MITEL

OPEN INTEGRATION GATEWAY (OIG)

DEVELOPER GUIDE
RELEASE 1.0

Open Integration Gateway Developer Guide
Release 1.0
February 2013

Introduction

About this Guide

This Developer Guide describes the Mitel® Open Integration Gateway (Mitel OIG). This guide helps software developers create applications that access a Mitel MCD system (cluster or single 3300) through the Mitel OIG.

In general, all developers are required to join the Mitel Solutions Alliance (MSA) at one of the Developer Advanced membership levels, in order to develop or modify OIG-based applications.

However, developers also have the option to evaluate the OIG development capabilities in advance of joining MSA, either by ordering the OIG 60-Day Trial (for Mitel end-customers – P/N 54005933), or by requesting a 30-day reservation on the Mitel OIG Hosted Virtual Lab (for prospective commercial developers – see http://www.mitel.com/partners/partner-programs/mitel-solutions-alliance/join-now/ for more information).

About the Mitel OIG

The Mitel OIG is a web server that provides a single access point to services available within a Mitel system. The OIG runs on the Mitel Standard Linux (MSL) operating system. It can be deployed as an MSL software blade through the Mitel AMC licensing server, or as a virtual appliance downloaded from the Mitel OnLine web site.

The Mitel OIG supports third-party call control capabilities and offers a full suite of functionality from simple call control to contact center monitoring and control.

The first release of the Mitel OIG offers MCD Call Control Service only. Future releases will offer more types of services.

The MCD call control functionality is offered in two types:

- Standard Call Control Service
- Advanced Call Control Service

Mitel OIG Release 1.0 offers Standard Call Control Service to allow an application to monitor and control the telephony activity of Mitel physical and logical devices including IP phones and Personal Ring Groups. Future releases of the Mitel OIG will offer Advanced Call Control Service that includes contact center monitoring and control.

Mitel OIG Call Control Service

The Mitel OIG Call Control web service is used to monitor and control devices programmed or configured in Mitel MCDs including IP/DNI Phones, Personal Ring Groups, and line appearances on multi-line phones. Applications open monitors on devices and then receive events about device state changes. For example, if an application opens a monitor on an IP Phone, the application will receive events when the IP Phone receives a call or makes a call. Opening a monitor also means the application is able to control the device. For example, an application can initiate a call on a monitored phone in place of a user manually initiating the call. Phone feature monitoring is also supported for phones. Applications can open monitors on phone features and receive events about phone feature state changes (e.g., phone Do-Not-Disturb (DND) status).

The Mitel OIG Call Control web service uses a device model; not a user model or call model. An application receives call events, feature events, ACD events and system events. To allow an application to track a call in an MCD system cluster, the Mitel OIG provides a Global Call ID in call events.
When using Mitel OIG Release 1.0, an application needs to provide the IP address of each MCD. The OIG must connect to an MCD to control and monitor the physical and logical devices configured in that MCD. An application cannot connect to MCD A to monitor and control a device configured in MCD B.

Note: MCD, or Mitel Communications Director, is the Mitel call control software that has, historically, been deployed on the Mitel 3300 ICP. MCD software can now also be deployed on industry-standard servers, or in a virtual environment. The MCD software also executes in off–the-shelf hardware. An off-the-shelf hardware configuration is called MiCD or MCD ISS. This guide uses 3300 ICP, MCD, MiCD, MCD ISS in the context of the OIG.

Developer Resources

Product Documentation

You can access product documentation on the Mitel Customer Documentation website (eDocs) at http://edocs.mitel.com. You require a Mitel OnLine account username and password to view and download technical documentation.

To access the Product Documentation website:

1. Log in to Mitel OnLine at http://www.mitel.com by clicking **Login**.
2. Point to **Support** and then click **Product Documentation**.

OIG Documentation

The following OIG documentation is available on the Mitel Customer Documentation website:

- OIG 1.0 Installation & Maintenance Guide
- OIG 1.0 Engineering Guidelines
- OIG 1.0 Developer Guide (this guide)

Note: A Mitel OIG 1.0 Product Bulletin is also available on Mitel OnLine.

Virtual Appliance Deployment Guide

The Virtual Appliance Deployment Guide provides engineering guidelines for deploying Mitel virtual appliances and applications in a VMware virtual infrastructure. The guide is available on the Mitel Customer Documentation website (under the Solutions Guides heading).

MSL Documentation

The MSL documentation is available on the Mitel Customer Documentation website (under the Applications heading).

MCD Documentation

The MCD documentation is included with the MCD product and is also available on the Mitel Customer Documentation website.

MSA Documentation and Sample Code

To access the MSA member website:

1. Log in to Mitel OnLine at http://www.mitel.com by clicking **Login**.
2. Point to **MSA** and then click **MSA Downloads**.

Note: MSA Membership is required for access to the MSA web portal.

The following OIG materials are available on the MSA member website:

- OIG 1.0 Developer Guide (this guide)
- OIG 1.0 Release Notes
- Sample Code package for application development

Open Integration Gateway Training

The following training material is available from the Training portal on Mitel OnLine:

- OIG Primer
- Developer I&M Course
- Installer I&M Course

Obtaining Technical Assistance

MSA member support is available to all current members of the MSA program who are in good standing and have sufficient incident credits available. Members can contact MSA via the web, telephone or e-mail when they wish to report a problem or incident.

Email Support (preferred)

Object: To report a problem and supply full problem information to Product Support

Email: MSAsupport@mitel.com.

Hours: 8:30 A.M – 6:00 P.M. EST

Response Time: 3 business days

Details: Include the following details:

- Your TSID code in the subject line of your email.
- A very detailed description of the problem/incident.
- Attach any files/logs pertaining to the problem. Refer to the OIG 1.0 Installation & Maintenance Guide for more information regarding the types of logs needed for MSA support.

General Support

Inquiries or other difficulties that are not considered to be defect-related can be described in an electronic mail message to MSAInfo@mitel.com.

Member Support

The Mitel MSA web site is designed for self-service. It provides all members with access to API documentation, SDK upgrades, troubleshooting tips, sample configuration, and sample code. The MSA web site is located at http://www.mitel.com/msa. Click Login in the uppermost navigation bar.

Web Support

In the uppermost navigation bar of the MSA web pages, you have the option to click CONTACT US. Since all Internet users can use this contact mechanism, we expect that the range of questions will be quite broad. Questions may vary from general information about the MSA program to specific product inquiries that may or may not be aimed at MSA. In the case of registered members using the web CONTACT US mechanism, Web Support will follow the policies and guidelines specified under Email Support.

Providing Feedback

Comments or suggestions relating to this document can be provided in an electronic mail message to MSAInfo@mitel.com.

OIG Overview

The Mitel Open Integration Gateway (OIG) provides each application a single access point to web services. The first release offers MCD Call Control Service. Future releases will offer more types of services. The OIG provides a services-oriented architecture. OIG Release 1.0 offers access to the features and functionality offered by a Mitel MCD cluster (See Figure 1). An application opens a communication session with an OIG by logging in. Once the application is authenticated and authorized, the application uses one communication session to access all MCDs in the system cluster.

Note: The OIG can communicate with a single MCD or a cluster of MCDs. When there are two or more MCDs, the MCDs must be configured in a cluster. Open Integration Gateway Release 1.0 cannot communicate with more than one MCD cluster.

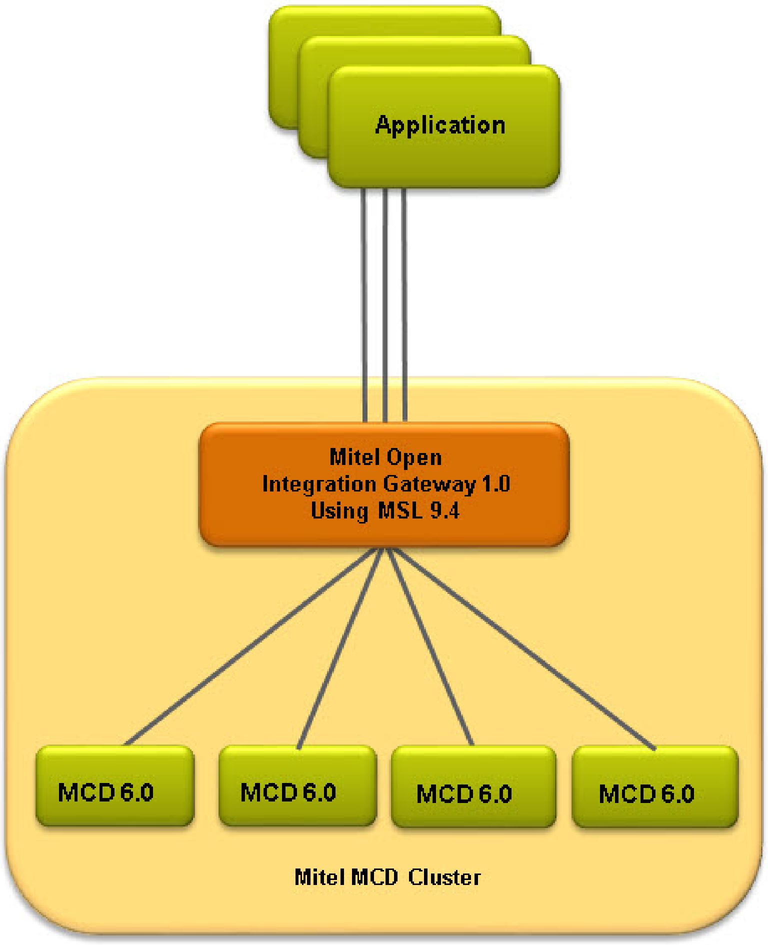


Figure 1: OIG System Configuration

Release 1.0 of the OIG provides an MCD Call Control web service that allows applications to monitor and control MCD IP phones (e.g., make call, answer call, end call, transfer a call, conference a call, forward a call, hold, retrieve, etc.). The Call Control web service allows an application to do what a user can do manually at an IP phone.

Note: MCD IP resiliency is not supported in this release. Refer to OIG Service Operations for a list of supported operations in this release of the OIG.

Future releases will provide additional Call Control web service operations to allow applications to monitor and control trunks, ACD paths, ACD agents, hunt groups, ring groups, conferences, etc. These additional Call Control Service operations support call center applications, call recording applications, messaging applications, etc.

Each OIG Call Control web service is defined by its own Web Service Description Language document (or WSDL) provided by Mitel. These XML documents describe the service, its commands (or operations), and its responses and events. When you import the WSDL file into your development environment, operation and parameter choices are provided automatically for each operation.

The web service is provided using SOAP and XML over HTTPS. Applications do not require software from Mitel to communicate with an OIG. An application does not need to integrate or compile in any Mitel code. Application developers are free to choose a programming language, a software development environment, an operating system, and a hardware platform. The web service model decouples the OIG software from the application; only the WSDL files are needed. Refer to Appendix E for a list of the WSDL-generated definitions for OIG operations and events.

OIG Communication Session Summary

Each application uses a session management web service (defined using WSDL) to communicate with the OIG. The session service describes how an application logs in, logs out, and resets a communication session keep alive timer. The application must reset a keep alive timer every 10 seconds or the OIG will terminate the communication session.

To log into an OIG, each application needs four parameters: 1) company name, 2) application name, 3) application password, and 4) local password. The first three parameters are defined by the application developer when registering their application with Mitel (see Application Registration for a detailed description of the application registration process). These first three parameters do not change in the application. The fourth parameter (local password) is site specific. An OIG administrator uses the OIG admin UI to create a local password specific to each application that uses a specific OIG. The local password controls the access of each individual application to each OIG. Each application that uses an OIG must provide a mechanism for a user to enter the local password and then the application must include the local password in the login operation along with the other three parameters mentioned above.

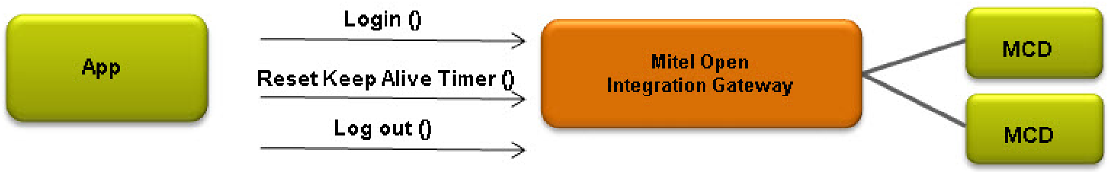


Figure 2: Session Management

OIG Call Control Service Summary

The OIG Call Control Service is defined using a command, response, and then event model. An application sends a command to activate a service operation and the OIG responds with success/ failure and return data. Operations trigger changes in the OIG and the Mitel MCD system. The changes are reported back to the application using events. An application can poll the OIG for events or provide an asynchronous reporting mechanism (register an event handler with the OIG so that the OIG can send events to the application - described later in the document).

The Call Control Service allows an application to control and monitor an IP Phone. In OIG 1.0, an application user must know information about the Mitel MCD system (i.e., MCD IP addresses, IP Phone prime line numbers, IP Phone line appearance button numbers, and hotdesk user numbers, etc.). The information needed is defined in this document.

Call Control Service Key Concepts

OIG Object IDs – The OIG allows an application to request an Object ID for each object of interest (i.e., MCD, IP Phone prime line, IP Phone line appearance button, hotdesk user). The OIG provides the Object ID to the application. The application stores the Object IDs and uses the Object IDs in commands and for processing events. When an application wants to monitor an object, the application initiates a monitor operation (command) with the related Object ID.

OIG Operations – The OIG offers web services. Each service offers an application a list of operations. Operations are commands to the OIG. For example, an application requests an Object ID for an MCD using the getIcpId operation. The OIG Call Control Service is divided into Standard and Advanced operations. The Advanced operations offer more complicated operations. To use the advanced operations requires a deeper knowledge of MCD behavior. A developer must specifically request the use of Standard or Advanced services when registering an application with Mitel (see OIG 1.0 Installation and Maintenance Guide for instructions on registering an application with Mitel). Only Standard Call Control Service is offered in OIG 1.0. Advanced Call Control Services will be offered in future OIG releases.

OIG Monitors – To control and monitor an object or device (e.g., IP Phone), the application must open a monitor (e.g., application initiates a monitor operation (command) with an IP Phone prime line Object ID). Once the application has created a monitor, the application can send commands and will receive events related to the monitored object (i.e., IP Phone prime line, IP Phone line appearance button, hotdesk user).

OIG Events – The OIG informs an application about status and state changes in objects (e.g., IP Phone prime line) using events. Each event has a name and associated event data.

Each event also has a class name that defines the event data to expect. The event data provides specific details about monitored objects. An application uses events to determine what commands to initiate next.

OIG Event Data – Each OIG event has data. The data provided depends on the event name and class. Events in the Call Event Class also have state and cause. The event state defines the state of the object being monitored. The event cause defines why the event was reported. All events contain data about the object being monitored. For example, events related to a monitor on an IP Phone Prime line contain the IP Phone number, IP Phone type, and name associated with the IP Phone number. The application needs to process the XML data provided in an event based on event name, class, and in some cases state and cause. An OIG event also has a time stamp. Time format is time_t: seconds elapsed since 00:00 hours, Jan 1, 1970 UTC. The time applied is from the OIG server and not an individual MCD.

Note: The Event Time in the Mitel OIG events is a 64-bit Integer (e.g., a MONITOR_SET event has an event time of 1349360583326). The time is a millisecond value that is an offset from the Epoch, January 1, 1970 00:00:00.000 GMT.

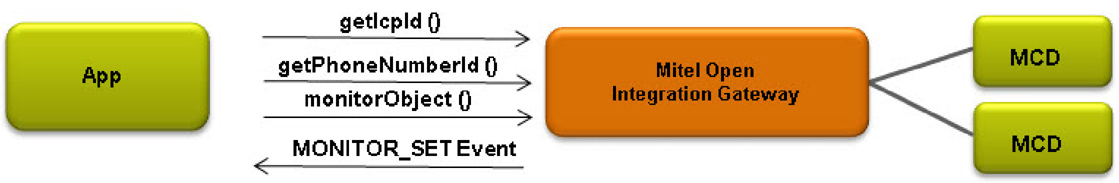


Figure 3: Create IP Phone Monitor

OIG Licensing

Licensing and delivery of the Mitel OIG is analogous to other Mitel software applications. That is, it is purchased by the Mitel customer (through their reseller); downloaded and licensed via the Mitel Licensing AMC portal; deployed on Mitel Standard Linux (MSL); and available in physical and virtual versions.

The OIG software requires the Mitel MSL operating system. The OIG software is deployed as a Mitel MSL blade through the Mitel AMC licensing server, or as a virtual appliance .ova file downloaded from the Mitel OnLine (MOL) Software Downloads web portal. Refer to the OIG Installation & Maintenance Guide for more information on licensing.

The Mitel OIG utilizes a new Mitel Certificate Server (MCS) and Access Control List (ACL) to ensure that only authorized developers and applications can connect to a customer MCD via the OIG, and make use of Mitel API Services. MSA administers the MCS and ACL. See Mitel Certificate Server for more information on MCS.

OIG System Configurations

A single application can communicate with one or more OIGs (a maximum of 6 is recommended).

The OIG can communicate with one or more applications (as many as 100).

An OIG can communicate with one or more MCDs (as many as 100). To monitor and control phones on a specific MCD, the application must provide the IP address of a specific MCD.

Up to six OIGs can communicate with one MCD.

The OIG and MCDs must be co-located in the same Enterprise LAN.

The OIG is not recommended for use in a DMZ or as a firewall to the WAN.

OIG Call Control Service Components

The OIG Call Control Service uses:

- a server component in each MCD.
- a client component in each OIG.

Note: Newer releases of server component software in an MCD will support older releases of the OIG client component software. However, only the functionality associated with the older OIG client component version will be available. When a newer version of the OIG client component software is communicating with an older version of MCD server component software, only the functionality associated with the older MCD server component version will be available.

OIG Call Control Service Device Support

The OIG supports control and monitoring of most Mitel IP phones with the exception of the following:

- Mitel IP Consoles and Attendant Consoles (no support)
- Mitel 500X, 520X, and 530X IP Phones (no support)
- IP trunks (no support)
- SIP devices have limited support and different function compared to IP Phones.
- ONS phones have limited support and different function compared to IP Phones.

Note: See Appendix D for SIP device support.

Mitel Certificate Server (MCS) Overview

Before using the OIG, an application developer (MSA member) must register their application(s) with the Mitel Certificate Server (MCS). The MCS is accessible through the internet. The OIG uses an Access Control List (ACL) from the MCS to identify approved applications. An application must be registered with the MCS and in the ACL before the application can communicate with the OIG.

The Mitel Certificate Server (MCS) also allows an application developer to request a Mitel certificate for their application. An application must have a Mitel certificate to use OIG Advanced Services. A Mitel certificate will be offered when using advanced service in future OIG releases.

Before an MSA-Authorized application can communicate with a deployed OIG the following is needed:

1. For “Standard” OIG communication, an MSA-Authorized User (MSA member) must register a company name, application name and application password with Mitel. This registration information is added to the Mitel Access Control List (ACL) in the MCS.
2. For “Advanced” OIG communication, an MSA-Authorized User (MSA member) must request a Mitel Certificate and register a company name, application name and application password with Mitel. This registration information is added to the ACL in the MCS. The Mitel Certificate is provided to the user by the MCS.

When requesting OIG access (steps 1 or 2 above), an MSA-Authorized User will login to the Mitel MSA-on-MoL web portal using their MSA-on-MoL account credentials. In this case the user must be an active MSA member and have an MSA account number (i.e., Mitel SAP number). New Mitel MSA-on-MoL web portal web links will direct the MSA-Authorized User to web forms provided by the MCS. See sections below.

