456/vmbox/admin
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{ "isAcitve":true }]
```

Delete Social Voicemail Access and Data for a Social Circle

Deactivates a subscribers's SVC Social Voicemail access, and deletes all SVC Social Voicemail data and recordings associated with that subscriber. Use this operation when you do not expect the subscriber to ever need the SVC Social Voicemail account or data ever again.

To deactivate a subscriber's account without removing and data (temporary deactivation), see "Activate/Deactivate Social Voicemail Access".

Authorization

Administrator

HTTP Method

DELETE

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/admin**

Request Body

Empty

Response Body

Empty

Examples

Example 8–57 Request

```
DELETE
https://10.182.98.78:9001/svc/vmsubscriber/bob345/onenumber/123456/vmbox/admin
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Get All Social Voicemail Subscriber Accounts

Returns a list of all SVC Social Voicemail accounts, and indicates whether those accounts are active or inactive. The response includes each user's user name, SVC Social Circle one number, the URI of their SVC Social Voicemail box, and whether the voice permission is active or inactive.

Authorization

Administrator

HTTP Method

GET

URI

/svc/admin/vmboxes

Request Body

Empty

Response Body

The response includes the following parameters for each subscriber:

- **subscriber**: (String) The subscriber's user name.
- **one_number**: (String) The subscriber's MSISDN SVC Social Circle one number.
- **address**: (String) The URI of the subscriber's voice-mail box.
- **isActive**: (Boolean) **true** for subscribers with active voice-mail access and **false** for subscribers with inactive voice-mail access.

Examples

Example 8–58 Request

```
GET http://10.182.98.78:9001/svc/admin/vmboxes HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–59 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Server: Jetty(8.0.1.0)
Content-Length: 137
[{"subscriber":"bob345","onenumber":"123456","address":"/svc/vmsubscriber/bob345/o
nehumber/123456/vmbox","isActive": true},
{"subscriber":"ming","onenumber":"5555551234","address":"/svc/vmsubscriber/ming/on
enumber/5555551234/vmbox","isActive": true},
{"subscriber":"George_
Michaels","onenumber":"4155550999","address":"/svc/vmsubscriber/George_
Michaels/onenumber/4155550999/vmbox","isActive": true},
{"subscriber":"555_813_
8910","onenumber":"2135550444","address":"/svc/vmsubscriber/555_813_
8910/onenumber/2135550444/vmbox","isActive": true}]
```

Get Social Voicemail Settings

Get these SVC Social Voicemail settings for a subscriber:

- Whether to route unanswered calls to a SVC Social Voicemail box.
- The greeting message.
- The name message.
- The SVC Social Voicemail box password.
- Voice-mail password status.
- A time limit for storing old messages.
- An email address to send a “you have new voice-mail message” announcement mail to. This feature is inactive if no email address is submitted.
- The maximum number of inbound messages stored in the voice-mail box.
- The maximum number of outbound messages.
- The maximum length for a voice-mail message.

Authorization

Administrator or subscriber.

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox**

Request Body

Empty

Response Body

The request body parameters accepted by the operation include:

- **enableVoiceMail**: (Boolean) **true** (default value) routes unanswered incoming calls to the subscriber’s voice-mail box. **false** drops unanswered calls.
- **password**: (String or null) Must be either a numeric string at least six digits long, or null. A null value deactivates the voice-mail account’s password protection.
- **voiceMessageExpireDays**: (Digit String) The number of days that voice-mail messages are stored before being automatically deleted. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumInboundMessagesInMailbox**: (Digit String) The maximum number of voice-mail messages that the voice-mail feature stores. Once this limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumOutboundMessagesInMailbox**: (Digit String) The maximum number of uploaded messages to store. Once this limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.

- **maximumPhoneMessageDuration**: (Digit String) the maximum number of seconds for a voice-mail message duration. Recording stops when the limit is reached. If no value is submitted, Service Broker uses the system MBean setting.
- **notifyEmailAddress**: (String or null) A valid email address or a null value. If an email address is submitted with this parameter, Service Broker sends an email message to this address each time a voice-mail message is received. A null value prohibits the voice-mail box from sending voice-mail announcements. This parameter is void if an administrator has deleted or deactivated the voice-mail box account.

