

-----

Overview

-----

The OpenScape Voice Server CSTA-SDK provides a set of API's to help simplify the development of Java based desktop applications which want to control the OpenScape Voice Server Call Control capabilities through the use of CSTA.

It provides simpler API's to connect and manage the CSTA Features, hiding some of the complexity of the CSTA Object model.

The OSV Server CSTA-SDK connects directly to the OpenScape Voice server via raw TCP connection. HTTP/SOAP connection support is added since V1.2.0. TLS connection support is added since V1.3.0.

-----

Release Notes

-----

v1.5.0

~~~~~

What's New?

CQ00332244: FRN8616 - CSTA SDK enhanced to support FRN8556 Do Not Disturb with Snooze Option.  
CQ00332245: FRN8613 - CSTA SDK enhanced to support FRN8181 Call Log Services and Events.

Interface Changes

The following new interfaces are introduced to the CstaProvider class to support the above new functionality.

CstaGetConfigurationData

CstaGetConfigurationDataResponse CstaGetConfigurationData(String ons, String cellDeviceGNF,  
String cellCarrier, String deviceModel,  
String deviceOS)

Support for the GetConfigurationData in order to get the appropriate data, providing the device with ONS service and OSMO device. Also, the new "snoozeDND" is supported as part of the supportedOSMOfeatureList parameters received in the GetConfigurationData Response.

The following new interfaces are introduced to the CstaLogicalDevice class to support the above new functionality.

SetDoNotDisturb

boolean SetDoNotDisturb(boolean doNotDisturbOn, int DNDResume)

This is a new SetDoNotDisturb interface in order to enable the DND Snooze functionality for the device. DND service will remain active for the period in minutes specified (DNDResume).

GetDoNotDisturbData

CstaDoNotDisturbData GetDoNotDisturbData()

Get the current status of the Do Not Disturb feature for the device (true if DND is on and active, false if DND is off) and the time in minutes remaining till the DND is deactivated when the DND snooze functionality has been enabled.

The following new interfaces are introduced to the CstaDevice class to support the above new functionality.

CallLogSnapshot

CstaCallLogResponseWithErrorCode CallLogSnapshot()

Given a specified CSTA monitored device, a Call Log Snapshot is requested. The request success is validated with the arrival of a Call Log Snapshot Response. Every modification of the device's Call Log will result in receiving a Call Log Event message with the updated Call Log.

In case of an error during the processing of Call Log Snapshot, a CSTA Internal Service Error Code is returned.

CallLogDelete

```
CstaCallLogResponseWithErrorCode CallLogDelete(CstaCallLogDeleteList recordList)
```

Given a specified CSTA monitored device and a list of call records, a deletion of these records is requested. The request success is validated with the arrival of a Call Log Delete Response. A Call Log Event with the updated Call Log is received as a result.

In case of an error during the processing of Call Log Delete, a CSTA Internal Service Error Code is returned.

v1.4.0

~~~~~

#### What's New?

-----  
CQ00288521: Support for GetSwitchingFunctionDevices and GetLogicalDeviceInformation.

CQ00283448: Support Application Session.

CQ00285193: Support for event filtering in Monitor Start Requests.

CQ00286729: Support for event filtering in Monitor Start Responses.

CQ00283593: Allow multiple CSTA links through the CstaSystem class.

CQ00285201: Faster method to detect link failure/closed sockets through the event listener interface.

#### Bug Fixes

-----  
CQ00283347: Fix invokeID overflow when invokeID reaches 9998 (max value set at 9999).

CQ00285928: Changes to stabilize the read and send message functions in high volume traffic situations.

CQ00286047: Prevent out of memory and overload conditions in volume traffic situations by limiting the CstaEventsQueue size.

#### Interface Changes

-----

The following new interfaces are introduced to the CstaProvider class to support the above new functionality.

```
CstaGetSwitchingFunctionDevices
```

-----

```
boolean CstaGetSwitchingFunctionDevices(CstaDeviceCategory category)
boolean CstaGetSwitchingFunctionDevices(CstaDeviceID deviceId, CstaDeviceCategory category)
```

If only the category is specified, then a list of all devices of that category are returned in a (potentially multiple) CstaSwitchingFuctionDevices event(s).

If a deviceId is specified, and that deviceId is an MLHG Pilot, a list of agents of that pilot are returned in a (potentially multiple) CstaSwitchingFuctionDevices event(s).

If a deviceId is specified, and that deviceId is an MLHG agent, a list of pilots that agent belongs to are returned in a (potentially multiple) CstaSwitchingFuctionDevices event(s).

```
CstaGetLogicalDeviceInformation
```

-----

```
CstaGetLogicalDeviceInformationResponse CstaGetLogicalDeviceInformation(CstaDeviceID deviceId)
CstaGetLogicalDeviceInformationResponse CstaGetLogicalDeviceInformationNID(CstaDeviceID
networkInterfaceDeviceId)
```

Note: The information returned is defined in the CSTA interface spec; note that if the deviceId is a keyset, then the appearanceList data contains appearances for each related keyset line. If the deviceId is an MLHG pilot, then the associatedGroupList contains deviceIds for each member agent.