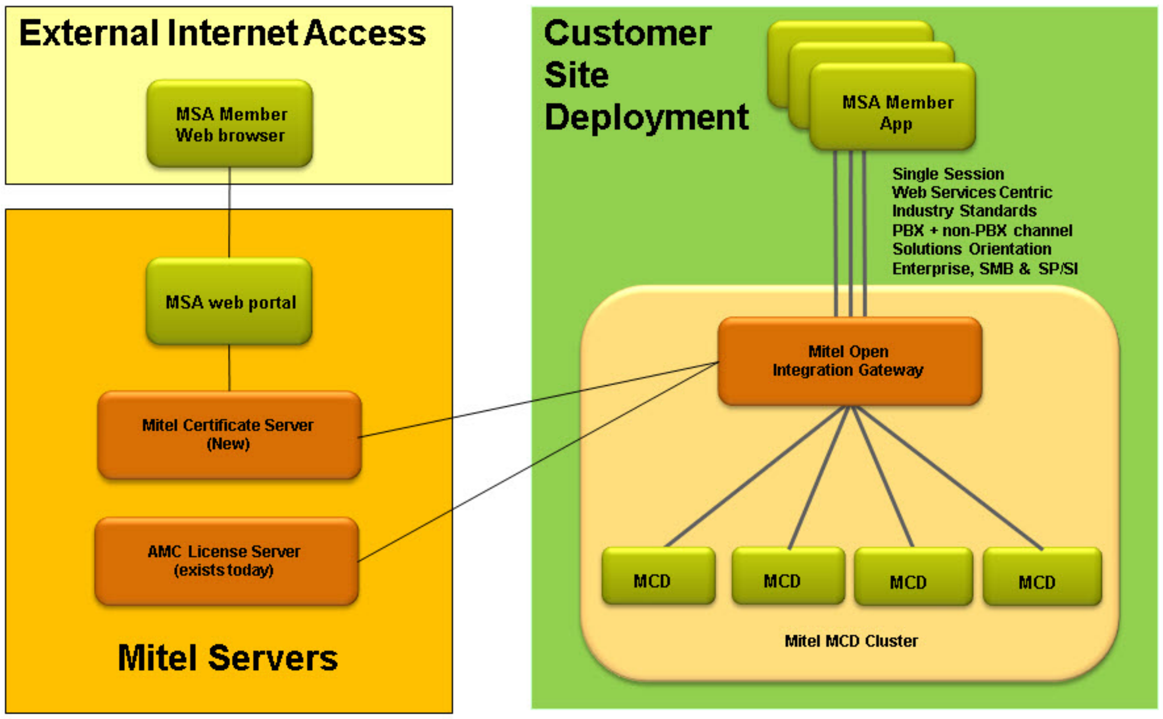


Figure 4: Accessing the Mitel Certificate Server

MCS General Requirements

1. The MCS can be accessed by MSA-Authorized Users (MSA members or prospective Virtual Lab Users). Only MSA members can request Mitel certificates or request application registration to the MCS ACL. Prospective MSA members can request a time slot on the Mitel Hosted Virtual Test Lab, after authorization by MSA.
2. MSA-Authorized Users login to the Mitel MSA-on-MoL web portal using their MSA-on-MoL account credentials (web browser interface). From an MSA web page, an MSA-Authorized User clicks a new web link to open a web page hosted by the Mitel Certificate Server. The following web browsers are supported: 1) Microsoft Windows Internet Explorer versions 8 & 9, 2) Google Chrome versions 16.0 & 17.0, 3) Firefox latest 2 versions.
3. If a Mitel Certificate request is approved, the MCS e-mails a Mitel Certificate and request approval to the MSA-Authorized User and registers the MSA-Authorized application in the ACL stored in the MCS.
4. If an application registration to ACL request is approved, the MCS e-mails the MSA-Authorized User notifying approval and registers the MSA-Authorized application in the ACL stored in the MCS.
5. The MSA Administrator has the ability to revoke an MSA-Authorized application. If an application is revoked, the MCS will indicate the application is revoked in the ACL. A revoked application is denied access to all OIGs. An e-mail will be sent to the application owner (MSA member) if an application is rejected, with text about contacting MSA for more information.

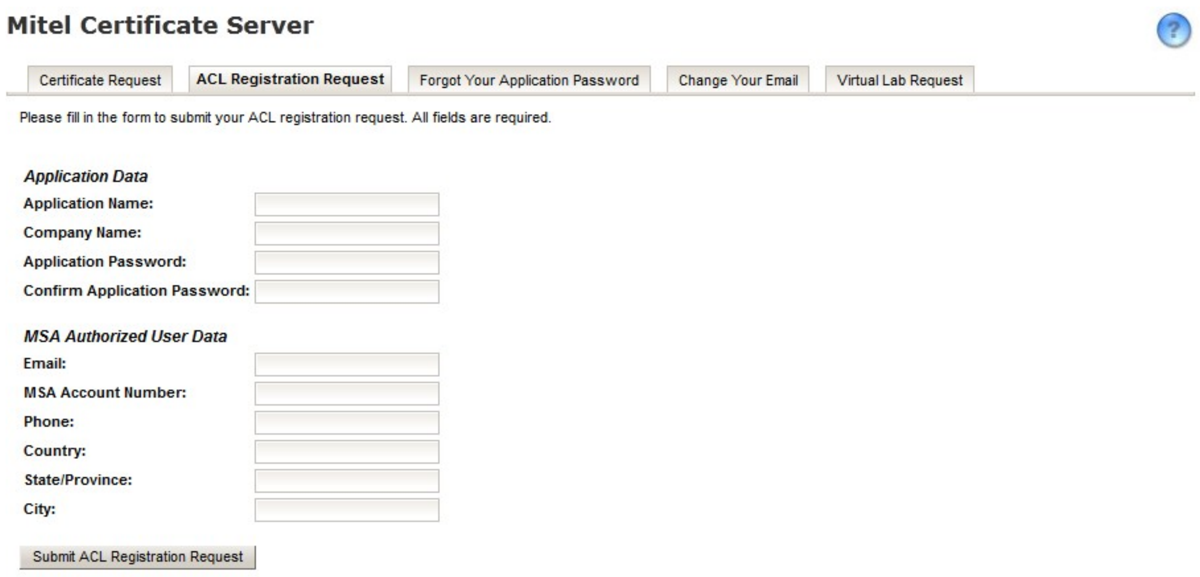
Mitel Certificate Server Access and Requests

This section describes what application developers need to request from the MCS to enable communication between their application(s) and an OIG. To use the standard web services offered by the OIG, a developer must register each of their applications with the MCS. To use advanced web services offered by the OIG, a developer must request a Mitel certificate for each of their applications. A Mitel certificate request includes application registration. See details in subsections below.

Application Registration in MCS Access Control List (ACL)

Application Registration Request

To access the MCS, an MSA-Authorized User (MSA Member) logs in through Mitel Online using an MSA-Authorized User account. The MSA-Authorized User is presented with the MSA web portal. The web portal provides web links and instructions on how to access the MCS. Eventually an MCS web form is presented with an “ACL Registration Request” tab. The MSA-Authorized User selects this tab. An ACL registration request form is presented to the MSA-Authorized User.



Mitel Certificate Server

Please fill in the form to submit your ACL registration request. All fields are required.

Application Data

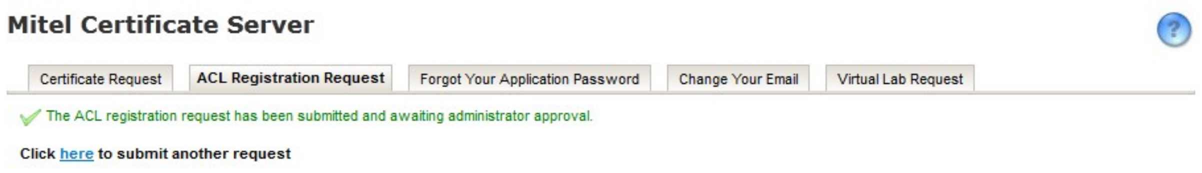
Application Name:	
Company Name:	
Application Password:	
Confirm Application Password:	

MSA Authorized User Data

Email:	
MSA Account Number:	
Phone:	
Country:	
State/Province:	
City:	

Figure 5: ACL Registration Request

The user must enter data into all the required fields and then click submit. The MCS validates the entered data. If the submitted data is valid, the MCS will redirect the MSA-Authorized User to a new web page displaying a message that states an ACL registration request has been submitted for MSA Administrator approval.



Mitel Certificate Server

The ACL registration request has been submitted and awaiting administrator approval.

Click here to submit another request

Figure 6: Successful ACL Registration Request

Application Registration Request Approval

Application registration approval is done by the MSA Administrator. The MSA Administrator logs in to the MCS and views a list of pending ACL registration requests. The MSA Administrator approves or denies each request based on the MSA-Authorized User information provided in the ACL registration request. Once approved, the MCS does the following:

1. Adds the user-provided company name, application name, MSA account number and application password to the MCS ACL. Once approved and added to the MCS ACL, these names and passwords cannot be changed.
2. E-mails the MSA-Authorized User notifying approval using the e-mail address entered by the MSA-Authorized User during the request.

Registration Request Denial

If the MSA Administrator denies the request, the MCS e-mails the MSA-Authorized User notifying denial with the reason using the e-mail address entered by the MSA-Authorized User. The denial e-mail includes text about contacting MSA for more information.

Revoking Applications

Revoking applications from the ACL is done by the MSA Administrator. After an MSA – Authorized application has been registered in the MCS ACL; an MSA Administrator can decide to revoke the application. Once an application has been revoked, the MCS does the following:

1. Updates the MCS ACL by tagging the application as revoked.
2. E-mails the MSA-Authorized User notifying of the revoke decision and the reason. The revoke e-mail includes text about contacting MSA for more information.

ACL Retrieval

The MCS ACL is retrieved by deployed OIG instances every four hours or upon manual OIG Administrator request. Deployed OIG instances request the ACL through a web service offered by the MCS. Deployed OIG instances use the ACL to deny or approve communication requests from applications.

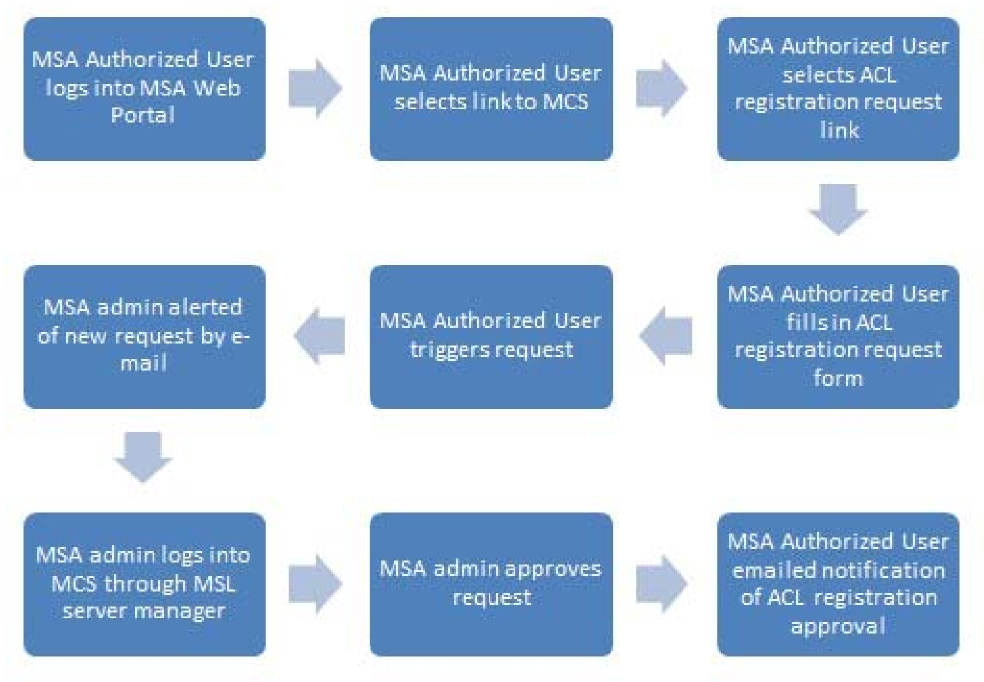

MSA Authorized User logs into MSA Web Portal
MSA Authorized User selects link to MCS
MSA Authorized User selects ACL registration request link
MSA Authorized User fills in ACL registration request form
MSA Authorized User triggers request
MSA admin alerted of new request by e-mail
MSA admin logs into MCS through MSL server manager
MSA admin approves request
MSA Authorized User emailed notification of ACL registration approval

Figure 7: ACL Registration Request and Approval Flowchart

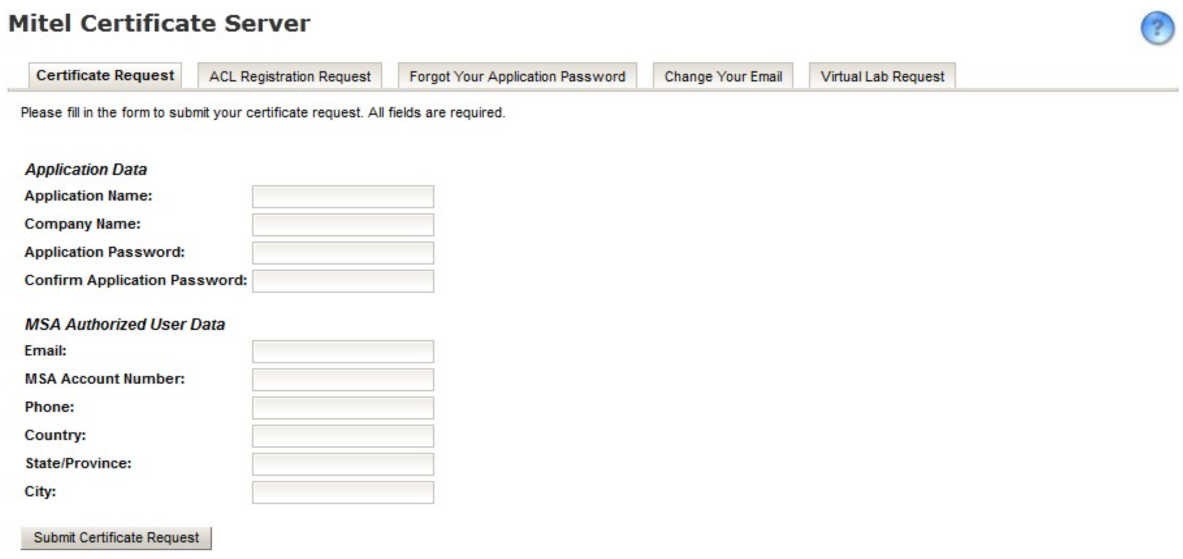
Mitel Certificate Request

MSA-Authorized Users (MSA Members) need to request a Mitel certificate when their applications require “Advanced” service operations from the OIG. A Mitel certificate request automatically includes an “add application to ACL” request. Thus a user does not need to submit two MCS requests to obtain access to an OIG.

Note: A Mitel certificate is not required in Release 1.0. The Mitel certificate process will be used in future OIG releases.

Certificate Request Process

To access the MCS, an MSA-Authorized User (MSA Member) logs in through Mitel Online using an MSA-Authorized User account. The MSA-Authorized User is presented with the MSA web portal. The web portal provides web links and instructions on how to access the MCS. Eventually an MCS web form is presented with a “Certificate Request” tab. The MSA-Authorized User selects this tab. A certificate request form is presented to the MSA-Authorized User.



Please fill in the form to submit your certificate request. All fields are required.

Application Data

Application Name:	
Company Name:	
Application Password:	
Confirm Application Password:	

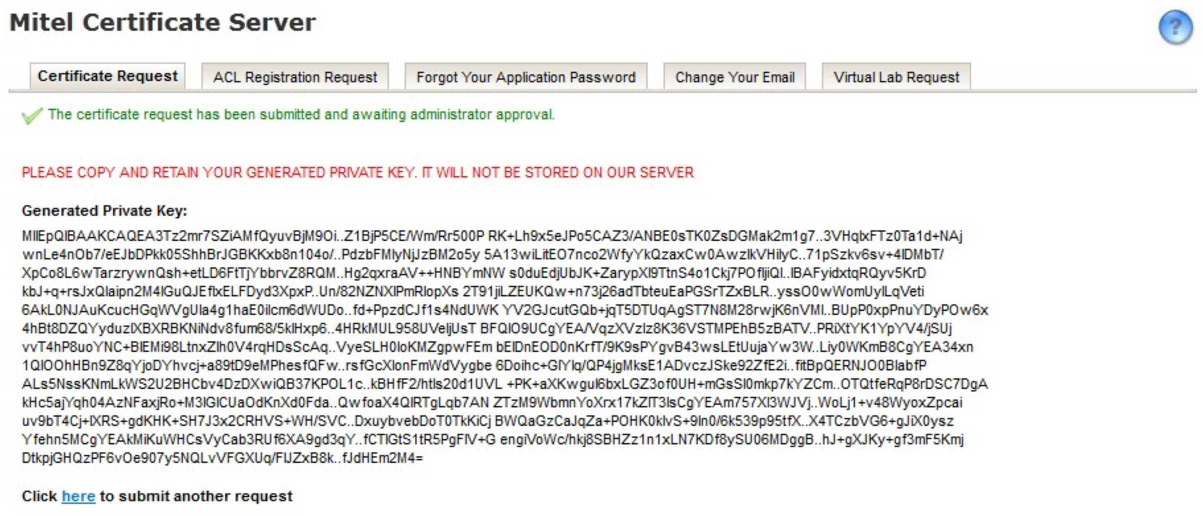
MSA Authorized User Data

Email:	
MSA Account Number:	
Phone:	
Country:	
State/Province:	
City:	

Figure 8: Certificate Request

The MSA-Authorized User must enter data in all the required fields on the MCS web form, then click submit. If the submitted data is valid, the MCS will redirect the MSA-Authorized User to another new web form displaying the following:

1. A message that states a certificate request has been submitted and is awaiting MSA Administrator approval.
2. A message that states the displayed MSA-Authorized User private key has to be copied before leaving this form. The MSA-Authorized User private key displayed is destroyed when the web form is closed (i.e., it is lost if not copied). If the private key is lost, then the pending certificate will be invalid (an application must have the matching private key for a Mitel certificate). If the private key is lost, then another certificate request will need to be submitted.
3. The private key for the requested Mitel certificate.



The certificate request has been submitted and awaiting administrator approval.

PLEASE COPY AND RETAIN YOUR GENERATED PRIVATE KEY. IT WILL NOT BE STORED ON OUR SERVER

Generated Private Key:

```
MIIEpQIBAAKCAQEA3Tz2mr7SZiAMfQyuvBjM9Oi..Z1BjP5CE/Wm/Rr500P RK+Lh9x5eJPo5CAZ3/ANBE0sTK0ZsDGMak2m1g7..3VHqlxFTz0Ta1d+NAj
wnLe4nOb7/eEJbDPkk05ShhBrJGBKKxb8n104o/..PdzbFMlyNjJzBM2o5y 5A13wiLitEO7nco2WfyYkQzaxCw0AwzlkVHilyC..71pSzkv6sv+4lDMbT/
XpCo8L6wTarzrywnQsh+etLD6FtTjYbbrvZ8RQM..Hg2qxraAV++HNBYmNW s0duEdjUbJK+ZarypXl9TtnS4o1Ckj7POfljiQl..lBAFyidxtqRQyv5KrD
kbJ+q+rsJxQlaipn2M4lGuQJEflxELFDyd3XpxP..Un/82NZNXlPmRlopXs 2T91jiLZEUKQw+n73j26adTbteuEaPGSrTZxBLR..yssO0wWomUylLqVeti
6AkL0NJAuKcucHGqWVgUla4g1haE0ilcm6dWUDo..fd+PpzdCJf1s4NdUWK YV2GJcutGQb+jqT5DTUqAgST7N8M28rwjK6nVMl..BUpP0xpPnuYDyPOw6x
4hBt8DZQYyduzlXBXRBKNiNdv8fum68/5klHxp6..4HRkMUL958UVeljUsT BFQlO9UCgYEA/VqzXVzlz8K36VSTMPEhB5zBATV..PRiXtYK1YpYV4/jSUj
vvT4hP8uoYNC+BlEMi98LtnxZlh0V4rqHDsScAq..VyeSLH0loKMZgpwFEm bElDnEOD0nKrfT/9K9sPYgvB43wsLEtUujaYw3W..Liy0WKmB8CgYEA34xn
1QlOOhHBn9Z8qYjoDYhvcj+a89tD9eMPhesfQFw..rsfGcXlonFmWdVygbe 6Doihc+GlYlq/QP4jgMksE1ADvczJSke92ZfE2i..fitBpQERNJO0BlabfP
ALs5NssKNmLkWS2U2BHCbv4DzDXwiQB37KPOL1c..kBHfF2/htls20d1UVL +PK+aXKwgul6bxLGZ3of0UH+mGsSl0mkp7kYZCm..OTQtfeRqP8rDSC7DgA
kHc5ajYqh04AzNFaxjRo+M3lGlCUaOdKnXd0Fda..QwfoaX4QlRTgLqb7AN ZTzM9WbmnYoXrx17kZlT3lsCgYEAm757Xl3WJVj..WoLj1+v48WyoxZpcai
uv9bT4Cj+lXRS+gdKHK+SH7J3x2CRHVS+WH/SVC..DxuybvebDoT0TkKiCj BWQaGzCaJqZa+POHK0klvS+9ln0/6k539p95tfX..X4TCzbVG6+gJiX0ysz
Yfehn5MCgYEAkMiKuWHCsVyCab3RUf6XA9gd3qY..fCTlGtS1tR5PgFlV+G engiVoWc/hkj8SBHZz1n1xLN7KDf8ySU06MDggB..hJ+gXJKy+gf3mF5Kmj
DtkpjGHQzPF6vOe907y5NQLvVFGXUq/FlJZxB8k..fJdHEm2M4=
```

Click here to submit another request

Figure 9: Successful Certificate Request Submission

The MSA-Authorized User must copy the private key and secure it in a safe location. The private key needs to be added to the application (i.e., the application named in the Mitel certificate request). The application will be required to use the private key to encrypt data as part of OIG login. The MCS does not store a copy of the generated private key. After copying the private key, the MSA-Authorized User can exit the web browser session.