Examples

Example 8–60 Request

```
PUT https://10.182.98.78:9001/svc/vmsubscriber/bob345/onenumber/1234567/vmbox
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–61 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"enableVoiceMail":true, "voiceMessageExpireDays":60,
"maximumInboundMessagesInMailbox":20,  "isPasswordEnable":true} ]
"maximumOutboundMessagesInMailbox":20, "maximumPhoneMessageDuration":300,
"notifyEmailAddress":"bob@bob.com","greeting" :
"/svc/vmsubscriber/alice/onenumber/123321/vmbox/greeting" ,
"voiceName":"/svc/vmsubscriber/bob/onenumber/123321/vmbox/name",
```

Modify Social Voicemail Settings

Change any of these SVC Social Voicemail settings for a subscriber:

- Whether to route unanswered calls to a SVC Social Voicemail box.
- The greeting message.
- The name message.
- The SVC Social Voicemail box password.
- Voice-mail password status.
- A time limit for storing old messages.
- An email address to send a "you have new voice-mail message" announcement mail to. This feature is inactive if no email address is submitted.
- The maximum number of inbound messages stored in the voice-mail box.
- The maximum number of outbound messages.
- The maximum length for a voice-mail message.

Authorization

Administrator or subscriber.

HTTP Method

PUT

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox**

Request Body

The request body parameters accepted by the operation include:

- **enableVoiceMail**: (Boolean) **true** (default value) routes unanswered incoming calls to the subscriber's voice-mail box. **false** drops unanswered calls.
- **password**: (String or null) Must be either a numeric string at least six digits long, or null. A null value deactivates the voice-mail account's password protection.
- **voiceMessageExpireDays**: (Digit String) The number of days that voice-mail messages are stored before being automatically deleted. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumInboundMessagesInMailbox**: (Digit String) The maximum number of voice-mail messages that the voice-mail feature stores. Once this limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumOutboundMessagesInMailbox**: (Digit String) The maximum number of uploaded messages to store. Once this limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumPhoneMessageDuration**: (Digit String) the maximum number of seconds for a voice-mail message duration. Recording stops when the limit is reached. If no value is submitted, Service Broker uses the system MBean setting.

- **notifyEmailAddress**: (String or null) A valid email address or a null value. If an email address is submitted with this parameter, Service Broker sends an email message to this address each time a voice-mail message is received. A null value prohibits the voice-mail box from sending voice-mail announcements. This parameter is void if an administrator has deleted or deactivated the voice-mail box account.

Response Body

The response body includes a **success** status for each of the changed settings.

Examples

Example 8–62 Request

```
PUT https://10.182.98.78:9001/svc/vmsubscriber/bob345/onenumber/1234567/vmbox
HTTP/1.1

Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"enableVoiceMail":true, "voiceMessageExpireDays":60,
"maximumInboundMessagesInMailbox":20, "maximumOutboundMessagesInMailbox":20,
"maximumPhoneMessageDuration":300, "notifyEmailAddress":"bob@bob.com","greeting" :
"/svc/vmsubscriber/alice/onenumber/123321/vmbox/greeting" ,
"voiceName":"/svc/vmsubscriber/bob/onenumber/123321/vmbox/name",
"isPasswordEnable":true} ]
```

Example 8–63 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"enableVoiceMail":true, "voiceMessageExpireDays":success,
"maximumInboundMessagesInMailbox":"success",
"maximumOutboundMessagesInMailbox":"success",
"maximumPhoneMessageDuration":"success",
"notifyEmailAddress":"success","greeting" : "success" , "voiceName":"success",
"isPasswordEnable":"success"} ]
```

SVC Social Voicemail Message Management Operations

These operations manage SVC Social Voicemail messages.

Delete a Social Voicemail Personal Greeting Message

Cancels a subscriber's personal greeting message by deleting their **personalGreeting.wav file.** The default greeting file is used instead.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/greeting**

Request Body

Empty

Response Body

Empty

Examples

Example 8–64 Request

```
DELETE
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/greeting
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Delete a Personal Voice Name Message

Removes a subscriber’s personal greeting message.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/name**

Request Body

Empty

Response Body

Empty

Examples

Example 8–65 Request

```
DELETE https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/name
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Modify Social Voicemail Message Public/Private Setting

Sets a SVC Social Voicemail messages as either publicly or privately accessible.

Authorization

Administrators or subscribers

HTTP Method

PUT

URI

/svc/vmsubscriber/_Subscriber_Name_**/onenumber/**_One_Number_**/vmbox/message/**_Message_Name_

Request Body

This operation submits the following parameter:

- **isPublic** (Boolean) **true** if this message is publicly accessible; **false** if it is only accessible by the recipient.

Response Body

Empty

Examples

Example 8–66 Request

```
PUT
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/message/abc
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"ispublic":true}]
```

Deactivate a Social Voicemail Name Message

Deactivates a subscriber's SVC Social Voicemail name message.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

`/svc/vmsubscriber/`*`Subscriber_Name`*`/onenumber/`*`One_Number`*`/vmbox/name`

Request Body

Empty

Examples

Example 8–67 Request

```
DELETE https://10.182.98.78:9001/vmsubscriber/bob345/onenumber/123456/vmbox/name
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Delete a Social Voicemail Message

Deletes a SVC Social Voicemail message.