#### Monitor Event Filtering

-----

```
CstaMonitor MonitorStart(CstaDevice device, boolean silmTag, boolean NID, CstaFilterEventList filter)
```

The full interface for MonitorStart is extended to allow a CstaFilterEventList filter to be applied. A CstaFilterEventList can be created and populated with the events that you do NOT wish to see. These events will not be sent for the subscriber monitored under this monitor point.

#### Application Session

-----

The following interfaces are provided for Application Session support:

```
CstaApplicationSession StartApplicationSession(int sessionDuration)
CstaApplicationSession StartApplicationSession(String applicationID, String protocolVersion,
                                              int sessionDuration)
CstaApplicationSession StartApplicationSession(String applicationID, String protocolVersion,
                                              int sessionDuration, int heartBeatTimer)
CstaApplicationSession getApplicationSession()
int ResetApplicationSessionTimer(int requestedSessionDuration)
int ResetApplicationSessionTimer(int requestedSessionDuration, int heartBeatTimer)
```

```

void StopApplicationSession(String definedEndReason, String appEndReason)

v1.3.2
~~~~~
What's New?
-----
CQ00275056: Enhancement to support appCallbackID and appCallbackName
in the privateData for Make Call requests

v1.3.1
~~~~~
What's New?
-----
CQ00242438: FRN4844 - SIP URI support

CQ00240730: Support AssociatedCallingDevice and AssociatedCalledDevice

v1.3.0
~~~~~
What's New?
-----
CQ00233503: CSTA-SDK: Support TLS connection for FRN4181

Bug Fixes
-----
CQ00221692: Null pointer when handling a basic Held/Retrieved event
CQ00222319: XML Parse error for element "restricted" and "acdCall"

v1.2.0
~~~~~
What's New?
-----
CQ00214198: CSTA-SDK: Support HTTP/SOAP connection

CQ00157701: Support Message Waiting Indicator Physical Device service

CQ00170233: Add suport for FRN4464 Support for Xpressions SAP phone with
ASC active voice recording (Heidelberger Cement)

CQ00157718: Support Routing Service for Telefonica Step 2

CQ00157473: Support CompoundCallState in SnapshotDevice response

CQ00159062: CSTA-SDK: Support of services permitted

CQ00159066: CSTA-SDK: Provide local CstaConnection object in the CstaEventObject

    CQ00200594: FRN4208 CSTA-SDK support for Continuous Silent Monitoring for Voice only
without Verint, ASC, and OpenScape Contact Center

    CQ00193909: FRN4242 CSTA-SDK support for CSTA Application external device name provider
via CSTA Network Interface Device (NID) service

    CQ00213840: FRN5670: Rebrand OSV - CSTA-SDK.

Bug Fixes
-----
CQ00158981: autoOriginate option in CSTA MakeCall and JoinCall request
should be a enum type instead of String

v1.1.0
~~~~~
What's New?
-----
CQ00145625: Updated Javadoc to include a brief description for
CstaException including all the CSTA Requests.

CQ00146946: Improved DeviceID / DeviceIdentifier String parsing.

CQ00146986: Improved ForwardingEvent to support optional tags such as
"forwardTo", "ringCount", and "ringDuration".

CQ00147181: Allow multiple MonitorStarts after a failed attempt.

CQ00157462: Added Application interface access to object "cstaEventObject"
and "CstaEventType". Allowing easy access to the type of
the currentObject and the ability to caste the object.
Example:
if(cstaEventObject.evtType == CstaEventType.DELIVERED)
{
    CstaDeliveredEvent deliveredXml = (CstaDeliveredEvent) cstaEventObject.currentXmlObject;
    CstaDeviceIdentifier deviceIdentifier = new CstaDeviceIdentifier(deliveredXml.getCalledDevice());
}

```

CQ00157464: Allows Application to be notified if there is a network failure to the heart beat. Next heart beat will fail and a CstaEvent with type="CstaException" and indicator="HeartbeatFailure" will be sent to the EventListener.

CQ00157469 Added ArrayList<CstaConnection> to the CstaCall object, which returns a snapshot object.

Note:

Application can only use this snapshot CstaCall object to check CallIDs, and is not able to send a request such as connection.HoldCall().

CQ00157704: Added Custom XML support for CSTA SDK Browser

CQ00157741: Allows incoming MonitorStop requests to remove SDK monitor objects, related connections, and to notify the application.

CQ00158008: Added Application interface enabling the application to specify an external inbound name in the AcceptCall.

Example:

```
CstaConnection.AcceptCallWithNameProvider(string)
```

CQ00158976: Extra handling for "restricted" tag as one of the CstaDevice elements. Thereby setting the device's FQDN to "restricted" if it is "restricted" in message.

Bug Fixes

-----

CQ00147199: Fixed NullPointerException issues when calling method  
CstaProvider.registerEventListener().

v1.0.0

~~~~~

Implemented basic functionality to the CSTA-SDK.

=====  
All rights reserved.  
=====