If the certificate request is approved, the MCS e-mails the Mitel certificate to the MSA-Authorized User.

Certificate Request Approval

Mitel certificate request approval can be accomplished manually by the MSA Administrator. The MSA Administrator logs in to the MCS to view a list of pending certificate requests. The MSA Administrator approves or denies each request based on the information entered by the MSA-Authorized User. Once approved, the MCS does the following:

1. Generates a Mitel certificate which includes the MSA-Authorized User public key that matches the provided private key. The certificate expiry date is set to 25 years.
2. Digitally signs the Mitel certificate using a Mitel private key.
3. E-mails the MSA-Authorized User the Mitel certificate. The e-mail address was entered by the MSA-Authorized User during the certificate request.
4. Adds the MSA-Authorized User application to the MCS ACL.

Certificate Request Denial

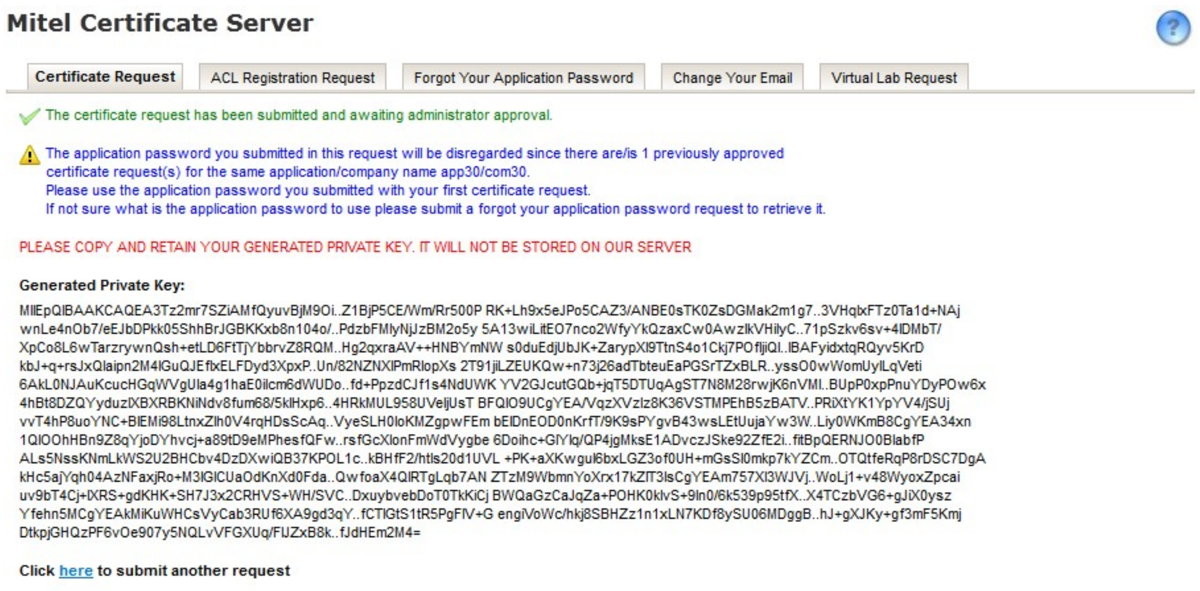
The MSA Administrator can deny a certificate request when reviewing the list of pending certificate requests. The MSA Administrator decides based on MSA-Authorized User information. Once denied, the MCS e-mails the MSA-Authorized User the denial and reason. The user e-mail address was provided by the MSA-Authorized User. The denial e-mail includes text about contacting MSA for more information.

Regenerating a Certificate for an Application

If an MSA member loses an application private key, the user must request another certificate. This additional request is different than the initial certificate request. The application registration in the ACL does not change, only a new certificate is generated. The MSA member logs in through Mitel Online using an MSA-Authorized User account. The MSA-Authorized User is presented with the MSA web portal. The web portal provides web links to a “Certificate Request” tab (see Figure 8). The MSA-Authorized User selects this tab. The MSA-Authorized User must enter data in all the required fields on the MCS web form, then click submit. If the submitted data is valid, the MCS will redirect the MSA-Authorized User to another web form displaying the following:

Note: The data entered must be the exact same information that was entered during the application registration process. This means that the Company Name, Application Name, and application password must be identical to the initial application registration.

1. A message that states a certificate regeneration request has been submitted and is awaiting MSA Administrator approval. This means another certificate is being requested but the MCS ACL will remain the same (i.e., Company Name, Application Name, and application password do not change). If the user enters another application password, the password will be ignored. The existing application password in the ACL must be used by the application when logging in to an OIG (when the initial Company name, Application name, and application password are approved and added to the MCS ACL - the names and password CANNOT be changed).
2. A message that states the displayed MSA-Authorized User private key has to be copied before leaving this form. The MSA-Authorized User private key displayed will be lost if not copied. If the private key is lost, then the pending certificate will be invalid and another certificate request will need to be submitted.
3. The private key associated with the additional Mitel certificate request.



The certificate request has been submitted and awaiting administrator approval.

The application password you submitted in this request will be disregarded since there are/is 1 previously approved certificate request(s) for the same application/company name app30/com30.
Please use the application password you submitted with your first certificate request.
If not sure what is the application password to use please submit a forgot your application password request to retrieve it.

PLEASE COPY AND RETAIN YOUR GENERATED PRIVATE KEY. IT WILL NOT BE STORED ON OUR SERVER

Generated Private Key:

MIlEpQIBAAKCAQEA3Tz2mr7SZiAMfQyuvBjM9Oi..Z1BjP5CE/Wm/Rr500P RK+Lh9x5eJPo5CAZ3/ANBE0sTK0ZsDGMak2m1g7..3VHqlxFTz0Ta1d+NAj
wnLe4nOb7/eEJbDPkk05ShhBrJGBKKxb8n104o/..PdzbFMlyNjJzBM2o5y 5A13wiLitEO7nco2WfyYkQzaxCw0AwzlkVHilyC..71pSzkv6sv+4lDMbT/
XpCo8L6wTarzrywnQsh+etLD6FtTjYbbrvZ8RQM..Hg2qxraAV++HNBYmNW s0duEdjUbJK+ZarypXl9TtnS4o1Ckj7POfljiQI..IBAFyidxtqRQyv5KrD
kbJ+q+rsJxQlaipn2M4lGuQJEflxELFDyd3XpxP..Un/82NZNXlPmRlopXs 2T91jiLZEUKQw+n73j26adTbteuEaPGSrTZxBLR..yssO0wWomUylLqVeti
6AkL0NJAuKcucHGqWVgUla4g1haE0ilcm6dWUDo..fd+PpzdCJf1s4NdUWK YV2GJcutGQb+jqT5DTUqAgST7N8M28rwjK6nVMl..BUpP0xpPnuYDyPOw6x
4hBt8DZQYyduzlXBXRBKNiNdv8fum68/5klHxp6..4HRkMUL958UVeljUsT BFQIO9UCgYEA/VqzXVzlz8K36VSTMPEhB5zBATV..PRiXtYK1YpYV4/jSUj
vvT4hP8uoYNC+BlEMi98LtnxZlh0V4rqHDsScAq..VyeSLH0loKMZgpwFEm bElDnEOD0nKrfT/9K9sPYgvB43wsLEtUujaYw3W..Liy0WKmB8CgYEA34xn
1QlOOhHBn9Z8qYjoDYhvcj+a89tD9eMPhesfQFw..rsfGcXlonFmWdVygbe 6Doihc+GlYlq/QP4jgMksE1ADvczJSke92ZfE2i..fitBpQERNJO0BlabfP
ALs5NssKNmLkWS2U2BHCbv4DzDXwiQB37KPOL1c..kBHfF2/htls20d1UVL +PK+aXKwgul6bxLGZ3of0UH+mGsSl0mkp7kYZCm..OTQtfeRqP8rDSC7DgA
kHc5ajYqh04AzNFaxjRo+M3lGlCUaOdKnXd0Fda..QwfoaX4QlRTgLqb7AN ZTzM9WbmnYoXrx17kZlT3lsCgYEAm757Xl3WJVj..WoLj1+v48WyoxZpcai
uv9bT4Cj+lXRS+gdKHK+SH7J3x2CRHVS+WH/SVC..DxuybvebDoT0TkKiCj BWQaGzCaJqZa+POHK0klvS+9ln0/6k539p95tfX..X4TCzbVG6+gJiX0ysz
Yfehn5MCgYEAkMiKuWHCsVyCab3RUf6XA9gd3qY..fCTlGtS1tR5PgFlV+G engiVoWc/hkj8SBHZz1n1xLN7KDf8ySU06MDggB..hJ+gXJKy+gf3mF5Kmj
DtkpjGHQzPF6vOe907y5NQLvVFGXUq/FlJZxB8k..fJdHEm2M4=

Click here to submit another request

Figure 10: Successful Certificate Regeneration Request Submission

The MSA-Authorized User must copy the private key and secure it in a safe location. The private key needs to be added to the application (i.e., the application named in the Mitel certificate request). The application will be required to use the private key to encrypt data as part of OIG login. The MCS does not store a copy of the generated private key. After copying the private key, the MSA-Authorized User can exit the web browser session.

The MCS notifies (by e-mail) the MSA Administrator of the pending certificate request. If the MSA Administrator approves the certificate request, the MCS e-mails the Mitel certificate to the MSA-Authorized User.

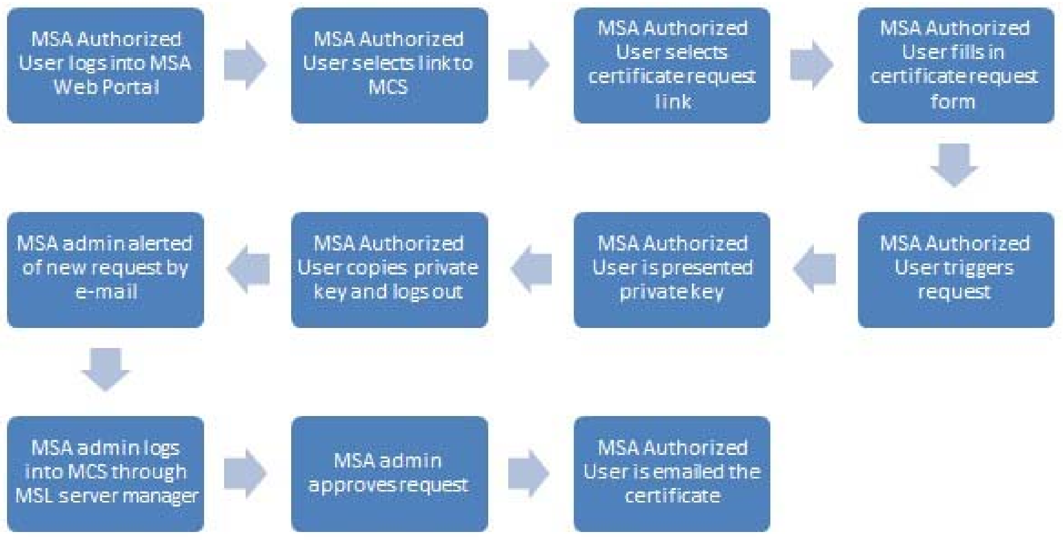

MSA Authorized User logs into MSA Web Portal
MSA Authorized User selects link to MCS
MSA Authorized User selects certificate request link
MSA Authorized User fills in certificate request form
MSA Authorized User triggers request
MSA Authorized User is presented private key
MSA Authorized User copies private key and logs out
MSA admin alerted of new request by e-mail
MSA admin logs into MCS through MSL server manager
MSA admin approves request
MSA Authorized User is emailed the certificate

Figure 11: Certificate Request and Approval Flowchart

Getting Started with OIG

Using the OIG is made easy with web services based on SOAP and XML. The OIG web services are defined using WSDL files. The internet provides all the material needed to learn about web servers, web services, WSDL, available toolkits, free development environments, tools, sample code, etc. This section will help get you started with how to develop applications for the OIG.

Recommended First Steps

The list below provides the recommended first steps to using the OIG:

- Read the OIG user documentation.
- Study the OIG online training.
- Use the Mitel hosted virtual test lab for the OIG.
- Install the OIG software.
- Use OIG sample applications to execute different MCD call control scenarios.
- Create your own application.

Mitel Hosted Virtual Test Lab

Mitel offers potential OIG applications developers access to a test lab. The test lab allows developers to use provided sample applications to investigate how the OIG works. Developers are also allowed to create their own applications and test them against the OIG. The test lab has a predefined, ready-to-use configuration for ease of use. See Developer Resources on page 2 for contacting MSA to get more information.

Install OIG

When an application developer is ready to start using the OIG, the developer needs to obtain the OIG software from Mitel. Refer to the OIG Installation & Maintenance Guide for instructions on how to license and install the software.

Mitel Sample Applications

To help a developer get started, Mitel offers several sample applications that work as is with the OIG. The sample applications are very easy to install and use. OIG sample applications are available on the MSA web portal (see Developer Resources on page 2). A help guide will be provided to explain how to use WSDL and how to implement synchronous and asynchronous communications with the OIG.

Creating Your Own Application

This section discusses OIG application creation. Application developers familiar with WSDL and web services will have little difficulty in creating an application. Developers are free to choose workstation hardware, operating system, software development environment and programming language. The OIG WSDL files allow software development environments (Visual Studio, NetBeans, Eclipse) to auto-generate much of the code needed to communicate with the OIG. The specific application behavior is left to the developer.

Mitel provides WSDL files that define the OIG Call Control Service operations, responses, events and data. Refer to Appendix E for a list of the WSDL-generated definitions for OIG operations and events.

The Mitel WSDL does not have an internet accessible target name space for XML schema definition. The developer has to write code to access the correct location of the schema definitions. More information on this topic is provided with the sample code.

WSDL in Visual Studio

If Microsoft Visual Studio is selected as the development environment and Visual Basic as the programming language, the following is an example of the simple steps needed to get started when creating an application.

- Start Microsoft Visual Studio 2010.
- Select File->New Project->Windows Forms Application.
- Enter the name (e.g., OIGSampleApp).
- Select “OK”.
- Select “Project->Add Service Reference” and press the “Advanced” button.
- Press the “Add Web Reference” button and enter the URL to the SessionService.wsdl file (e.g., http://10.40.224.130/axis2/services/SessionService?wsdl where 10.40.224.130 is the IP address of the OIG server) and press the green arrow button.
- Enter Web reference name: “SessionService” and press “Add Reference”.
- Repeat steps for the Standard Call Control Service, enter Web reference name “StdCCService”, and press “Add Reference”.
- Create an instance for the SessionService and StandardCCService in the Visual Basic form by selecting Form1.vb and selecting view code. Then enter the following code in the public class Form1:

```
‘Create an instance of the OIG SessionService Web Service to login/logout
    Dim OIGSessionSrv As New SessionService.SessionService
‘Create an instance of the OIG StandardCCService Web Service to access call
control operations
    Dim OIGcc As New StdCCService.StandardCCService
```

- Add code to create URLs to the OIG services:

```
OIGSessionSrv.Url = "https://" + OIGServerIpAddr.Text +
"/axis2/services/SessionService"
        OIGcc.Url = "https://" + OIGServerIpAddr.Text +
"/axis2/services/StandardCCService"
```

- Then add application interfaces (full source for a Visual Basic sample application is available).

WSDL in NetBeans (Sample Java Web Application)

If NetBeans is selected as the software development environment for Windows and Java as the programming language, the following sections provide the steps needed to create a Java application with a functional UI (all the sample code is available). This sample is more involved because of the user interface and asynchronous communications being provided.

Note: This example requires Windows 7, NetBeans, Apache Tomcat, Oracle Mojarra Java Server Faces (JSF, JBoss RichFaces) and MCD 6.0.

Introduction

This section provides an overview of how to create a Java web application in NetBeans. The developer creates an application from WSDL files. The application is developed with a Java Server Faces (JSF) UI. The UI allows a user to control the application from a web browser while the Java web application communicates with the Mitel OIG. The Mitel OIG communicates with a Mitel MCD 6.0.

Required System Setup

The following components are required: Windows 7 PC, Mitel OIG server, MCD 6.0. The Mitel OIG requires Internet access to the Mitel AMC server and Mitel MCS server. The Java web application will be deployed to Apache Tomcat in the Windows 7 PC.

The complete Java web application source code (with required jars) can be downloaded from the Mitel OnLine MSA web portal. Some software installation is needed (e.g., NetBeans, Apache Tomcat). Also, a Mitel OIG and Mitel MCD 6.0 need to be installed and configured. The Java Server Faces framework (JSF) is a request-driven MVC web framework based on a component-driven UI design model. JSF uses XML files called view templates (Facelets views). JSF easily combines complex GUIs into a single manageable component. This example uses JBoss RichFaces to allow easy integration of Ajax capabilities resulting in a better user experience.

Getting Started

- Download the Mitel sample code. The needed WSDL & XSD files are packaged with the provided sample source code. Copy the SessionService WSDL & XSD files and the StandardCCService WSDL & XSD files to C:\wsdl\SessionService and C:\wsdl\StandardCCService on the Windows 7 PC.
- Download Java Platform (JDK) JAVA SE – version 7 from the Internet. http://www.oracle.com/technetwork/java/javase/downloads/index.html/
- Download NetBeans 7.1 from the Internet. Select the NetBeans for Java EE version to allow web application creation and deployment directly through NetBeans onto a web server container (i.e., Apache Tomcat). http://netbeans.org/downloads/
- Download Apache Tomcat 7.0.27 (apache-tomcat-7.0.27.zip) from the Internet. http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.29/bin/
- Download JSF 2.1.12 (javax.faces.jar) from the Internet. Copy the jar to a C:\CallFeaturesPolling\jars directory on the Windows 7 PC. http://javaserverfaces.java.net/

Note: These files are provided with the Mitel Sample Code.

- Download RichFaces 4.2.2 from the Internet and install on the Windows 7 PC under the C: directory.
  http://www.jboss.org/richfaces/download/stable.html

  The needed RichFaces jars will be found under the following folders:

  C:\richfaces-4.2.2.Final\artifacts\framework

  C:\richfaces-4.2.2.Final\artifacts\ui

  Copy all the jars in these two directories to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC. The URL below provides a document that describe all the needed jars. http://docs.jboss.org/richfaces/latest_4_2_X/Developer_Guide/en-US/html/chap-Developer_Guide-Getting_started_with_RichFaces.html

  The required jars are:

  richfaces-core-api.jar

  richfaces-core-impl.jar

  richfaces-components-api.jar

  richfaces-components-ui.jar

Note: These files are provided with the Mitel Sample Code.

- Download the axis2-1.6.2 binary distribution ZIP from the Internet and install on the Windows 7 PC under the C: directory. The resulting C:\axis2-1.6.2\lib directory has all the axis2 jars needed. Copy all axis2 jars to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC. Also download the axis2.war file and copy the axis2 war file to the webapps directory of your Apache Tomcat (e.g., C:\Program Files (x86)\Apache Software Foundation\Apache Tomcat 7.0.27\webapps).
  http://axis.apache.org/axis2/java/core/download.cgi

Note: These files are provided with the Mitel Sample Code.

- Download the guava.jar from the Maven Repository and copy to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC.
  http://mvnrepository.com/artifact/com.google.guava/guava/10.0.1

Note: These files are provided with the Mitel Sample Code.

- Download the cssparser.jar from the Maven Repository and copy to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC.
  http://mvnrepository.com/artifact/net.sourceforge.cssparser/cssparser/0.9.5

Note: These files are provided with the Mitel Sample Code.