Authorization

Administrator or subscriber

HTTP Method

DELETE

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/message/***Message_Name*

Request Body

Empty

Response Body

Empty

Examples

Example 8–68 Request

```
DELETE
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/message/abc.w
av HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Get a List of Social Voicemail Messages

Returns a list of all the SVC Social Voicemail message names and types for a subscriber.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/messages**

Request Body

Empty

Response Body

On success the response includes an **HTTP 200 OK** message and an array of entries with the syntax: *`messageName`* **:** *`type`*. Where **type** is one of **receivedMessage** or **sentMessage**.

Examples

Example 8–69 Request

```
GET https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/messages
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–70 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"messageName":""sample_123124","type":"ReceivedMessage"},
{"messageName":"fridaymessage_
123124","type":"SentMessage"},{"messageName":""20100102063002_
123124","type":"ReceivedMessage"}, {"messageName":""sample_
greeting","type":"SentMessage"}]
```

Get a Social Voicemail Message File

Gets a SVC Social Voicemail message WAV file.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/message/***Message_Name***/file**

Request Body

Empty

Response Body

Empty. On success the response includes an **HTTP 200 OK** message and the bit stream of the WAV file requested.

Examples

Example 8–71 Request

```
GET
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/message/abc.w
av HTTP/1.1

Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–72 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
Host: 10.182.98.78:9001
Content-Length: 231837
Content-Type: multipart/form-data; boundary="----=_Part_0_20761102.1296024147640"
MIME-Version: 1.0
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
Content-Length: 231837
------=_Part_0_20761102.1296024147640
Content-Type: audio/x-wav; name=test.wav
Content-Transfer-Encoding: binary
Content-Disposition: form-data; name="message1.wav"; filename="name.wav"
(binary data)
```

Get Details for a Social Voicemail Message

Returns details about a SVC Social Voicemail message, including:

- The message type.
- The date and time it was created.
- The caller name and number.
- The file location.
- Whether the message is public or private.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_ Number***/vmbox/message/***Message_Name*

Request Body

Empty

Response Body

The response includes the following parameters:

- **type**: (String) Can be one of **SentMessage** or **ReceivedMessage**.
- **createTime:** (UTC Timestamp) The time the message was created.
- **callerName:** (String) The name of the caller leaving the message.
- **callerNum**: (Digit string) The phone number of the caller leaving the message.
- **fileLocation**: (String) The relative address to the voice name message file.
- **isPublic**: (Boolean) **true** if this message is publicly accessible; **false** if it is only accessible by the recipient.

Examples

Example 8–73 Request

```
GET
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/message/abc.w
av HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–74 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
Content-Length: 43
Server: Jetty(8.0.1.0
[{"type":"SentMessage","time":"201101201333","callerName":"john","callerNum":"3231
","fileName":"abc.wav", "fileLocation":"/21/32/3231/abc.wav","isPublic":"true"} ]
```

Get Details for All of a Subscriber’s Voice Mail Messages

Returns the caller name and number, message name, creation time,   and message file location of all voice mail messages in a subscriber’s voice mail box.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/detail/messages**

Request Body

Empty

Response Body

The response includes an array of voice mail messages, one per line with these parameters:

- **callerNumber**: (Digit string) The phone number of the caller leaving the message.
- **createTime:** (UTC Timestamp) The time the message was created.
- **fileLocation**: (String) The relative address to the voice name message file, ending with the file name.
- **callerName:** (String) The name of the caller leaving the message.
- **messageName**: (String) An identifier for the uploaded file created by combining the messages’s **createtime** and **onenumber** values with this syntax: `createtime_ onenumber`.
- **type**: (String) Can be one of **SentMessage** or **ReceivedMessage**

Examples

Example 8–75 Request