- Download the sac.jar from the Maven Repository and copy to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC.
  http://mvnrepository.com/artifact/org.w3c.css/sac/1.3
- Download all the following jars from the Internet and copy to the C:\CallFeaturesPolling\jars directory on the Windows 7 PC.

  Jars to download:

Note: All jar files required for the sample code are saved in the jars directory.

jsf-api-2.0.2.jar

jsf-facelets-1.1.15.jar

jsf-impl-patch-2.0.2.jar

slf4j-api-1.6.1.jar

validation-api-1.0.0.GA.jar

annotations-4.0.0.Final.jar

javax.inject.jar

javax.jms.jar

javax.servlet.jar

hibernate-validator-4.2.0.Final.jar

hibernate-validator-annotation-processor-4.2.0.Final.jar

Note: These files are provided with the Mitel Sample Code.

Setup NetBeans Environment with Apache Tomcat

Within NetBeans, in the left top pane, you will see three tabs (Projects, Files, Services). Go to Services, highlight Servers, right-click and select Add Server. Select Apache Tomcat and hit Next. Browse to the Apache Tomcat location and enter a username (admin) and password (password). Then click Finish. If successful, under Servers, you will see an entry for Apache Tomcat. Right-click this entry and hit Start to start the Apache Tomcat server. See start up messages output for the process in the lower right pane of NetBeans (Output Tab).

Setup JSF Web Application

Within NetBeans, in the left top pane, click on the Projects tab. Right-click and select New Project. In the dialog under Categories, select Java Web and then under Projects, select Web Application. Click Next. Choose a name for the project (e.g., CallFeaturesPolling) and a location to store it. Click Next. For the Server and Settings options, make sure that Apache Tomcat is selected as the server and the Java EE version is Java EE 6 web. The context path will be a slash followed by the project name (e.g., /CallFeaturesPolling), then click Next. For the Frameworks option, select Java Server Faces and then click Finish. If successful, you will see a new project created with all the required files and proper structure for a web application with a simple index.xhtml page saying Hello from Facelets. You can right-click on your project and hit Run. This will deploy the web application in Apache Tomcat and launch the browser with the following URL (http://localhost:8080/CallFeaturesPolling/). You should see Hello from Facelets in your browser.

Note: If you want to use the sample application provided by Mitel, simply open the sample application project CallFeaturePolling by selecting Open Project from the File menu.

Adding Required Jar Files to the Web Application Project

To add the required Jar files to our project, right-click the project and select Properties. In the left pane Categories, select Libraries. Then, in the right pane Compile, select and remove the JSF 2.1 and JSTL 1.1 already listed (need newer versions of JSF with RichFaces 4.2.2). Click on Add JAR/Folder and then browse to the jars directory you created on the Windows 7 PC and add all the downloaded jars, then click OK.

Note: Relative links to the required JAR files is configured within the Sample Application provided by Mitel.

Creating the CallFeaturesPolling Java Web Application

- Now generate java code stubs (SessionServiceStub.java, and StandardCCServiceStub.java) for the Mitel OIG web services using the Mitel WSDL files. Remember that you previously copied the SessionService WSDL & XSD files and the StandardCCService WSDL & XSD files to C:\wsdl\SessionService and C:\wsdl\StandardCCService on the Windows 7 PC. The WSDL & XSD files are provided with the Mitel sample application source code.
- Open a Windows console screen and change directory (cd) to c:\axis2-1.6.2\bin. Type the following command to create the SessionServiceStub.java file (wsdl2java -uri C:\wsdl\SessionService\SessionService.wsdl -s -g -p com.mitel.oig.sample -o .\ -wv 1.1 -d adb –sd).
- Type the following command to create the StandardCCServiceStub.java file (wsdl2java -uri C:\wsdl\StandardCCService\StandardCCService.wsdl -s -g -p com.mitel.oig.sample -o .\ -wv 1.1 -d adb –sd).
- The SessionServiceStub.java and StandardCCServiceStub.java files will be created under C:\axis2-1.6.2\bin\src\com\mitel\oig\sample.
  Copy the com directory under C:\axis2-1.6.2\bin\src (com\mitel\oig\sample contains SessionServiceStub.java and StandardCCServiceStub.java files) to the src\java directory under your NetBeans web application project folder (e.g., C:\Users\"user name is here"\Documents\NetBeansProjects\CallFeaturesPolling\src\java).
- Right-click on the project, select New and create one JSF managed bean with a class name of ClientBean.java in a package named com.mitel.oig.samplePolling.beans.
- Right-click on the project, select New and create one java class called ClientView.java in a package named com.mitel.oig.samplePolling.model.
- Modify the index.xhtml, ClientView.java and ClientBean.java files by copying code from the Mitel source code provided for CallFeaturesPolling (i.e., create your sample Java web application from the source code provided by Mitel).
- To load Mitel source code in NetBeans, select File-->Open Project, then browse to the Mitel Source code (for example, C:\CallFeaturesPolling), then select CallFeaturesPolling, then click the Open Project button. This should load the source code project into NetBeans. Click the CallFeaturesPolling project, then, right-click and select Run. The web application should display in a browser. To get the CallFeaturesPolling application to work, you need to first configure the OIG (refer to the OIG 1.0 Installation & Maintenance Guide for instructions on how to set up the OIG Server).
- To use the CallFeaturesPolling web application, please read the Readme file provided with the sample code.

MCD Configuration

The Open Integration Gateway communicates with a Mitel MCD system. Each MCD must have two Class of Service Options enabled, as defined in the table below. Refer to the OIG Installation & Maintenance Guide for additional OIG configuration information.

Note: If either of these options are disabled, then the Open Integration Gateway operations will not function correctly.

Class of Service Options	Each monitored device must have its Class of Service (COS) changed as indicated below: HCI/CTI/TAPI Call Control Allowed: **Yes** HCI/CTI/TAPI Monitor Allowed: **Yes** **Examples:** 1. Routing Devices use COS #1 only. If you are monitoring a routing device, COS #1 must be changed. 2. The COS for an ACD Path and an Agent Group is defined by the COS of the first agent in the prime agent group for the ACD path. If you are monitoring ACD paths and querying ACD groups, the COS of the first agent must be changed. 3. When monitoring ACD agents, the COS for each monitored agent must be changed. 4. When monitoring trunks, the COS for each monitored trunk must be changed. 5. When monitoring phones, the COS for each monitored phone must be changed.

Troubleshooting and Testing

After installing the OIG, see the OIG Tools section in this document and the OIG Installation and Maintenance Guide for troubleshooting suggestions and for instructions about using the provided Mitel OIG tools.

OIG Sample Scenarios

This section defines a few scenarios to introduce typical OIG behavior from an application viewpoint. See operation descriptions within this document and OIG WSDL for complete details.

Create Phone Monitor

In this scenario an application uses Session Service to open a communication with the OIG. Next the application opens a connection to an MCD and requests asynchronous event reporting. Then the application opens and closes a monitor on a phone. Finally the application closes the communication session with the OIG by logging out. Events associated with the operations are not shown.

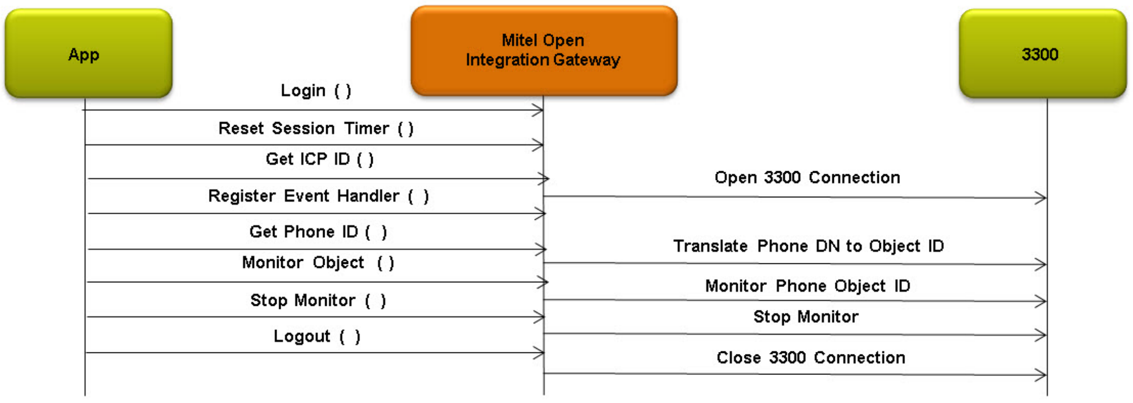


Figure 12: Create Phone Monitor

Basic Call Monitoring

In this scenario the application has already created a communication session with the OIG. The application requests an Object ID for a phone and then opens a monitor. Once a monitor set event is received, the application calls a phone and then hangs up. In this scenario the called phone answered.

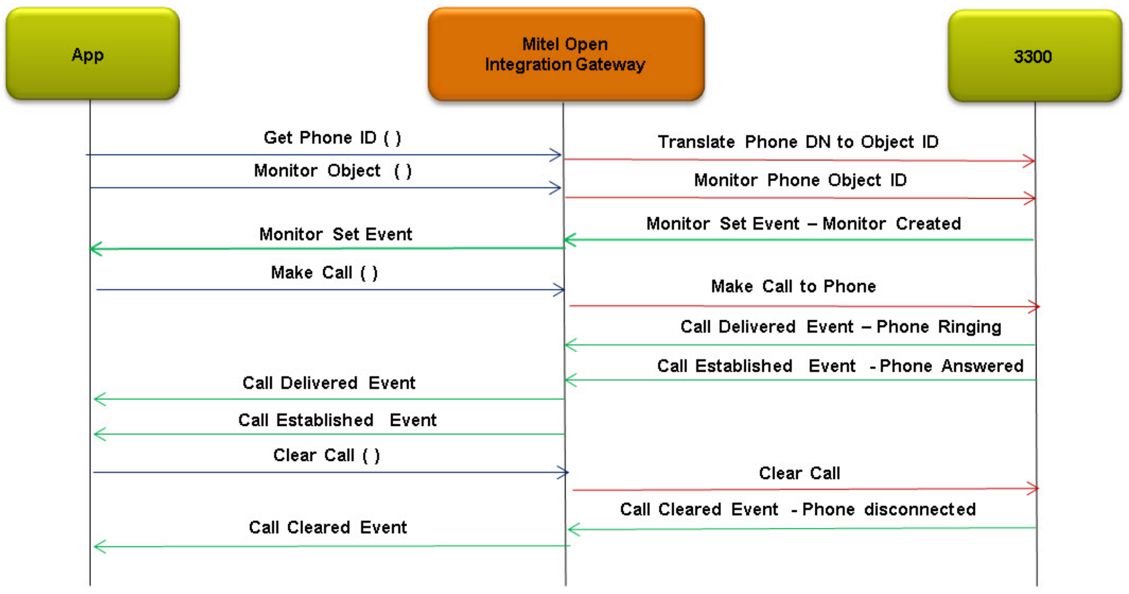

App
Mitel Open Integration Gateway
3300
Get Phone ID ( )
Translate Phone DN to Object ID
Monitor Object ( )
Monitor Phone Object ID
Monitor Set Event
Monitor Set Event – Monitor Created
Make Call ( )
Make Call to Phone
Call Delivered Event – Phone Ringing
Call Established Event - Phone Answered
Call Delivered Event
Call Established Event
Clear Call ( )
Clear Call
Call Cleared Event - Phone disconnected
Call Cleared Event

Figure 13: Basic Call Monitoring

OIG Service Operations

This section describes the operations that an application uses to communicate with the OIG. The operations are grouped into the following services:

- Session – This service is used by the application to open a communication session with the OIG.
- Call Control – This service is used by the application to control and monitor MCD objects (e.g., IP phones). The Call Control Service is provided in two types: Standard and Advanced. The Advanced type will be offered in future OIG releases. The Standard type provides operations to control and monitor phone objects (e.g., phone prime line DN, phone line appearance button number, hotdesk user DN, personal ring group DN). Call Control Service Standard provides operations that model what a user can manually initiate at a phone. The Advanced type provides operations to control and monitor MCD trunks, hunt groups, ACD paths, ACD agents, conferences, etc.

Session Service

The Session service allows an application to open a communication session with an OIG.

App to OIG Session	Service Type
login (localPassword, companyName, applicationName, applicationPassword) returns result (true/false), sessionId if true, errorDescription if false	Standard
login (localPassword, companyName, applicationName, applicationPassword, certificate) returns result (true/false), errorDescription if false, authenticationData if true, sessionId if true **Note:** A Mitel certificate must be provided to access OIG Advanced Services	Advanced
authenticate (signedAuthenicatedData, sessionId) returns result (true/false), errorDescription if false, sessionId if true **Note:** If false App is denied access to Adv. services	Advanced
logout (sessionId) returns result (true/false), errorDescription if false **Note:** If application does not close all monitors before logging out, the OIG ensures monitors are properly closed.	Both
resetSessionTimer (sessionId) returns result (true/false), errorDescription if false	Both
serviceVersions (sessionId) returns result (true/false), errorDescription if false, sessionId if true, serviceVersions if true	Both

Session Service – Standard Type

login

Definition

login (localPassword, companyName, applicationName, applicationPassword)

Description

This operation creates a communication session between the application and the OIG.

Attributes

Attribute	Description
localPassword	Each application requires a local password defined in the OIG. The application must allow entry of the local password at runtime. The OIG administrator of a specific OIG creates a local password for each application. The application local password is unique to each OIG installation.
companyName	Company Name is known to the developer at application creation. Company Name is NOT provided to the application at runtime (it is hard coded). Company Name is provided to Mitel as part of Application Registration.
applicationName	Application Name is known to the developer at application creation. Application Name is NOT provided to the application at runtime (it is hard coded). Application Name is provided to Mitel as part of Application Registration.
applicationPassword	Application password is known to the developer at application creation. Application password is NOT provided to the application at runtime (it is hard coded). Application password is provided to Mitel as part of Application Registration.

Returns

- result – true or false
- errorDescription – if result false
- sessionId – if result true

Notes:

- sessionId identifies the communication session between the Application and the OIG.

logout

Definition

logout (sessionId)

Description

This operation terminates a communication session between the application and the OIG.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.

Returns

- Result – true or false
- errorDescription – if result false

Notes:

- sessionId identifies the communication session between the Application and the OIG.

resetSessionTimer

Definition

resetSessionTimer (sessionId)

Description

This operation resets the keep alive timer used to detect stale communication sessions. If the session timer is not reset, the OIG closes the communication session. If session times out, the application must login again and get a new sessionId.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- sessionId identifies the communication session between the Application and the OIG.

serviceVersions

Definition

serviceVersions (sessionId)

Description

This operation obtains the software version of the OIG, the version of each service within the OIG and the version of any connected MCDs.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.

Returns

- result – true or false
- errorDescription – if result false
- serviceVersions – if result true the software versions for OIG, each service within the OIG and any connected MCDs

Notes:

- sessionId identifies the session between the Application and the OIG.

Session Service – Advanced Type

login

Definition

login (localPassword, companyName, applicationName, applicationPassword, certificate)

Description

This operation creates a session between the application and the OIG for advanced services.

Attributes

Attribute	Description
localPassword	Each application requires a local password defined in the OIG. The application must allow entry of the local password at runtime. The OIG administrator of a specific OIG creates a local password for each application. The application local password is unique to each OIG installation.
companyName	Company Name is known to the developer at application creation. Company Name is NOT provided to the application at runtime (it is hard coded). Company Name is provided to Mitel as part of Application Registration.

Attribute	Description
applicationName	Application Name is known to the developer at application creation. Application Name is NOT provided to the application at runtime (it is hard coded). Application Name is provided to Mitel as part of Application Registration.
applicationPassword	Application password is known to the developer at application creation. Application password is NOT provided to the application at runtime (it is hard coded). Application password is provided to Mitel as part of Application Registration.
certificate	Each application that requires advanced services from an OIG requires a Mitel Certificate. Application developers request a Mitel certificate using the Mitel Online MSA web portal. The same certificate is used in all instances of an application.

Returns

- result – true or false
- errorDescription – if result false
- authenticateData – if result true
- sessionId - if result true

Notes:

1. sessionId identifies the session between the Application and the OIG.
2. authenticateData needs to be sent back to the OIG to authenticate the session. The application must sign the data before sending to the OIG. See authenticate operation below.

authenticate

Definition

authenticate (signedAuthenticatedData, sessionId)

Description

This operation confirms that the application that provided the Mitel certificate has the associated private key.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
signedAuthenticatedData	The application must sign the authenicateData provided by the OIG with the application's private key. The application private key was provided by Mitel as part of the Mitel certificate process.

Returns

- result – true or false
- errorDescription – if result false

- sessionId – if result true

Notes:

1. sessionId identifies the session between the Application and the OIG.
2. If authenticate fails in the OIG, application access is denied.

Call Control Service

The table below provides a summary of the OIG Call Control Service operations for the Standard type.

•alternateCall	This operation is is used to alternate between an active call and a call on consultation hold. The active call is placed on consultation hold and the call on consultation hold is made active.
•answerCall	This operation answers an incoming call that has been offered to the monitored device.
•camponCall	This operation allows the caller to camp onto a busy device. The busy device may then trade to the camped on call, or the busy device will ring due to this camp on as soon as the device goes idle.
•cancelConsCall	This operation clears the active call at the specified device and reconnects it to the party on Consultation Hold.
•clearCall	This operation releases the specified call from the monitored device.
•clearCallMeBackMsg	This operation allows a caller to clear a message at an HCI routing device identified by deviceDn. This operation is only supported on an MCD HCI routing device. To create an HCI Routing Device use the MCD Configuration UI and select “Users and Devices”, Group Programming, Hunt Groups, and then create a device with type HCIRoute.
•conferenceCall	This operation merges two calls into a single conference call.
•consultationCall	This operation puts the active call on consultation hold and places a call to the provided number.
•getCallStatus	This operation allows the application to request the call state of a monitored device.
•getDeviceConfiguration	This operation gets device configuration information about the monitored device. This includes the line apperance button numbers.
•getDeviceFeatures	This operation gets the feature settings for a phone DN or a hot desk user DN. If the objectId of a hotdesk user DN is provided, the registrationDn will provide the DN of the phone where the hotdesk user is logged in and hotDeskUserLoggedInDn will be blank. If the objectId of a phone DN is provided, then the registrationDn will be blank and hotDeskUserLoggedInDn will be provided.

•getEvent	This operation is used by the application to synchronously call for events held in the OIG. This operation is only used if the application polls for events. The registerEventHandler operation described above must be used when the application wants to receive events from the OIG asynchronously. The operation returns immediately if an event is pending.
•getHotdeskUserDn	This operation gets hotdesk user DN based on phone DN Object ID.
•getHotdeskUserLoginDevice	This operation gets device DN based on hotdesk user DN Object ID.
•getIcpId	This operation provides the object ID for a MCD. The application uses this MCD object ID in other operations to specifically identify an MCD to the OIG. The application calls this operation to open a connection to an MCD. Note: An error condition occurs when an invalid MCD IP address is given.
•getLineAppearanceId	This operation provides the object ID for a MCD phone line appearance. The application uses this line appearance object ID in other operations to specifically identify a MCD phone line appearance to the OIG. An application must call this operation before setting a monitor on a phone line appearance. The application must provide a phone button number because a phone can have more than one line appearance with the same DN.
•getPhoneNumberId	This operation provides the object ID for a MCD phone prime line. The application uses this phone number object ID in other operations to specifically identify a MCD phone prime line DN to the OIG. An application must call this operation before setting a monitor on the phone prime line.
•holdCall	This operation temporarily suspends communication on the active call at the specified device. This operation places a call on hard hold.
•loginExtHotDeskUser	This operation logs in an External hotdesk user.
•loginHotDeskUser	This operation logins in a hotdesk user and places the device that is hotdesk enabled into out-of-service.
•logoutExtHotDeskUser	This operation logs out an External hotdesk user.
•logoutHotDeskUser	This operation logs out a hotdesk user and places the device that is hotdesk enabled into in-service.
•makeCall	This operation makes a call from this monitored device (identified by a monitor object ID) to the number (DN) provided. An application can also set a forced account code; see the notes for details.
•monitorFeatures	This operation creates a device feature monitor. Device feature include DND, call forwarding, hotdesk user login and logout. The provided object ID identifies the object to be monitored. This operation instructs the MCD to report feature events when a device feature changes. The object ID also identifies the MCD to use to set the feature monitor. If this operation is successful the application receives feature events for the monitored object.

•monitorObject	This operation creates a MCD monitor. The provided object ID identifies the object to be monitored. This operation generates a monitored set event that confirms the monitor has been set successfully on a MCD. The object ID also identifies the MCD to use to set the monitor. Once the monitor set event is received by the application, the application can start to control (call operations on) the monitored object.
•monitorPRGPresence	This operation establishes a monitor on the MCD that reports all presence activites of all Personal Ring Groups.
•newCall	This operation ends a consultation call without retrieving the held call, making it possible for the user to consult someone else (i.e., the monitored device is placed in dialing state). An Application must use makeCall to call another number or cancel the consultation to retrieve the held call.
•outPulseDigits	This operation sends DTMF digits from the monitored device to the other device while a call is in EstablishedState. Tones are not heard at the monitored device. An application can not use this operation to instruct one IP Phone to send DTMF digits to another IP Phone.
•pickupCall	This operation picks up a call which is ringing at another device. This operation cannot be used to pickup a call which has landed at a Routing Device; see the notes for details.
•redirectCall	This operation redirects (transfers) a ringing call to another device. This operation can also be used to redirect calls from an ACD queue or routing device to a specific phone.
•registerEventHandler	This operation registers an event callback with the OIG. The OIG uses the URL to send events asynchronously to the application. The application must process the https request from the OIG (accept the event) as fast as possible.Mitel provides sample code that demonstrates how an application must support asynchronous event reporting if choosing this approach.
•retrieveCall	This operation reconnects an existing held call at the specified holding device. The call was previously suspended using the holdCall or the MCD call hold feature. This operation retrieves a call from hard hold.The application must remember the local Call ID of the call that was placed on hold.
•sendCallMeBackMsgNoCall	This operation allows a caller to leave a message at a device identified by deviceDn.
•setAccountCode	This operation is used to assign a non-verified account code to an established call on any monitored phones. When attempting to use this operation with an invalid verified account code, this operation does not verify the account code but returns command success even though account code is not valid (i.e., MCD does not check the account code).

•setCallMeBack	This operation allows the caller to have the MCD ring the caller with a distinctive pattern as soon as the called party becomes available, and then ring the called party as soon as the caller answers this distinctive ring. This operation can be invoked when an attempt to make a call has failed due to the called party being busy. Application needs to provide objectId of device setting the call me back.
•setCallMeBackMsgForCall	This operation allows a caller to leave a message at the called phone indicating who called. The called party could have been busy on the phone at the time or unable to answer.
•setCFAlways	This operation sets the call forwarding always (also known as follow me) of the DN on or off.
•setCFBusyExternal	This operation sets the call forwarding for a DN on or off related to an external call. If the called Dn is busy when an external call is received, the call is forwarded to the specified destination.
•setCFBusyInternal	This operation sets the call forwarding for a DN on or off related to an internal call. If the called Dn is busy when an internal call is received, the call is forwarded to the specified destination.
•setCFNAExternal	This operation sets the call forwarding for a DN on or off related to an external call. If the called Dn does not answer when an external call is received, the call is forwarded to the specified destination.
•setCFNAInternal	This operation sets the call forwarding for a DN on or off related to an internal call. If the called Dn does not answer when an internal call is received, the call is forwarded to the specified destination.
•setDeviceDND	This operation sets or clears the device DND.
•setPRGPresence	This operation sets the presence of a PRG member.
•splitConferenceCall	This operation splits a conference call into two calls.
•stopFeatureMonitor	This operation stops a specific device feature monitor.
•stopMonitor	This operation stops a specific device monitor. If stopMonitor is called on a monitored device that also has a device feature monitor, this operation also stops the device feature monitor.
•stopPRGPresenceMonitor	This operation stops a monitor on the MCD that reports all presence activites of all Personal Ring Groups.
•tradeCall	This operation trades the camped-on party with the currently connected party.
•transferCall	This operation establishes a call between the party on Consultation Hold and the other party in the active call. The requesting device ends its participation in the call.
•verifyHotdeskUserPin	This operation verfies a PIN used to login in a hotdesk user.

Call Control Service Operations – Standard Type

The sections below define the OIG Call Control Service operations for type Standard. Also see the Appendices for a description of all services auto-gnerated from WSDL.

getIcpId

Definition

getIcpId (sessionId, IcpIpAddress)

Description

This operation provides the object ID for a MCD. The application uses this MCD object ID in other operations to specifically identify an MCD to the OIG. The application calls this operation to open a connection to an MCD.

An error condition occurs when an invalid MCD IP address is given.

Steps to reproduce error condition with invalid IP address:

1. Invoke GetIcpID() with invalid MCD IP Address (say 10.112.60.25 on which there is no MCD configured)
2. GetICPID() returns errorDescription error: "Failed to open connection to ICP at: 10.112.60.25"

An error condition also occurs when an MCD is in an unstable / usable state.

Steps to reproduce error condition when MCD not ready but connected on network:

1. Invoke GetIcpID()
2. GetICPID() returns errorDescription error: "Failed to open connection to ICP at: 10.112.60.25"

Application recovery suggestion is to wait 5 minutes before retry. Also check the MCD IP Address used by application.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
IcpIpAddress	The IP address of the MCD that the application wants to communicate with.

Returns

- result – true or false
- errorDescription – if result false
- IcpId - if result true
- connectedState – indicates if OIG is actually connected to the MCD.

Notes:

- IcpId identifies a specific MCD instance.

registerEventHandler

Definition

registerEventHandler (sessionId, eventHandlerURL)

Description

This operation registers an event callback with the OIG. The OIG uses the URL to send events asynchronously to the application. The application must process the https request from the OIG (accept the event) as fast as possible. The application should not process the event within the event handler logic. The event handler should retrieve the event and store event for later processing . Mitel provides sample code and supporting notes that demonstrates how an application must support asynchronous event reporting if choosing this approach.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
eventHandlerURL	The URL used by the OIG to send events back to the application in a async manner.

Returns

- result – true or false
- errorDescription – if result false.

Notes:

1. App registers event handler for async events (not needed for polling – see getEvent below).
2. App must provide the handleEvent operation when using async events so that OIG can call this web service to report events.

getPhoneNumberId

Definition

getPhoneNumberId (sessionId, IcpId, primeDn)

Description

This operation provides the object ID for a MCD phone prime line. The application uses this phone number object ID in other operations to specifically identify a MCD phone prime line DN to the OIG. An application must call this operation before setting a monitor on the phone prime line.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
IcpId	A specific MCD object ID provided by the OIG when the application called the getIcpId operation.
primeDn	The prime line number of the phone being monitored.

Returns

- result – true or false
- errorDescription – if result false
- objectId - if result true

Notes:

- Provides objectId for a phone prime line DN even when phone has line appearances..

getLineAppearanceId

Definition

getLineAppearanceId (sessionId, IcpId, primeDn, buttonNum)

Description

This operation provides the object ID for a MCD phone line appearance. The application uses this line appearance object ID in other operations to specifically identify a MCD phone line appearance to the OIG. An application must call this operation before setting a monitor on a phone line appearance. The application must provide a phone button number because a phone can have more than one line appearance with the same DN.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
IcpId	A specific MCD object ID provided by the OIG when the application called the getIcpId operation.
primeDn	The prime line number of the phone being monitored.
buttonNum	The button number of a physical button on a phone. The buttons are numbered from 1 (starting with the phone prime line DN button and moving upward and then to the left) to 16. Only the first 16 buttons / lines can be monitored including the prime line DN.

Returns

- result – true or false
- errorDescription – if result false
- objectId - if result true

Notes:

- Provides objectId for a line appearance on a phone using the physical phone button number. The OIG uses the button number as a group member number in requests to the MCD.

monitorObject

Definition

monitorObject (sessionId, objectId)

Description

This operation creates an MCD monitor. The provided object ID identifies the object to be monitored. This operation generates a monitored set event that confirms the monitor has been set successfully on a MCD. The object ID also identifies the MCD to use to set the monitor. Once the monitor set event is received by the application, the application can start to control (call operations on) the monitored object.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	A specific MCD object ID provided by the OIG when the application called a specific get object ID operation.

Returns

- result – true or false
- errorDescription – if result false
- objectId - if result true
- IcpId – if result true

Notes:

1. The IcpId is the object ID for the MCD that is controlling the monitored object.
2. When an application is setting a monitor on a phone, the application must wait for a monitor set event before sending any commands to the MCD using such a monitor on the phone.

monitorFeatures

Definition

monitorFeatures (sessionId, objectId)

Description

This operation creates a device feature monitor. Device feature include DND, call forwarding, hotdesk user login and logout. The provided object ID identifies the object to be monitored.

This operation instructs the MCD to report feature events when a device feature changes. The object ID also identifies the MCD to use to set the feature monitor. If this operation is successful the application receives feature events for the monitored object.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	A specific MCD object ID provided by the OIG when the application called a specific get object ID operation.

Returns

- result – true or false
- errorDescription – if result false
- objectId - if result true
- IcpId – if result true

Notes:

- The IcpId is the object ID for the MCD that is controlling the monitored object.

stopMonitor

Definition

stopMonitor (sessionId, objectId)

Description

This operation stops a specific device monitor. If stopMonitor is called on a monitored device that also has a device feature monitor, this operation also stops the device feature monitor.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	A specific MCD object ID provided by the OIG when the application called a specific get object ID operation.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application can no longer use this objectId.

stopFeatureMonitor

Definition

stopFeatureMonitor (sessionId, objectId)

Description

This operation stops a specific device feature monitor.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	A specific MCD object ID provided by the OIG when the application called a specific get object ID operation.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application can no longer use this objectId.

getEvent

Definition

getEvent (sessionId, timeout)

Description

This operation is used by the application to synchronously call for events held in the OIG. This operation is only used if the application polls for events. The registerEventHandler operation described above must be used when the application wants to receive events from the OIG asynchronously. The operation returns immediately if an event is pending. The application must implement the getEvent so that the application collects events as fast as possible. After retrievening the event from the OIG, the application should simply stored the event for later processing and immediately ask for another event. The OIG will provide many events to the application and this operation must execute as fast as possible.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
timeout	The period of time the application wants to block while waiting for an event to be returned by the OIG. When the time out period expires the operation returns. The operation returns immediately if an event is pending.

Returns

- result – true or false
- errorDescription – if result false
- standardEvent – if result true

Notes:

- See the events section of this document for the possible events that are returned (i.e., standardEvent is a generic XML doc with different event types). Only one event is returned at a time.

alternateCall

Definition

alternateCall (sessionId, localCallId, objectId)

Description

This operation is is used to alternate between an active call and a call on consultation hold. The active call is placed on consultation hold and the call on consultation hold is made active.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the active call that will be placed on consulation hold.
objectId	The monitor object ID for the phone line or hotdesk DN involved.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on a phone line or hotdesk DN (i.e., the object with the active call).

answerCall

Definition

answerCall (sessionId, localCallId, objectId)

Description

This operation answers an incoming call that has been offered to the monitored device.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID is provided in the call received event from the OIG. The local call ID represents the active call to be answered.
objectId	The monitor object ID for the device involved.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. The application must have a monitor on the device.
2. If the phone has an active call on another line, then that call is automatically placed on hold if the phone is configured for Auto Hold Class of Service.
3. Under some conditions if a localCallId of zero is provided in the answerCall operation, the answer will succeed. However, this approach is not recommended because many cases will fail. The answerCall operation requires a valid localCallId (i.e., the local call ID is provided in the call received event from the OIG. The local call ID represents the active call to be answered). If a zero call Id is used for answer call, call control will answer the active incoming call under some conditions. If an invalid local call id is used, the OIG will return invalid call id error.

camponCall

Definition

camponCall (sessionId, localCallId, objectId)

Description

This operation allows the caller to camp onto a busy device. The busy device may then trade to the camped on call, or the busy device will ring due to this camp on as soon as the device goes idle.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the call to be camped on.
objectId	The monitor object ID for the phone trying to camp on.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on the calling phone.

cancelConsCall

Definition

cancelConsCall (sessionId, localCallId, objectId)

Description

This operation clears the active call at the specified device and reconnects it to the party on Consultation Hold.

Attributes

Attibute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the call to be cancelled.
objectId	The monitor object ID for the phone doing the cancel.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on the phone doing the cancel.
- An example usage of cancelConsCall follows:
    - Party A is in conversation with party B.
    - Party A makes a Consultation Call to party C.
    - Party B is now on Consultation Hold.
    - Party C is busy OR rings and party C answers.
    - Party A wishes to return to party B.
    - Cancel Consultation Call clears the (active) call to party C and reconnects party B (on Consultation Hold) to party A.
    - Party A resumes the conversation with party B.

clearCall

Definition

clearCall (sessionId, localCallId, objectId)

Description

This operation releases the specified call from the monitored device.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. localCallId is the call to be dropped.
objectId	The monitor object ID for the phone clearing the call.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. The application must have a monitor on the phone.
2. If the specified device is in hands-free or headset mode, then the device is left in IdleState. If the device is off-hook, the user receives dial tone, or in some cases, silence.
3. Calls on hard hold cannot be cleared.
4. If the device is a participant in a conference, then the remaining members of the conference are usually not affected, depending upon the class of service and the MCD configuration.

conferenceCall

Definition

conferenceCall (sessionId, localCallId, objectId)

Description

This operation merges two calls into a single conference call.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the call to be conferenced.
objectId	The monitor object ID for the phone dong the conference.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on the phone.
- An example usage of conferenceCall follows:
    - “A” is talking to “B” and wishes to conference in “C”.
    - “A” makes a Consultation Call to “C” which puts “B” on Consultation Hold.
    - While talking to “C”, “A” invokes Conference Call.
    - “A”, “B”, and “C” are joined in a conference call.

consultationCall

Definition

consultationCall (sessionId, localCallId, objectId, number)

Description

This operation puts the active call on consultation hold and places a call to the provided number.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the call to be conferenced.
objectId	The monitor object ID for the phone dong the conference.
number	The other device number to be called.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on the phone.
- An example usage of consultationCall follows:
    - “A” is talking to “B” and wishes to call “C”.
    - “A” makes a Consultation Call to “C” which puts “B” on Consultation Hold.

holdCall

Definition

holdCall (sessionId, localCallId, objectId)

Description

This operation temporarily suspends communication on the active call at the specified device. This operation places a call on hard hold.

An example of event flow goes as follows:

1. Establish a two-party call between Phone1 and Phone2
2. Invoke CC Operation HoldCall() on Phone1
3. Verify Events:

   On Phone2 :

   - CallHeldEvent HeldState ConsHold
   - CallHeldEvent HeldState HardHold

   On Phone1 :

   - CallHeldEvent HeldState HardHoldInvoked

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	Local call ID was provided in the last call status event from the OIG. The local call ID represents the call to be held.
objectId	The monitor object ID for the phone involved.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. The application must have a monitor on the phone.
2. The held call can be subsequently retrieved using the retrieveCall operation.

makeCall

Definition

makeCall (sessionId, objectId, number, accountCode)

Description

This operation makes a call from this monitored device (identified by a monitor object ID) to the number (DN) provided. An application can also set a forced account code; see the notes for details.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone involved.
number	The number of the device to be called.
accountCode	A number terminated by #.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. The application must have a monitor on the phone.
2. The account code is used for call charging purposes. It is used in the Call Detail Record (CDR/SMDR) output by the MCD at the end of the call via the CDR (SMDR) port.
3. A forced non-verified account code may be specified with makeCall if required. The first makecall operation specifying the DN to call must be followed immediately by another makeCall using the number attribute to specify the forced non-verified account code. The forced account code must be terminated by #.
4. A forced verified account code may also be specified with makeCall, but the order of calls is reversed: the account code must come first.
5. Non-forced account codes with makeCall are not supported.

newCall

Definition

newCall (sessionId, localCallId, objectId)

Description

This operation ends a consultation call without retrieving the held call, making it possible for the user to consult someone else (i.e., the monitored device is placed in dialing state). An Application must use makeCall to call another number or cancel the consultation to retrieve the held call.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone involved.
localCallId	localCallId is the call to be dropped

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The application must have a monitor on the phone.

outPulseDigits

Definition

outPulseDigits (sessionId, dtmfDigits, objectId)

Description

This operation sends DTMF digits from the monitored device to the other device while a call is in EstablishedState. Tones are not heard at the monitored device. An application can not use this operation to instruct one IP Phone to send DTMF digits to another IP Phone. This operation is typically used to send digits out a PSTN trunk.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone involved.
dtmfDigits	Dtmf digits to be sent. The receiving end turns the digits into tones.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. The application must have a monitor on the phone.
2. The monitored device is required to be in EstablishedState.
3. Digit strings longer than 26 digits can be used with this service, provided that these strings are subdivided into (at most) 26 digit sub-strings using pause (~2s) markers or zero-pause (~0s) markers between the sub-strings.
4. An application can not use this operation to instruct one IP Phone device to send DTMF digits to another IP Phone Device

pickupCall

Definition

pickupCall (sessionId, ringDn, pickupDn, objectId)

Description

This operation picks up a call which is ringing at another device. This operation cannot be used to pickup a call which has landed at a Routing Device; see the notes for details.

The pickupCall operation requires the MCD to have a FAC for directed pickup or the pickupCall operation will fail with Invalid_Device. The user can confirm this by using an actual physical phone to do a call pickup. If the physical phone is able to pickup a call, the OIG operation should also work.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone involved.
ringDN	The directory number of the device which is ringing.
pickupDn	The directory number of the device which is going to pickup the call for the ringing phone.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. Use pickupCall if the remote call is in ReceivedState.

2. When no ring DN is provided, the operation attempts to pickup a call that is ringing in the phone's pickup group.
3. pickupCall is not supported for picking up a call from a Routing Device, since in that case, the call is actually queued rather than ringing. In this case, use redirectCall to deflect the call to the desired destination. The end result is the same.

monitorPRGPresence

Definition

monitorPRGPresence (sessionId, IcpId)

Description

This operation establishes a monitor on the MCD that reports all presence activites of all Personal Ring Groups.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
IcpId	The monitor object ID for the MCD controlling PRGs.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- Group Presence Monitor is not resilient.

stopPRGPresenceMonitor

Definition

stopPRGPresenceMonitor (sessionId, IcpId)

Description

This operation stops a monitor on the MCD that reports all presence activites of all Personal Ring Groups.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
IcpId	The monitor object ID for the MCD controlling the PRG.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- Application no longer interested in PRG presence information can stop the PRG Presence Monitor.

redirectCall

Definition

redirectCall (sessionId, redirectDn, localCallId, objectId))

Description

This operation redirects (transfers) a ringing call to another device. This operation can also be used to redirect calls from an ACD queue or routing device to a specific phone.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone involved.
localCallId	The identity of the call to be redirected.
redirectDn	The directory number of the device to which the call is to be redirected.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. Use redirectCall if the call is in ReceivedState or QueuedState.
2. Use transferCall if the call is in EstablishedState or HeldState.
3. If the call's new destination is busy redirectCall returns SXERR_FEATURE_NOT_ALLOWED.
4. If calls are made to the same phone at different line appearances, this operation always redirects the first call, no matter what localCallId is used.
5. Upon redirection of an ACD call, the MCD no longer regards (nor reports) that call as an ACD call. This is an inherent characteristic of MCD ACD packages.
6. Applications can: 1) redirect to a remote directory number from a ringing DN or from a redirect hunt group, and 2) redirect to an ARS string from an ACD Path. Applications cannot redirect from an ACD Path to a remote directory number.

retrieveCall

Definition

retrieveCall (sessionId, localCallId, objectId)

Description

This operation reconnects an existing held call at the specified holding device. The call was previously suspended using the holdCall or the MCD call hold feature. This operation retrieves a call from hard hold.The application must remember the local Call ID of the call that was placed on hold.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone with held call.
localCallId	The identity of the call to be retrieved.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- Retrieving a call that is on hold at some other device is similar to the MCD remote retrieve feature.

sendCallMeBackMsgForCall

Definition

sendCallMeBackMsgForCall (sessionId, localCallId, objectId)

Description

This operation allows a caller to leave a message at the called phone indicating who called. The called party could have been busy on the phone at the time or unable to answer.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone with held call.
localCallId	The identity of the call to be called device.