```
GET
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/detail/messag
es
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–76 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"callerNumber":"alice@open-ims.test","createTime":1334134003000,
"fileLocation":"./2/27/20122012/inbound/20120411164643_20122012.wav",
"callerName":"alice","type":"ReceivedMessage","messageName":"20120411164643_
20122012"},
{"callerNumber":"tel:20122012","createTime":1334134216000,
"fileLocation":"./2/27/20122012/outbound/1334134216468.wav",
"callerName":"bob","type":"SentMessage","messageName":"20122012_1334134216468"}]
```

Get Personal Greeting Message Settings

Retrieves the location and status of a subscriber′s voice greeting message.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/greeting**

Request Body

Empty

Response Body

This operations returns the following parameters:

- **isPersonalActive**: (Boolean) **true** indicates that the voice-mail personal greeting is activated; **false** indicates that the voice-mail greeting is not activated.
- **fileLocation**: (String) Returned if **isPesonalActive** is **true**; the relative address to the voice name message file.

Examples

Example 8–77 Request

```
GET https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/greeting
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–78 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"isPersonalActive":"true","fileLocation":"123456/vmbox/greeting" }]
```

Get Social Voicemail Message Settings

Returns these SVC Social Voicemail settings for a subscriber:

- Whether to route calls to a voice-mailbox.
- The Social Voicemail service password.
- The number of days to keep a voice-mail message is stored before it is automatically deleted.
- The maximum number of inbound messages in the voice-mailbox.
- The maximum number of outbound messages.
- The maximum length for a voice-mail message.
- An email address to send a “new voice-mail message” announcement email to.

Authorization

Administrator or subscriber.

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox**

Request Body

The request body parameters accepted by the operation include:

- **enableVoiceMail**: (Boolean) A value of **true** routes unanswered calls to the subscriber′s voice-mail. A value of **false** drops unanswered calls.
- **password** - (Digit string) The password used by subscribers to access their voice-mail service. Must be at least six digits long. A null value deactivates the password and prevents the subscriber from accessing the voice-mail service. Can be reset by subscriber.
- **voiceMessageExpireDays**: (Digit string) The number of days that voice-mail is stored before being automatically deleted. This setting cannot be longer than the system MBean setting.
- **maximumInboundMessagesInMailbox**: (Digit string) The maximum number of voice-mail messages that the voice-mail feature stores. Once the limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumOutboundMessagesInMailbox**: (Digit String) The maximum number of uploaded messages to store. Once this limit is reached, the oldest message is deleted when a new message is stored. If no value is submitted, Service Broker uses the system MBean setting.
- **maximumPhoneMessageDuration**: (Digit String) the maximum number of seconds for a voice-mail message duration. Recording stops when the limit is reached. If no value is submitted, Service Broker uses the system MBean setting.

- **notifyEmailAddress** - (String) The email address to receive an email notification when a voice-mail message arrives.   A null value prohibits the SVC Social Voicemail service from sending voice-mail announcements.

Response Body

The response includes the same parameters as the request, with a **success** or **failure** value in each parameter field to indicate whether the field was successfully changed.

Examples

Example 8–79   Request

```
GET https://10.182.98.78:9001/vmsubscriber/bob345/onenumber/123456/vmbox HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
[{"enableVoiceMail":true, "password":"password", "voiceMessageExpireDays":60,
"maximumInboundMessagesInMailbox":20, "maximumOutboundMessagesInMailbox":20,
"maximumPhoneMessageDuration":300, "notifyEmailAddress":"bob345@bob.com", } ]
```

Example 8–80   Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[ {"enableVoiceMail":"success","maximumInboundMessagesInMailbox":"success",
"maximumInboundMessagesInMailbox":"success",maximumPhoneMessageDuration":"success"
,
"notifyEmailAddress":
"success","voiceMessageExpireDays":"success","password":"failure: password should
be six digit numeric type. "} ]
```

Get Social Voicemail Name Message Settings

Retrieves the location and status (**active**/**inactive**) of a subscriber's SVC Social Voicemail name message.

Authorization

Administrator or subscriber

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/name**

Request Body

Empty

Response Body

This operations returns the following parameters:

- **isPersonalActive**: (Boolean) **true** indicates that the voice-mail name is activated; **false** indicates that the voice-mail message is not activated.
- **fileLocation**: (String) Present if **isPesonalActive** is **true**; the relative address to the voice name message file.

Examples

Example 8–81 Request

```
GET https://ocsb_hostport/vmsubscriber/bob345/onenumber/123456/vmbox/name HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–82 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"isPersonalActive":"true","fileLocation":"123456/vmbox/name" }]
```

Send a Social Voicemail Message to Other Subscribers

Sends a message from one SVC Social Voicemail subscriber to another. Used by "Create sentMessage Voice Message" to sent messages to other subscribers.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/message/***Message_Name***/sendmessage**

Request Body

This operation submits the following parameters:

- **receiverOneNumbers**: (Array of digit strings) An array of the SVC Social Circle one numbers to send the message to.

Response Body

This operation returns the following parameters:

- **receiverOneNumbers**: (Array of digit strings) An array of SVC Social Circle one numbers that the message was successfully sent to.

Examples

Example 8–83 Request

```
POST
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/test.wav/send
message HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
{"receiverOneNumbers": ["122345", "3211"]}
```

Example 8–84 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0)
[ {"receiverOneNumbers": ["122345", "3211"]} ]
```

Create Social Voicemail Greeting Message

Sets (records) the subscriber's SVC Social Voicemail service greeting as **personalgreeting.wav** in the subscriber's greeting folder.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/vmsubscriber/ *Subscriber_Name* **/onenumber/** *One_Number* **/vmbox/greeting**

Request Body

This operation does not submit any parameters, but it does require the voice-mail greeting message WAV file as an attachment.

Response Body

Empty

Examples

Example 8–85 Request

```
POST https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/greeting
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: multipart/form-data; boundary="----=_Part_0_20761102.1296024147640"
MIME-Version: 1.0
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
Content-Length: 231837
------=_Part_0_20761102.1296024147640
Content-Type: audio/x-wav; name=greeting.wav
Content-Transfer-Encoding: binary
Content-Disposition: form-data; name="greeting.wav"; filename="greeting.wav"
(binary data)
```

Create Social Voicemail Box Name Message

Sets a WAV file as the subscriber's SVC Social Voicemail service greeting message. The WAV file is renamed to **name.wav** and placed in the subscriber's **/onenumber/***One_Number***/vmbox/name** directory.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/name**

Request Body

This operation does not submit any parameters, but the HTTP request must include the WAV file to use as a voice name message as an attachment.

Examples

Example 8–86 Request

```
POST https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/name
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: multipart/form-data; boundary="----=_Part_0_20761102.1296024147640"
MIME-Version: 1.0
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
Content-Length: 231837
------=_Part_0_20761102.1296024147640
Content-Type: audio/x-wav; name=name.wav
Content-Transfer-Encoding: binary
Content-Disposition: form-data; name="name.wav"; filename="name.wav"
(binary data)
```

Create sentMessage Voice Message

Uploads a WAV message file. This operation appends the SVC Social Circle one number prefix to the filename root to create a unique name. For example, if the subscriber with an SVC Social Circle one number prefix of **123** uploads a files called **vm34.wav** the uploaded file is named **vm34_123.wav**.

After uploading, use:

- "Get Details for a Social Voicemail Message" to get the message.
- "Send a Social Voicemail Message to Other Subscribers" to send this message to other users.
- "Modify Social Voicemail Message Public/Private Setting" to make the message available to the public.

Authorization

Administrator or subscriber

HTTP Method

POST

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/message**

Request Body

This operation does not submit any parameters, but the HTTP request must include the WAV file to use as a voice-mail message as an attachment.

Response Body

This operation returns the following parameters:

- **messageName**: (String) An identifier for the uploaded file created by combining the messages's **createtime** and **onenumber** values with this syntax: *createtime_onenumber*.

Examples

Example 8–87 Request

```
POST https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/message
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: multipart/form-data; boundary="----=_Part_0_20761102.1296024147640"
MIME-Version: 1.0
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
Content-Length: 231837
------=_Part_0_20761102.1296024147640
Content-Type: audio/x-wav; name=test.wav
Content-Transfer-Encoding: binary
Content-Disposition: form-data; name="message1.wav"; filename="name.wav"
(binary data)
```

Example 8–88 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0)
[{"messageName":"test_321"}]
```

Get a List of the Public Social Voicemail Messages in a Mailbox

Retrieves a list of the publicly-accessible SVC Social Voicemail messages in a subscriber's mailbox.

Authorization

This operation does not perform authorization checking.

HTTP Method

GET

URI

/svc/vmsubscriber/*Subscriber_Name***/onenumber/***One_Number***/vmbox/messages/public**

Request Body

Empty

Response Body

On success the response includes an **HTTP 200 OK** message, and an array of message names and file locations of each voice-mail message file with an **isPublic** parameter value of **true**.

Examples

Example 8–89 Request

```
GET
https://10.182.98.78:9001/vmsubscriber/alice/onenumber/5556789/vmbox/messages/publ
ic HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.182.98.78:9001
```

Example 8–90 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43
Server: Jetty(8.0.1.0
[{"messageName":"fridaymessage","fileLocation":"/21/32/3231/friday.wav"},{"message
Name":"0100102063000_123124","fileLocation":"/21/32/3231/20100102063000_
123124.wav"}]
```