Returns

- result – true or false

- errorDescription – if result false

Notes:

- None.

sendCallMeBackMsgNoCall

Definition

sendCallMeBackMsgNoCall (sessionId, objIdOfDeviceToCall, deviceDn)

Description

This operation allows a caller to leave a message at a device identified by deviceDn.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objIdOfDeviceToCall	The monitor object ID for the phone to call back.
deviceDn	The identity of the phone to leave message.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

clearCallMeBackMsg

Definition

clearCallMeBackMsg (sessionId, objIdOfDeviceToCall, deviceDn)

Description

This operation allows a caller to clear a message at an HCI routing device identified by deviceDn. This operation is only supported on an MCD HCI routing device. To create an HCI Routing Device use the MCD Configuration UI and select “Users and Devices”, Group Programming, Hunt Groups, and then create a device with type HCIRoute.

Steps to reproduce an error condition on a normal device:

1. Use Application and log into the OIG and set monitors on Phone1 and Phone2
2. Make call to Phone1 from Phone3 so that Phone1 is busy.
3. Have Phone2 call to Phone1 so that call fails.
4. Invoke CC Operation SendCallMeBackMsgForCall() from Phone2

5. Verify messagewaitingLamp = true by invoking GetDeviceFeature() on Phone1
6. Invoke clearCallMeBackMsg CC Opeartion On phone1

Expected result on HCIRoute device:
clearCallMeBackMsg should be successful

Expected result on regular phone:
clearCallMeBackMsg fails with error: UNSUPPORTED_BY_PBX

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objIdOfDeviceToCall	The monitor object ID for the phone to call back.
deviceDn	The identity of the phone to leave message.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setAccountCode

Definition

setAccountCode (sessionId, localCallId, accCode, objectId)

Description

This operation is used to assign a non-verified account code to an established call on any monitored phones. When attempting to use this operation with an invalid verified account code, this operation does not verify the account code but returns command success even though account code is not valid (i.e., MCD does not check the account code).

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone with call.
accCode	The code to be used for charging this call.
localCallId	The identity of the call to which account code is assigned.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- The monitored phone must be in EstablishedState when this operation is called. An account code is a string of 2 to 12 alphanumeric characters. setAccountCode will return SXERR_INVALID_ATTRIBUTE_VALUE if this format is not correct.

setCallMeBack

Definition

setCallMeBack (sessionId, localCallId, objectId)

Description

This operation allows the caller to have the MCD ring the caller with a distinctive pattern as soon as the called party becomes available, and then ring the called party as soon as the caller answers this distinctive ring. This operation can be invoked when an attempt to make a call has failed due to the called party being busy. Application needs to provide objectId of device setting the call me back.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
localCallId	The identity of the call that is failing (e.g., called party is busy).
objectId	objectId of device setting the call me back on a busy phone.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

getCallStatus

Definition

getCallStatus (sessionId, objectId)

Description

This operation allows the application to request the call state of a monitored device.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone.

Returns

- result – true or false
- errorDescription – if result false
- callEventMsg – if true indicates if device is in a call or idle
- UNKNOWN – if event type is not known

Notes:

1. A Snapshot can be obtained on a line appearance.
2. A Snapshot can be obtained for members of Personal Ring Group.

splitConferenceCall()

Definition

splitConferenceCall (sessionId, objectId);

Description

This operation splits a conference call into two calls.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

tradeCall()

Definition

tradeCall (sessionId, objectId);

Description

This operation trades the camped-on party with the currently connected party.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- When tradeCall() has effected the trade so that the invoker is talking to the queued party, the other party becomes consultation-held rather than queued. Therefore, a second call to tradeCall will not work a second time; the application must invoke alternateCall instead.

transferCall()

Definition

transferCall (sessionId, objectId);

Description

This operation establishes a call between the party on Consultation Hold and the other party in the active call. The requesting device ends its participation in the call.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	The monitor object ID for the phone.

Returns

- result – true or false
- errorDescription – if result false

Notes:

1. A Supervised Transfer is where the requesting party is talking to the target party at the time this operation is called. The requesting party has made a consultation call and waited for the target party to answer.
2. An Unsupervised Transfer is where the requesting party has dialed the target party, using Consultation Call, but does not wait for the target party to answer. The target party can be ringing or busy. The transferred party (previously on hold) hears ringing or music while waiting for the target party to answer.

getDeviceConfiguration()

Definition

getDeviceConfiguration (sessionId, objectId, primeDn);

Description

This operation gets information about the monitored device.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
icpId	The MCD Object ID.
primeDn	The number of the device to get configuration.

Returns

- result – true or false
- errorDescription – if result false
- lineConfig – Includes the following: Device Type info, device number, objectId, all button numbers if any

Notes:

1. Only phone prime line DN allowed in this operation (i.e., no line appearance).
2. Hotdesk user DN supported.

loginHotdeskUser()

Definition

loginHotdeskUser (sessionId, pin, hotDeskDn, objectId);

Description

This operation logins in a hotdesk user and places the device that is hotdesk enabled into out-of-service.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
pin	The security PIN for the hostdesk login
hotDeskDn	The number of the hotdesk user.
objectId	Monitor object ID of the device to be hotdesk into.

Returns

- result – true or false
- errorDescription – if result false
- hotdeskDeviceDn – if result true
- hotdeskDeviceObjectId – if result true

Notes:

- The hotdesk enabled phone must be an internal phone.

logoutHotdeskUser()

Definition

logoutHotdeskUser (sessionId, objectId);

Description

This operation logs out a hotdesk user and places the device that is hotdesk enabled into in-service.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the hotdesk user.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

verifyHotdeskUserPin()

Definition

verifyHotdeskUserPin (sessionId, pin, hotDeskDn);

Description

This operation verfies a PIN used to login in a hotdesk user.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
pin	The security PIN for the hostdesk login
hotDeskDn	The number of the hotdesk user.

Returns

- result – true or false
- errorDescription – if result false
- hotdeskDeviceDn – if result true
- hotdeskDeviceObjectId – if result true

Notes:

- The hotdesk enabled phone must be an internal phone.

loginExtHotdeskUser()

Definition

loginExtHotdeskUser (sessionId, pin, objectId);

Description

This operation logins in an External hotdesk user.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
pin	The security PIN for the hostdesk login.
objectId	Monitor object ID of the device to be hotdesk into.

Returns

- result – true or false
- errorDescription – if result false
- extHotdeskDeviceDn – if result true
- extHotdeskDeviceObjectId – if result true

Notes:

- None.

logoutExtHotdeskUser()

Definition

logoutExtHotdeskUser (sessionId, objectId);

Description

This operation logs out an External hotdesk user.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the hotdesk user.

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setDeviceDND()

Definition

setDeviceDND (sessionId, objectId,dndState);

Description

This operation sets or clears the device DND.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the hotdesk user.
dndState	Indicates that DND is to be set or cleared

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

getHotdeskUserLoginDevice()

Definition

getHotdeskUserLoginDevice (sessionId, hotDeskUserobjectId);

Description

This operation gets device DN based on hotdesk user DN Object ID.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
hotDeskUserobjectId	Monitor object ID of the hotdesk user DN.

Returns

- result – true or false
- errorDescription – if result false
- hotDeskDeviceDn – if true returns phone DN of where user is logged in

Notes:

- None.

getHotdeskUserDn()

Definition

(sessionId, hotDeskUserobjectId);

Description

This operation gets hotdesk user DN based on phone DN Object ID.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
hotDeskDeviceobjectId	Monitor object ID of the phone DN.

Returns

- result – true or false
- errorDescription – if result false
- userDn – hotdesk user DN

Notes:

- None.

setPrgPresence()

Definition

setPrgPresence (sessionId, objectId, PrgDn, presenceValue, invokerDn, dn);

Description

This operation sets the presence of a PRG member.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the member.
PrgDn	Personal Ring Group DN
presenceValue	Sets the member to be in or out of PRG
invokerDn	invokcerDn allows a supervisor to change presence of a member
Dn	The DN can be used in place of ObjectId

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setCFNAExternal()

Definition

setCFNAExternal (sessionId, objectId, dn, state);

Description

This operation sets the call forwarding for a DN on or off related to an external call. If the called Dn does not answer when an external call is received, the call is forwarded to the specified destination.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the phone.
dn	The DN to forward to
state	On or off

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setCFNAInternal()

Definition

setCFNAInternal (sessionId, objectId, dn, state);

Description

This operation sets the call forwarding for a DN on or off related to an internal call. If the called Dn does not answer when an internal call is received, the call is forwarded to the specified destination.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the phone.
dn	The DN to forward to
state	On or off

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setCFBusyExternal()

Definition

setCFBusyExternal (sessionId, objectId, dn, state);

Description

This operation sets the call forwarding for a DN on or off related to an external call. If the Dn is busy when an external call is received the call is forwarded to the specified destination**.**

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the phone.
dn	The DN to forward to
state	On or off

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setCFBusyInternal()

Definition

setCFBusyInternal (sessionId, objectId, dn, state);

Description

This operation sets the call forwarding for a DN on or off related to an internal call. If the Dn is busy when an internal call is received the call is forwarded to the specified destination.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the phone.
dn	The DN to forward to
state	On or off

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

setCFAlways()

Definition

setCFAlways (sessionId, objectId, dn, state);

Description

This operation sets the call forwarding of the DN on or off.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the phone.
dn	The DN to forward to
state	On or off

Returns

- result – true or false
- errorDescription – if result false

Notes:

- None.

getDeviceFeatures()

Definition

getDeviceFeatures (sessionId, objectId);

Description

This operation gets the features settings for a phone DN or a hot desk user DN. If the objectId of a hotdesk user DN is provided, the registrationDn will provide the DN of the phone where the hotdesk user is logged in and hotDeskUserLoggedInDn will be blank. If the objectId of a phone DN is provided, then the registrationDn will be blank and hotDeskUserLoggedInDn will be provided.

Attributes

Attribute	Description
sessionId	sessionId was provided by the OIG upon successful login.
objectId	Monitor object ID of the hotdesk user DN or the monitor object ID of the phoner DN.

Returns

- result – true or false

- errorDescription – if result false
- Device Features includes hotDeskUserLoggedInDn, registrationDn, forwardingInfo (CFNAExt, CFNAInt, CFBusyExt, CFBusyInt, CFAlways). The forwarding info includes the DN destination and is on or off in the boolean fwdOn. The following three boolean items are also included: DND, autoAnswer, msgWaitingLamp

Notes:

- None.

Call Control Service Error Definition – Standard Type

Most operations return success or an error on failure. Errors are organized into the following categories:

- General Errors
- Communication Errors
- Service-Specific Errors

General Errors

Error	Description
COMMAND_REJECTED	A command was rejected by the MCD.
FAIL	A generic error was returned by the MCD.
UNSPECIFIED	An unspecified error has occurred. The OIG did not received error information from the MCD.

Communication Errors

Communication errors occur when a failure has occurred somewhere in the communication path between the application and the MCD.

Error	Description
CANNOT_CONNECT_SERVICE	The OIG was unable to connect to an MCD. This error indicates that the MCD was not found or the network is not configured correctly.
SERVER_UNAVAILABLE	When opening a connection to an MCD the OIG found the MCD was not started.
SERVER_NOT_READY	In opening the connection to the MCD, the MCD indicated call processing was not yet available.
SERVICE_FULL	The application was unable to connect to an OIG. This error indicates that the OIG is serving the maximum number of clients.

Service-Specific Errors

Service-specific errors are those relating to the failure of individual operations. These errors include attempts to monitor or manipulate unknown devices, and attempts to perform operations which are not valid.

Call Control Service Events – Standard Type

This section describes the OIG events provided to an application. The application can receive events synchronously or asynchronously. This means that the application can poll the OIG for events or ask the OIG to send events when they occur.

Error	Description
UNABLE_TO_ORIGINATE_CALL	The monitored device cannot place the requested call. This can occur because of a phone privilege violation.
Device is already monitored by the session.	Attempt to monitor the same device more than once.
DEVICE_NOT_MONITORABLE	Attempt to set monitor on a device which cannot be monitored by the MCD.
Failed to open connection to ICP at:	The application was unable to connect to the MCD because the MCD IP address provided is incorrect.
FEATURE_NOT_ALLOWED	Operation failed because the device was in a state in which the request could not be completed.
INVALID_ATTRIBUTE_VALUE	A parameter provided in an OIG operation is invalid or missing.
INVALID_LOCAL_CALL_ID	The specified call-ID is not valid. The call-ID changed at the device before the application invoked operation or the call-ID was never valid.
INVALID_DEVICE	Operation failed because the called device was determined not to be valid.
INVALID_DN	A specified DN is not valid.
INVALID_FEATURE	Attempt to alter a device feature failed / not valid.
MAX_DN_LENGTH_EXCEEDED	Specified DN exceeded allowed size.
INVALID_OBJECT_ID	An object ID was not valid. The monitor was previously closed or never opened.
FEATURE_NOT_ALLOWED	The active call was cleared or placed on hold before the application invoked call operation.
FEATURE_NOT_ALLOWED	Attempt to answer a non active call existed, possibly because the call had cleared.
FEATURE_NOT_ALLOWED	Attempted to clear a call when no call was present at the specified device.
FEATURE_NOT_ALLOWED	Attempted to split a conference call which does not exist.
FEATURE_NOT_ALLOWED	Attempted to manipulate a held call which does not exist.
MCD_TIMEOUT_ERROR	The MCD did not respond to an OIG operation request in the allowed time.
PRIVILEDGE_VIOLATION	Invoker does not have sufficient privileges.

Event Classes

OIG events are generated as a result of monitoring devices or features on the MCD. Events are divided into the following classes.

- Call Event Class (for Call Control Server Standard)
- Feature Event Class (for Call Control Server Standard)
- System Event Classs (for Call Control Server Standard)
- ACD Event Class (for Call Control Server Advanced)
- Conference Event Class (for Call Control Server Advanced)

Events in Call Event Class

Events included in the Call Event Class have the following names (Event Type).

- Account Code Set
- Call Cleared
- Call Conferenced
- Call Delivered
- Call Diverted
- Call Established
- Call Failed
- Call Held
- Call Originated
- Call Queued
- Call Received
- Call Retrieved
- Call Transferred
- Conference Held
- Device Dropped
- Extension In Use
- Group
- In Service
- Monitor Failed
- Monitor Set
- Out Of Service
- Remote Party Update

AccountCodeSet

Description

This Event is sent when an account code change is attempted. Both verified and non-verified account code changes are reported. The Event is generated when the account code is changed using operations (setAccountCode or makeCall) or by entering account codes manually from a phone. This Event is reported for monitored phones and trunks.

Event Data

Data	Description
Event Type	AccountCodeSet
Event State	OriginatedState, DeliveredState, EstablishedState.
Event Cause	AccountCodeSet (if succeeded) , InvalidAccountCode (if failed).
Event Attribute (Account Code)	The new Account Code (not valid if Cause is InvalidAccountCode).

CallClearedEvent

Description

This Event is sent when a call is cleared from the monitored device. This Event is generated when the monitored device invokes disconnect, or when the monitored device is operating in hands-free mode and the other device invokes disconnect. When the other device disconnects, the monitor for this device receives a CallOriginatedEvent because the phone is off hook. If the monitored phone is being operated in hands-free mode, a CallClearedEvent is received. An Event with a Cleared cause is reported to a monitor on an ACD2 agent when the caller abandons the call while the agent phone is ringing. This Event happens after a CallReceivedEvent and before the CallEstablishedEvent. The monitored phone goes to Idle State after the CallClearedEvent.

Event Data

Data	Description
Event Type	CallClearedEvent.
Event State	IdleState.
Event Cause	ClearInvoked (this device cleared) Cleared (other device cleared, this device is operating in hands-free mode). ACDAgentTimeout (this device is an ACD2 agent who did not answer in time. Such calls will normally be requeued to the path).
Event Attribute (Group Member Answered)	Event data used to determine whether a Personal Ring Group member has answered a call.

Data	Description
Event Attribute (External HotDesk User DN)	Event data indicating the External Hot Desk User DN digits.
Event Attribute (Is Work Timer Active)	TRUE if work timer is active. Otherwise FALSE.
Event Attribute (OIG Time Stamp)	The time when the Event was received at the OIG.
Object ID (Monitor ID)	Identifies the monitor for the Event.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallConferencedEvent

Description

This Event indicates that a conference call has been created. The identity of the conference invoker is provided. When a Silent Monitor is established on a conference, the use of Event reporting for splits or Events for more than three conference members is not supported. When a three-device conference is silent monitored, redundant Events should also be expected. When any member or the silent monitor clears out of a Silent Monitored three-device conference, a DeviceDroppedEvent will be reported, with a cause of ConferenceMemberDropped, or a cause of ACDSilentMonitorMemberDropped. These causes are equivalent in this case, and device information must be used to determine whether a regular member or the Silent Monitor cleared from the silent-monitor type of conference. After this Event the monitored device goes into Established State.

Conference data is also provided in ConferenceHeldEvent, DeviceDroppedEvent, CallRetrievedEvent when a conference has been created.

Event Data

Macro	Description
Event Type	CallConferenceEvent
Event State	EstablishedState.
Event Cause	ACDSilentMonitor ACDSilentMonitorEstablished Conferenced ConferenceInvoked Intrusion IntrusionInvoked.
Event Local Call ID	Identifies the conferenced call.

Macro	Description
Event Attribute (Number of Members)	Returns the number of members involved in the conference.
Event Attribute (List of Member Device Types)	The device type for each conference member (e.g., VOICE_SET, PRIVATE_TRUNK).
Event Attribute (List of Member Local Call IDs)	A list of Local Call IDs (i.e., local call ID reported to each monitor on a phone in the conference). This allows an application to correlate the call legs of the conference seen by different phone monitors.
Event Attribute (List of Member DNs)	A list of DNs (i.e., DN of each conference member).
Event Attribute (List of Member Names)	A list of Names (i.e., Name of each conference member).
Event Attribute (Conference Invoker Device)	Identity of the device that invoked the conference. In the case of a conference created by intrusion, device is intruded-upon member, since the intruded device puts its currently connected device on soft hold, then accepts the intrusion and creates a conference.
Event Attribute (DN of Remote Conference Member)	Directory number of the remote conference member.
Event Attribute (Name of Remote Conference Member)	Name of the remote conference member, if present.
Event Attribute (DN & Node ID of Remote Conference Member)	Directory number of the remote conference member, including the Node ID if programmed.
Event Attribute (DN of External Hotdesk User Conference Member)	DN of External Hot Desk User.
Event Attribute (OIG Time Stamp)	The time when the Event was received at the OIG.
Object ID (Monitor ID)	Identifies the monitor for the Event.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallDeliveredEvent

Description

This Event is sent when the monitored device has completed dialing and the destination is ringing (calling device is receiving a ringback tone). The monitored device is now in DeliveredState. An advisory status message is only provided in this Event when the cause is ReceivedState, DeliveredState, or FailedState. A trunk monitor will report a CallDeliveredEvent for an incoming external call that rings a local extension. For dial-in trunks, this Event may be preceded with a CallOriginatedEvent. If the call is to a local ACD2 path, no CallQueuedEvent will be generated for the trunk, even if the call is queued to a group within the path. This queued information must be obtained by monitoring the ACD2 path.

Event Data

Data	Description
Event Type	CallDeliveredEvent
Event State	DeliveredState
Event Device Data (Local Call ID of ringing Device)	Local Call ID reported to monitor on the device at the other end of the call. This allows correlating calls seen by the different monitors.
Event Cause	ACDSilentMonitorInvoked Delivered (destination alerting) DivertedAlwaysTo DivertedNoAnswerTo DivertedOnBusyTo HandoffPush RedirectedOnError RedirectedTo UnsupervisedTransfer.
Event Device Data (Called Device)	Identity of the device which is now ringing.
Event Device Data (Original Dialed Destination)	Device initially associated with this call if different from currently ringing device (forwarded from, diverted from).
Event Attribute (Suite Pilot Number)	The Suite (or Linked Suite) Pilot Number if the device is a Suite Extension, provided that STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Suite Pilot Name)	The Name associated with the Suite (or Linked Suite) Pilot Number if the device is a Suite Extension, provided that STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Line Appearance Number)	The line appearance (keyline or multicall group number) of the other device, if such is involved.
Event Attribute (Line Appearance Name)	Returns the name associated with the line appearance (keyline or multicall group number) of the other device, if such is involved.
Event Attribute (Prime Line Number)	The prime line number of the called phone.
Event Attribute (Dialed Digits)	Digits dialed by caller (for example, DDI, DNIS).
Event Attribute (Device Advisory Status Message)	Advisory status message from the called device.
Event Attribute (Device Network Extension)	Identity of the remote called device, including the Node ID if programmed in the MCD.

Data	Description
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if provided.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number.
Event Attribute (External Hotdesk User DN)	The External Hot Desk User DN digits.
Event Attribute (OIG Time Stamp)	The time when the Event was received at the OIG.
Object ID (Monitor ID)	Identifies the monitor for the Event.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallDivertedEvent

Description

This Event is sent when a call is diverted from the monitored device as a result of call forwarding, rerouting, or being picked up by a third device. The monitored device is now in Idle State (device is no longer ringing). When a call is diverted from a device due to CallForwardAlways or CallForwardBusy, no Event is reported for the diverting device. A monitor on the calling device reports a Call Delivered Event with a cause of DivertedAlwaysTo or DivertedOnBusyTo. A monitor on the diverted-to device reports a Call Received Event with a cause of DivertedAlwaysFrom or DivertedOnBusyFrom. When a call is diverted from a monitored device by the user pressing the Forward Call button, a Call Diverted Event with a cause of RedirectedAway is reported. The same cause is reported when redirectCall() is used to effect the forwarding. Events do not distinguish between user pressing the Forward Call button and application using redirectCall() operation.

Event Data

Data	Description
Event Type	CallDivertedEvent
Event State	IdleState (call has been diverted away from the monitored phone).
Event Cause	DivertedNoAnswerAway DivertedPickupAway (picked up at a third device) RedirectedAway.
Event LocalCallId	Local Call ID is 0 for this Event.
Event Device Data (New Destination Device)	Identity of device that is now ringing or has answered.

Data	Description
Event ObjectID	Identifies the monitor for which the Event is reported.
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if there is one in the Event.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallEstablishedEvent

Description

CallEstablishedEvent is sent when the device has answered a call, either directly or via call pickup. A trunk monitor will report the call established Event on incoming calls when they are connected to a local extension, as you would expect. On outgoing calls, however, this Event implies a complete connection only for certain types of digital trunks. For analog trunks, this Event will be reported even if the remote phone set is busy; the next Event seen by the trunk monitor will be a DeviceDroppedEvent when the local user hears the busy tone and hangs up. For this reason, Call Established Events on outgoing trunk calls should never be used for billing purposes. The monitored device is now in the EstablishedState.

Event Data

Data	Description
Event Type	CallEstablishedEvent
Event State	EstablishedState.
Event Attribute SX_DeviceCallid (Event Attribute SX_ConnectedDevice(Eventp))	The call reference number that will be reported for this call by a monitor set on the device at the other end of the call. This allows programs to correlate calls seen by different monitors
Event Cause	Indicates how the call entered this state. Possible values include: Answered AnswerInvoked ConferenceSplit ConferenceSplitInvoked Pickup PickupInvoked SupervisedTransfer.

Data	Description
Event Device Data (Connected Device Local Call ID)	The Local Call ID reported for this call by a monitor on the phone at the other end of the call.
Event Device Data (Connected Device Number)	Identity of the device connected to this monitored device.
Event Attribute (Suite Pilot Number)	The Suite (or Linked Suite) Pilot Number if the device is a Suite Extension, provided that STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Suite Pilot Name)	The Name associated with Suite (or Linked Suite) Pilot Number if the device is a Suite Extension, provided that STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Line Appearance Number)	The line appearance (keyline or multicall group number) of the connected device, if such is involved. (The prime number of the set is always reported when Event Attribute SX_DeviceNumber is applied to the device).
Event Attribute (Prime Line Number)	The prime line number of the connected device.
Event Device Data (Pickup Device)	Identity of the Device that picked up the call to cause call to this device.
Event Device Data (Spliting Device)	Identity of the Device that split a conference to cause call to this device.
Event Device Data (Transfering Device)	Identity of the Device transferring a call to this device.
Event Attribute (Remote Device Number)	Number of the remote called device, including the Node ID if programmed.
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if provided.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number.
Event Attribute (External Hotdesk User DN)	The External Hot Desk User DN digits.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallFailedEvent

Description

This Event is sent when the monitored device initiates a call that cannot be completed. The MCD can be programmed to reroute failed calls to an error handling number (e.g., attendant). In his case failed calls from a monitored device result in CallDeliveredEvent with a RedirectedOnError cause (possibly in addition to an CallFailedEvent). With no error handling number only a CallFailedEvent is reported. A trunk monitor reports this Event for an incoming call to a busy extension unless the trunk is analog. An analog trunk will camp on to the busy local extension, generating a CallQueuedEvent. Outgoing calls which fail do not report a CallFailedEvent to the trunk number. An advisory status message is only provided when the call cause is in ReceivedState, DeliveredState, or FailedState. The monitored device is now in the FailedState.

Event Data

Data	Description
Event Type	CallFailedEvent
Event State	FailedState
Event Cause	CFADestinationBusy CFBDestinationBusy DestinationBusy DestinationDoNotDisturb DestinationNotObtainable (locked out / in maintenance) ErrorDetected NetworkBusy NetworkCongestion NetworkNotObtainable InvalidAccountCode NoAnswer NoSuchNumber PrivilegeViolation Destination Unavailable TrunksBusy.
Event Device Data (Called Device)	Identity of device which was called.
Event Attribute (Advisory Status Message)	The advisory status message from called device.
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if provided.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallHeldEvent

Description

This event is sent when the monitored device has placed a call on hard hold or has been placed on hold by another device in the call. The monitored device is now in the HeldState.

There is a special condition where the event needs to be understood from an application. Consider the following scenario:

1. Set Monitor on Phone1 and Phone2
2. Establish a two party call between Phone1 and Phone2
3. Invoke CC Operation HoldCall() on Phone1
4. Verify Events:

On Phone2:

1. CallHeldEvent HeldState ConsHold
2. CallHeldEvent HeldState HardHold

On Phone1:

CallHeldEvent HeldState HardHoldInvoked

Expected result:

Phone1 does not get Call Held Event with Cause Conshold

Event Data

Data	Description
Event Type	CallHeldEvent
Event State	HeldState
Event Cause	ConferenceSplit (softhold) ConsHold (Other device invoked softhold) HardHold (Other device invoked hardhold) HardHoldInvoked (This device invoked hardhold)
Event Attribute (External Hotdesk User DN)	External Hot Desk User DN digits.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallOriginatedEvent

Description

This event is sent when the monitored device has initiated dialing activity (for example, has gone off-hook or is dialing hands-free). The MCD is providing dial tone or processing dialed digits. A call is about to be originated. The monitored device is now in the OriginatedState. Monitors set on dial-in trunks report a CallOriginatedEvent for incoming external calls. This event indicates a call is being sent from the monitored device.

Event Data

Macro	Description
Event Type	CallOriginatedEvent
Event State	OriginatedState.
Event Cause	CallbackInvoked ConsCallInvoked (with no directory number or invoked from the set) HardHoldInvoked (2500 sets enter OriginatedState after invoking hard hold) NewCallInvoked (the normal case) OtherDeviceCleared
Event Attribute (External Hotdesk User DN)	External Hot Desk User DN digits.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallQueuedEvent

Description

CallQueuedEvent is sent when this device queues onto another device, or when another device queues onto this monitored device. This event is generated when a call from this monitored device is sent by unsupervised transfer to a busy device. A trunk monitor receives this event for an incoming call on an analog trunk when the local destination device is busy. Analog trunks queue rather than failing (CallFailedEvent). Digital trunks receive CallFailedEvent when the local destination device is busy.

Event Data

Macro	Description
Event Type	CallQueuedEvent
Event State	**QueuedState**: The monitored device has queued onto another device. **Any State**: When the monitored device has been queued onto the device state remains unchanged from the previous Event.
Event Cause	CallQueued – the monitored device has queued onto another device. CallIsWaiting – a caller has queued onto this monitored device. TransferToBusy – this monitored calling device was transferred without supervision to a busy device and is now queued on that device. **Exception**: If this monitored device is a Ring Group or Hunt Group, and the state of this device is ReceivedState rather than QueuedState, then the caller was transferred to this group and is waiting for an available member. The Event Attributes Waiting Device and Transferrer Device are provided. CFACallQueued – this monitored calling device has been queued onto a ring group, with all busy members, by a third device with call forwarding always. CFBCallQueued – this monitored calling device has been queued onto a ring group, with all busy members, by a third device set to call forwarding when busy. CFNACallQueued – this monitored calling device has been queued onto a ring group, with all busy members, by a third device set to call forwarding no answer.
Event Device Data (Waiting Device)	Identity of the Other Device. The device which this monitored device is queued onto; or the other device queued onto this monitored device.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallReceivedEvent

Description

This event is sent when the monitored device is ringing (a call is presented to the device). This monitored device is now in ReceivedState.

Event on a trunk monitor - A digital trunk monitor reports this event for an outbound external call (except DPNSS and DASS2). This event is not reported for analog trunk monitors. If a trunk monitor, ANI is in this event if provided by the MCD.

Event on a ACD agent monitor - This event has ACDCallReceived cause if an ACD2 agent and the call is distributed to this ACD2 agent via ACD path. When this event is from a remote ACD Path and an application is monitoring the ACD Agent Group DN, the identity of the remote ACD Path can be determined; correlate this event with the corresponding ACDRemoteCallDistributed event reported to the ACD Agent Group monitor. The correlation is the same Local Call ID in both Events. The application must be prepared to receive either Event first. The ACDRemoteCallDistributed event provides the ACD Path DN.

An advisory status message is only provided when the call status is in ReceivedState, DeliveredState, or FailedState.

The Calling Device Number may have network information separated from DN by a " / ". The slash differentiates the network portion of the CLI from the DN. Application should parse the number string for 0-9, # and * to remove any slashes and spaces.

Event Data

Data	Description
Event Type	CallReceivedEvent
Event State	ReceivedState.
Event Cause	ACDHelpCall ACDCallReceived CallbackMatured DivertedAlwaysFrom DivertedNoAnswerFrom DivertedOnBusyFrom NewCall (the normal case) ONSCallOriginated RecallHeldParty (see Note 6.) RedirectedFrom (for other reasons) TapCall TransferRecallNoAnswer TransferRecallOnBusy TransferredCall WaitingCallRinging
Event Attribute (Calling Device Local Call ID)	The Local Call ID for this call reported to the Calling Device (Other Device).

Data	Description
Event Device Data (Calling Device)	Identity of the Calling Device (Other Device) **Exception**: Can not be used to determine the device calling an IP RAD. In this case the IP RAD device is provided.
Event Attribute (Remote Calling Device Number)	The number of remote calling Device including the Node ID if programmed.
Event Device Data (Original Called Device – Third Device)	The Original Called Device if different from this monitored device (diverted from device, forwarded from device, and device doing unsupervised tranfer). For ACDCallReceived cause only, this monitored device is ACD2 Path (except when the agent is part of a Remote Agent Group, and this call is from the remote Path. See description above.
Event Attribute (Dialed Digits)	Digits dialed by caller (e.g., DDI, DNIS).
Event Device Data (ACD Agent Group)	The identity of ACD group that distributed call to this monitored device (i.e., ACD agent). This device data only applies when the event cause is ACDCallReceived. If this event cause is ACDCallReceived and this event ACD Agent Group Number is NULL, and this agent is part of a Remote Agent Group, then this call has come from a remote ACD Path. In this case Dialled Digits above provides the Agent Group DN. If the application has a monitor on the Agent Group DN, the application can determine the remote ACD Path that sent the call that triggered this event.
Event Attribute (Advisory Status Message)	The advisory status message for this monitored device.
Event Attribute (Suite Pilot Number)	The Suite (or Linked Suite) Pilot Number for a call from a Suite Extension, provided STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Suite Pilot Name)	The Name associated with the Suite (or Linked Suite) Pilot Number for a call from a Suite Extension, provided that STS (Shared Telephone Service) is set to YES on the MCD.
Event Attribute (Line Appearance Number)	The line appearance (keyline or multicall group number) of the calling device, if such is involved.
Event Attribute (Prime Line Number)	The prime line Number of the device with the calling line appearance (keyline or multicall group number) if such is involved.
Event Attribute (Calling Device Type)	Event Attribute SX_PUBLIC_TRUNK Event Attribute SX_PRIVATE_TRUNK Event Attribute SX_VOICE_SET Event Attribute SX_DATA_EXTENSION or –1 for error.

Data	Description
Event Attribute (Calling Line Category)	For trunk calls, this attribute gives "-1" if information is not available or: CLC_ORD CLC_DEC CLC_DASS2 CLC_PSTN CLC_MF5 CLC_OP CLC_NET CLC_CONF CLC_ANI
Event Device Data (Group Device)	A group device if involved or NULL if not.
Event Attribute (Group Distribution Type)	The Group Distribution Type: RINGGRP_RINGALL, RINGGRP_CASCADE, HUNTGRP_LINEAR, HUNTGRP_CIRCULAR, or PERSONALRINGGRP.
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if there is one in the Event.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number if there is one in the Event.
Event Attribute (External Hotdesk User DN)	External Hot Desk User DN.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallRetrievedEvent

Description

This event is sent to both devices (now connected) when a call is retrieved from either consultation hold or hard hold. The monitored device is now in EstablishedState.

Event Data

Data	Description
Event Type	CallRetrievedEvent
Event State	EstablishedState.

Data	Description
Event Cause	AlternateCallInvoked ConfRetrieveInvoked Retrieve RetrieveInvoked.
Event Attribute (Calling Device Local Call ID)	The Local Call ID for this call reported to the Calling Device (Other Device).
Event Device Data (Connected Device)	Device to which monitored device is now connected.
Event Attribute (Calling Line Category)	Provides category of calling device on incoming trunk call (indicates whether call is from within a cluster or from PSTN). Values are: CLC_ORD - Call originated within private network CLC_OP - Operator initiated call, Call originated within private network CLC_NET – Call originated within private network CLC_CONF - Call originated within private network. **Note**: If a device in PSTN conferences in a device from the private network then the CLC can be one of CLC_DEC CLC_DASS2 CLC_PSTN CLC_MF5. But **never** CLC_CONF CLC_DEC - Call originated from PSTN CLC_DASS2 - Call originated from PSTN CLC_PSTN – Call originated from PSTN CLC_MF5 - Call originated from PSTN. CLC_NIL – not provided. A value of “-1” is provided when not a trunk call.
Event Attribute (IsExternalTrunk)	Indicates if calling trunk is Internal (0) or External (1). A value of “-1” means info not available. Internal: CLC_ORD CLC_OP CLC_NET CLC_CONF External: CLC_DEC CLC_DASS2 CLC_PSTN CLC_MF5
Event Device Data (ACD Agent Group)	ACD group that distributed call to the agent (for cause ACDCallReceived only).
Event Attribute (Remote Device Number)	The remote device number of the remote conference member (only for Retrieved Events with cause ConfRetrieveInvoked).

Data	Description
Event Attribute (Remote Device Name)	The remote device name of the remote conference member (only for Retrieved Events with cause ConfRetrieveInvoked).
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if there is one in the Event.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number if there is one in the Event.
Event Attribute (External Hotdesk User DN)	External Hot Desk User DN.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

CallTransferredEvent

Description

This event is sent when an existing call is transferred to another device and the device performing the transfer has been dropped from the call. Both the transferring device and the transferred device receive this event.

A monitor on an ACD agent receives a CallTransferredEvent when an ACD agent has transferred an existing call to another device and the agent performing the transfer has been dropped from the call. When the agent has dropped from the call, a work timer is activated (provided it has been programmed).

Event Data

Data	Description
Event Type	CallTransferedEvent
Event State	The current state of the monitored device.
Event Cause	TransferInvoked Transfer.
Event Device Data (Transferred Device)	The transferred device when a transfer is invoked by this monitored device.
Event Attribute (External Hotdesk User DN)	External Hot Desk User DN.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID

Data	Description
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.
Event Attribute (IsWorkTimerActive)	Is work timer is active or not.

ConferenceHeldEvent

Description

This event is sent when the monitored device is in a conference and another conference member places the conference on hold, leaving this monitored device and other members still communicating. The monitored device is now in the HeldState.

Event Data

Data	Description
Event Type	ConferenceHeldEvent
Event State	HeldState
Event Cause	ConsHold (invoked by another device) HardHold (invoked by another device) HardHoldInvoked (monitored device invoked)
Event Attribute (Conference Holding Device Number)	The number of the device which invoked the hold.
Event Device Data (Conference Holding Device)	Identity of the device which invoked the hold.
Event Attribute (Conference Holding Device Name)	The name of the device which invoked the hold.
Event Attribute (Remote Conference Member Number)	DN of the remote conference member.
Event Attribute (Remote Conference Member Name)	Name of the remote conference member, if present.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

DeviceDroppedEvent

Description

This event is sent to this monitored device when another device drops out of a conference call or a held/waiting device abandons. The identity of the device that dropped from the call is provided. This event is not generated when this monitored device does a transfer. This case reports CallTransferredEvent instead.

A trunk monitor reporting a locally cleared call presents a special case of a DeviceDroppedEvent. With a trunk in a "conference" and the local phone drops out, the trunk reports a DeviceDroppedEvent. The trunk monitor then reports a CallClearedEvent. No DeviceDroppedEvent is reported on a trunk monitor when the remote end clears.

An agent monitor receives a DeviceDroppedEvent when a held/waiting device abandons or disconnects and a work timer is activated.

Event Data

Data	Description
Event Type	DeviceDroppedEvent
Event State	Any state: The state will remain unchanged from the last event. IdleState and OriginatedState are not possible in this event.
Event Cause	ConferenceMemberDropped ConsHeldPartyDropped HardHeldPartyDropped OtherDeviceDropped (ONS phones and trunks only; state = unavailable) QueuedPartyDropped (ACD and call waiting) ThisDeviceDropped (ONS phone only; state = unavailable). ACDSilentMonitorDropped (only from a Silent-Monitored two-device call) ACDSilentMonitorMemberDropped (any member of a conference formed with a Silent Monitor)
Event Device Data (Dropped Device)	The device which has dropped out of the call.
Event Attribute (Remote Conference Member DN)	DN of the remote conference member (only when the cause is ConferenceMemberDropped and there are three or more parties left in the conference).
Event Attribute (Remote Conference Member Name)	Name of the remote conference member, if present (only when the cause is ConferenceMemberDropped and there are three or more parties left in the conference).
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.
Event Attribute (IsWorkTimerActive)	Is work timer is active or not.

ExtensionInUseEvent

Description

ExtensionInUseEvent is sent when a device, which is not the monitored device, is using this this monitored device. This occurs when another device makes, answers, retrieves, etc., a call on this monitored device.

Event Data

Data	Description
Event Type	ExtensionInUseEvent
Event State	ExtensionInUseState.
Event Cause	ExtensionInUse.
Event Device Data (Using Device)	The device currently using this monitored device.
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if there is one in the Event.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number if there is one in the Event.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

GroupEvent

Description

This event is sent when monitoring Ring Group or Hunt Group pilot numbers. For distribution device type RINGGRP_RINGALL, a GroupEvent with CallDistributed cause is generated for each member all at once. When one member answers, only one GroupEvent with CallAnswered cause is generated. For distribution device types RINGGRP_CASCADE, HUNTGRP_LINEAR and HUNTGRP_CIRCULAR, a GroupEvent with CallDistributed cause is generated for each member in sequence. A timeout triggers the MCD to the next group member. A GroupEvent with CallAnswered cause is generated when one of those members answers.

Event Data

Macro	Description
Event Type	GroupEvent
Event State	idle or received
Event Cause	**CallDistributed** – call has been distributed to group member(s) . **CallAnswered** – call has been answered by a group member. **CallDisconnected** – call has been distributed to the members of a group and the caller has disconnected the call. **CallOverflowed** – call has been distributed to the members of a group and a ringing timer has expired. The call is transfered to an overflow point. **CallWaiting** – call is queued to a group when all group members are busy. **CallWaitingPartyDisconnected** – caller disconnects while queued on a busy group.
Event Device Data (Group Member Device)	Identifies member device. Valid for CallDistributed and CallAnswered event causes.
Event Device Data (Group Overflow Device)	Identifies the overflow device. Valid for CallOverflowed event cause.
Event Device Data (Calling Device)	Identifies the calling device. Valid for CallDistributed, CallDisconnected, CallAnswered and CallOverflowed event causes.
Event Device Data (Group Waiting Device)	Identifies the waiting device of the queued call. Valid for CallWaiting and CallWaitingPartyDisconnected event causes.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

InServiceEvent

Description

This event is sent when this monitored device is put into service by the MCD. A device is taken out of service when a phone user exceeds a MCD allowable time period while attempting to place a call (e.g., user lifted phone receiver and di not dial any digits). When the phone receiver is placed back on hook, an InServiceEvent is received, indicating the device is back in service. A device is also be taken out of service and put back into service, by MCD maintenance activities (e.g., power up, reset of MCD, reset device).

Event Data

Data	Description
Event Type	InServiceEvent
Event State	IdleState.
Event Cause	DeviceInService **ControllerCommsRestored**: when communication with MCD restored for non-Resilient monitor.
Event Attribute (Local Call ID)	Null (0).

MonitorSetEvent

Description

This event is sent when a monitor is initially set on a device.

To determine the state of the device application must use getCallStatus operation.

Event Data

Macro	Description
Event Type	MonitorSetEvent
Event State	Nil
Event Cause	Unknown
Event Attribute (Local Call ID)	Local Call ID for active call (if any).

MonitorFailedEvent

Description

This event is sent when an attempt to set a monitor fails on a device.

To determine the state of the device an application must use getCallStatus operation.

Event Data

Macro	Description
Event Type	MonitorFailedEvent
Event State	Nil
Event Cause	Unknown

OutOfServiceEvent

Description

This event is sent when a device is taken out of service. A device is taken out of service when a phone user exceeds a MCD allowable time period while attempting to place a call (e.g., user lifted phone receiver and di not dial any digits). When the phone receiver is placed back on hook, an InServiceEvent is received, indicating the device is back in service. A device is also be taken out of service and put back into service, by MCD maintenance activities (e.g., power up, reset of MCD, reset device).

Event Data

Data	Description
Event Type	OutOfServiceEvent
Event State	UnavailableState, IdleState.
Event Cause	DeviceOutOfService **ControllerCommsFailed** -- reported to a non-Resilient monitor when communication with MCD fails.
Event Attribute (Local Call ID)	Null (0).

RemotePartyUpdateEvent

Description

This event occurs when this monitored device is connecting to or connected to, a remote device via a MCD trunk and the remote device changes. Normally a result of the remote device transferring this monitored device to a different remote device (remote device Number and remote device Name change).

This event occurs when a PRI trunk call is being set up. The CallReceivedEvent or CallDeliveredEvent carry the remote device Number but not Name. The Name is reported after via this Remote Party Update Event.

No RemotePartyUpdateEvent is reported as a remote conference is built up. When a conference collapses back to a 2-device call, a RemotePartyUpdateEvent is generated.

Event Data

Data	Description
Event Type	RemotePartyUpdateEvent
Event State	Any State: The current state of the device.
Event Cause	RemotePartyUpdated GlobalCallIDUpdated
Event Attribute (Local Call ID)	Local Call ID for active call.

Data	Description
Event Device Data (Connected Device)	Identity of the newly connected or connecting device
Event Attribute (Connected Device Number)	Number of the newly connected or connecting device
Event Attribute (Connected Device Name)	Name of the newly connected or connecting device
Event Attribute (Personal Ring Group Name)	The Personal Ring Group Name if there is one in the Event.
Event Attribute (Personal Ring Group Number)	The Personal Ring Group Number if there is one in the Event.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

RoutingDeviceEvent

Description

This event is only reported when the monitored device is a routing device or a RAD of the routing device.

Event Data

Data	Description
Event Type	RoutingDeviceEvent
Event State	The current state of the device.
Event Cause	RoutingDeviceCallQueued. RoutingDeviceCallOverflow RoutingDeviceCallRerouted RoutingDeviceCallAbandoned RoutingDeviceTransCallQueued RoutingDeviceReroutedCallQueued RoutingDeviceCFBUSYCallQueued RoutingDeviceCFNOANSCallQueued RoutingDeviceCFALWAYSCallQueued RoutingDeviceRadStarted
Event Attribute (Local Call ID)	Local Call ID for the monitored routing device.
Event Device Data (Routing Device)	Identity of the routing device.

Data	Description
Event Device Data (Calling Device)	Identity of the calling device.
Event Device Data (New Destination Device Name)	Identity of the new destination device name for RoutingDeviceCallRerouted cause.
Event Device Data (New Destination Device Number)	Identity of the new destination device number for the RoutingDeviceCallRerouted cause.
Event Device Data (Transferrer Device)	Identity of the transferring device for RoutingDeviceTransCallQueued cause.
Event Device Data (Rerouter Device)	Identity of the transferring device for RoutingDeviceReroutedCallQueued cause.
Event Device Data (Forwarder Device)	Identity of the forwarding device for RoutingDeviceCFBUSYCallQueued RoutingDeviceCFNOANSCallQueued RoutingDeviceCFALWAYSCallQueued causes.
Event Attribute (Current Global Call ID)	Each call has a Global Call ID
Event Attribute (Primary Global Call ID)	When a Global Call ID is changed for a call, the old Global Call ID is reported in the Primary Global Call ID.
Event Attribute (Secondary Global Call ID)	In some cases, where a secondary call exists, then the Secondary Global Call ID is used.

Call Cause Data

The following table summarizes data that is present for the various call causes:

Call Cause	Routing Device	Other Party	Third Party
RoutingDeviceCallQueued	yes	caller	no
RoutingDeviceCallOverflow	yes	destination provided by HCI or default destination	no
RoutingDeviceCallRerouted	yes	caller	new destination
RoutingDeviceCallAbandoned	yes	no	no
RoutingDeviceTransCallQueued	yes	caller	transferring
RoutingDeviceReroutedCallQueue	yes	caller	re-router
RoutingDeviceCFBUSY (CFNOANS, CFALWAYS) CallQueue,	yes	caller	forwarder
RoutingDeviceRadStarted	yes	no	no

PrgMonitorSetEvent

Description

This event is sent when a monitor is initially set on a Personnel Ring Group.

Event Data

Macro	Description
Event Type	PrgMonitorSetEvent
Event State	Nil
Event Cause	Unknown

PrgMonitorFailedEvent

Description

This event is sent when an attempt to set a monitor fails on a Personnel Ring Group.

Event Data

Macro	Description
Event Type	PrgMonitorFailedEvent
Event State	Nil
Event Cause	Unknown

Events in Feature Event Class

Events included in the Call Event Class have the following names (Event Type):

- Activate Feature
- Forward
- Hot Desk User
- JoinGroup
- LeaveGroup
- User ID

ActivateFeatureEvent

Description

This event identifies a binary feature that has changed.

Even Data

Data	Description
Event Attribute (Feature)	AutoAnswerFeature DoNotDisturbFeature InServiceFeature MakeBusyFeature MakeBusyStateFeature DoNotDisturbAll (Disabled state only)
Event Attribute (Make Busy Code)	Reason Code for the MakeBusyStateFeature.

ForwardFeatureEvent

Description

ForwardFeatureEvent is sent when the forwarding is changed on a device.

Event Data

Data	Description
Event TYpe	ForwardFeatureEvent.
Event State	FeatureEnabled or FeatureDisabled.
Event Attribute (Feature)	CFAlways CFBusyExternal CFBusyInternal CFNoAnswerExternal CFNoAnswerInternal CFNoAnswerBusy CFAllModes. **Note**: CFAllModes is reported with Disabled state only, and may be received even when Call Forwarding is already disabled.
Event Attribute (Forward Destination Number)	Destination to which calls are now set to be forwarded, when forwarding is enabled.

JoinGroupFeatureEvent

Description

The JoinGroupFeatureEvent is sent when a personnel ring group member is marked as presence in the group (setPrgPresence). When a Device Feature Monitor is set on a group member a JoinGroupFeatureEvent is generated when member is changed to present in the group.

Event Data

Data	Description
Event Type	JoinGroupFeatureEvent.
Event Attribute (Feature)	JoinGroupFeature
Event Attribute (Group DN)	Group DN
Event Attribute (Member DN)	The member DN
Event Attribute (objectID)	objectID for the member. Only provided when using device feature monitor. Class system monitor events for all Personnel Ring Groups do not provide objectIDs.

LeaveGroupFeatureEvent

Description

The LeaveGroupFeatureEvent is sent when a personnel ring group member is marked as absent in the group (setPrgPresence). When a Device Feature Monitor is set on a group member, a LeaveGroupFeatureEvent is generated when member is changed to absent in the group.

Event Data

Data	Description
Event Type	LeaveGroupFeatureEvent.
Event Attribute (Feature)	LeaveGroupFeature
Event Attribute (Group DN)	Group DN
Event Attribute (Member DN)	The member DN.
Event Attribute (objectID)	objectID for the member. Only provided when using device feature monitor. Class system monitor events for all Personnel Ring Groups do not provide objectIDs.

HotDeskUserFeatureEvent

See **UserIDFeatureEvent** below.

UserIDFeatureEvent

Description

UserIDFeature Event is sent when a Hot Desk User logs in or logs out.

Event Data

Data	Description
Event Type	UserIdFeatureEvent
Event State	FeatureEnabled (logged in) OR Feature Disabled (logged out)
Event Attribute (User Number)	Identity of the User who is logging in (User DN).
Event Attribute (Reg Number)	Identity of the set that the User is logging into (Registration DN).
*Event Attribute SX_ExtHotDeskUserDn (98evice)	Returns External Hot Desk User DN digits.
Event Attribute SX_ExtractUserType (Eventp)	Returns the user type of EHDA: Internal or External

Events in System Event Class

ICPCommunicationEvent

Description

This event indicates the connection status to a specific MCD.

Event-Specific Event Data

Data	Description
Event Type	ICPCommunicationEvent
icpId	MCD object Id
time	OIG time
connectedState	Indicates the MCD connection status to the OIG

OIGGatewayShutdownEvent

Description

This event indicates the OIG has shutdown.

Event-Specific Event Data

Data	Description
Event Type	OIGGatewayShutdownEvent
time	OIG time

Event States

The following is a list event states. The event states inform the application about the state of the object being monitored. For example, when a monitored phone is used to make a call to another phone, when the called phone receives the call and is ringing the calling device goes into delivered state and receives a call delivered event.

Call States

Call State	Description
Delivered	The user has finished dialing and is awaiting answer from the other end. The user would normally be listening to ringback tone at this point for an internal call.
Established	The user is connected to another device (that is, talking).
ExtensionInUse	The device is in use.
Failed	A problem has occurred while trying to process user’s request. Explanations for this failure include (among others): The user has dialed a busy extension The user has been off-hook too long without dialing The user has dialed a phone that has activated do-not-disturb The user has dialed a non-existent number The target has violated class of service, class of restriction, or interconnect privileges in the call attempt. If the device remains off-hook for a long enough period while in this state, the device is moved to UnavailableState (in-lockout) to preserve MCD resources.
FeatureDisabled	A particular feature is disabled.
FeatureEnabled	A particular feature is enabled.
HeldState	The user has been placed on consultation hold by the other end. The user has been placed on call hold by the far end.
Idle	The device is idle (that is, on hook).

Call State	Description
Originated	The user has gone off-hook and is listening to dial tone, that is, ready to originate a call. User remains in originated state until enough digits are dialed to allow the call to be routed or a time-out/error occurs.
OutofService	The device is out of service.
Queued	The user has queued (camped) onto another device.
Received	The device is ringing. In the case of a trunk, the trunk has been seized and has responded properly and trunk signalling is in progress.
Unavailable	The set has been taken out of service as a result of being off hook for too long or by MCD maintenance.

Feature States

Feature State	Description
FeatureDisabled	A particular feature is disabled.
FeatureEnabled	A particular feature is enabled.

Device States

Device State	Description
ExtensionInUseState	The device is in use.
OutofServiceState	The device is out of service.
UnavailableState	The set has been taken out of service as a result of being off hook for too long or by MCD maintenance.

Event Causes

The following is a list of the possible causes for each event and an explanation of what each event cause means. Thus each event has information relating to why the event was generated.

Event Cause	Description
AccountCodeSet	This party provided a valid account code.
ACDAgentTimeout	This device is an ACD2 agent who did not answer in time. Such calls will normally be requeued to the path.

Event Cause	Description
ACDCallAbandoned	The caller hung up either while waiting for an agent or while talking to an agent. This cause is reported from the perspective of the ACD structure being monitored. It occurs only for ACD structure monitors or ACD2 path monitors.
ACDCallDelivered	The call was delivered to an ACD agent. This event is sent when the ACD agent is being rung. The monitored device is receiving ringback tone. This cause is reported from the perspective of the ACD structure being monitored. It occurs only for ACD structure monitors or ACD2 path monitors.
ACDCallQueued	A call has entered the ACD queue being monitored.This cause is reported from the perspective of the ACD structure or path being monitored. It occurs only for ACD structure monitors or ACD2 path monitors.
ACDCallQueuedFailed	A call was presented to an ACD2 group which was unavailable. This cause is reported only for ACD2 path monitors.
ACDCallQueuedOverflowed	A call has overflowed (via predictive overflow) from one ACD2 group to another in the path. This cause is reported only for ACD2 path monitors.
ACDCallReceived	An agent has received a call which was distributed to it via an ACD2 path.
ACDCallRedirected	A call to an ACD path was redirected by command such as from . This cause is reported only for ACD2 path monitors.
ACDRemoteCallQueued	
ACDRemoteCallDequeued	
ACDRemoteCallDistributed	
ACDHelpCall	This device is an ACD supervisor. It has just received a request for ACD Help from an ACD agent. The supervisor is now ringing. When the supervisor answers he silently monitors (is listen-only conferenced to) the agent's call.When the supervisor answers this device generates a CallConferencedEvent with a cause of ACDSilentMonitor. If the ACD Agent's line is being monitored it generates a CallConferencedEvent with a cause of ACDSilentMonitorEstablished.
ACDInterflowedDisconnected	A call interflowed (interflow timer expired before call could be delivered) out of the ACD path, and no alternate was programmed. This cause is reported only for ACD2 path monitors.
ACDInterflowedRerouted	A call interflowed (interflow timer expired before call could be delivered) out of the ACD path, to the programmed alternative answer point (Interflow Point). This cause is reported only for ACD2 path monitors.
ACDRadStarted	A RecordedAnnouncementDevice has begun to play its message for a caller.

Event Cause	Description
ACDRequest	A new call was presented to an ACD path for distribution. This cause is reported only for ACD2 path monitors.
ACDRequestDisconnected	A new call presented to an ACD path found the path unavailable, and no alternative answer point was programmed. This cause is reported only for ACD2 path monitors.
ACDRequestRerouted	A new call presented to an ACD path found the path unavailable, and was rerouted to the programmed Path Unavailable Answer Point. This cause is reported only for ACD2 path monitors.
ACDRequeueRequest	A call was presented to an ACD path for requeue (for example, following a prior delivery to an agent who failed to answer before timeout). This cause is reported only for ACD2 path monitors.
ACDRequeueRequestDisconnected	A call was presented to an ACD path for requeue, but the path was unavailable and no alternate was programmed. This cause is reported only for ACD2 path monitors
ACDRequeueRequestRerouted	A call presented to an ACD path for requeue found the path unavailable, and was rerouted to the programmed Path Unavailable Answer Point. This cause is reported only for ACD2 path monitors.
ACDSilentMonitor	Another device has invoked ACD Silent Monitor against this device. The call is now established. Both devices must necessarily be ACD devices. Initially this device must have been engaged in a call. ACDSilentMonitor is an ACD feature (also called a listen-in conference) which allows an ACD agent/ supervisor to quickly conference into an active ACD call. The conference created allows the supervisor to listen, but not talk, to the conferenced call. ACDSilentMonitor should not be confused with device monitors set with the SXMonitor routine.
ACDSilentMonitorDropped	An ACD Silent Monitor on a call has dropped, i.e. is no longer listening.
ACDSilentMonitorEstablished	This device invoked ACD Silent Monitor against another device and that monitor is now established (i.e., this device is now listening).
ACDSilentMonitorInvoked	This device invoked ACD Silent Monitor against another device. When the other device answers the switch conferences the call. Both devices must necessarily be ACD devices. Initially this device must have been engaged in a call. See also: ACDSilentMonitorEstablished, ACDSilentMonitor and Conferenced call causes.
ACDSilentMonitorMemberDropped	Indicates that any member (including supervisor) may have dropped from a silent-monitored conference. Also, silent-monitored conferences now report party lists, with silent monitor listed last.

Event Cause	Description
AlternateCallInvoked	This device has invoked alternate (swap) call. The previously connected device has been placed on consultation hold, and previously held device is now connected.
Answered	The other party has answered a call from this party.
AnswerInvoked	This party has answered the call.
CFADestinationBusy	This device has called a busy CFA destination.
CFBDestinationBusy	This device has called a busy CFB destination.
CFACallQueued	This party has queued onto its CFA destination, typically by invoking the call-waiting (camp-on) feature or by entering an ACD queue.
CFBCallQueued	This party has queued onto its CFB destination, typically by invoking the call-waiting (camp-on) feature or by entering an ACD queue.
CFNCallQueued	This party has queued onto its CFN destination, typically by invoking the call-waiting (camp-on) feature or by entering an ACD queue.
CallbackInvoked	This party has invoked the switch’s callback feature against another device. The callback feature is sometimes invoked against a device which is busy, has do-not-disturb activated or is not answering. The feature causes the switch to alert this device when the other device becomes available. When this device is alerted an event containing CallbackMatured cause is received by this device.
CallbackMatured	The switch is alerting this device that another device (against which this device previously activated the callback feature) is now available.This device need only answer the call[back] to redial the other device.
CallIsWaiting	A caller has camped-on to (activated call-waiting for) this party’s current call.
CallQueued	This party has queued onto its destination, typically by invoking the call-waiting (camp-on) feature or by entering an ACD queue.
Cleared	The other device invoked disconnect against a call with the monitored device.
ClearInvoked	The monitored device invoked disconnect against a previously active call.
Conferenced	A device (possibly this device) has been added to a conference call as a result of action taken by another device. If the conferenced device had setup an ActiveMonitor (ACD Help) or a SilentMonitor (ACD Listen-in), then an event carrying the Conference cause-code is sent to all conference members as soon as the ACD agent/ supervisor answers (for these features answer initiates a conference).
ConferenceCallHeld	With ConferenceFeatureEvent: A conference has been placed on Hard Hold

Event Cause	Description
ConferenceCallRetrieved	With ConferenceFeatureEvent: A hard-held conference has been retrieved.
ConferenceConsHeld	With ConferenceFeatureEvent: A conference has been placed on Consultation Hold (soft hold).
ConferenceConsRetrieved	With ConferenceFeatureEvent: A soft-held conference has been retrieved.
ConferenceEstablished	With ConferenceFeatureEvent: A Conference has just been established.
ConferenceInvoked	This device has formed a new conference or added a new member(s) to an existing conference.
ConferenceMemberAdded	With ConferenceFeatureEvent: A member has been added to the conference.
ConferenceMemberDropped	With ConferenceFeatureEvent: A member has been dropped from the conference.
ConferenceMonitorSet	With ConferenceFeatureEvent: SXConferenceMonitor() has just been invoked for a MCD, and a conference already in progress is being reported.
ConferenceRemoteUpdate	With ConferenceFeatureEvent: A remote conference device has transferred to a different DN.
ConfRetrieveInvoked	A conference on hold has been retrieved
ConferenceSplit	Another device split a conference (to which the monitored device was connected) into two calls.
ConferenceSplitInvoked	With ConferenceFeatureEvent: A three-party conference has been split. The monitored device splits a conference into two calls. This leaves a caller connected, and another caller on consultation hold.
ConferenceTransferred	With ConferenceFeatureEvent: A local conference member transferred the conference.
ConsCallInvoked	This device has placed its connected party on consultation hold. Its line is now available to make another call. Consultation hold is usually invoked in order to transfer or conference.
ConsHeldPartyDropped	This device had a party on consultation hold. That party has disconnected.
ConsHold	This device has been placed on consultation hold by another device.
ConsHoldInvoked	
ControllerCommsFailed	Communication with the device's primary or (or only, if the device is not Resilient) MCD has been broken. This cause will not be seen if SX_RESILIENT_MONITORS has not been specified in the applications SXInit() call; instead, a CommsErrorProc will occur.

Event Cause	Description
ControllerCommsRestored	In a Resilient environment (the application has specified SX_RESILIENT_MONITORS in its SXInit() call) this cause will be reported to a non-Resilient monitor when communication with its MCD is restored.
Delivered	A call from this device has been delivered to its destination. The other device is ringing. This device is waiting for the other device to answer.
DestinationBusy	This device has called a busy destination.
DestinationDoNotDisturb	This device has called a destination which on which do- not-disturb has been activated.
DestinationNotObtainable	This device has called a destination which is not currently obtainable. This can be caused by switch congestion, detection of a busy network or for other reasons.
DestinationUnavailable	The device has called a destination which is unavailable. This can be caused when the device is out of service.
DeviceInService	The device has been put back in service.
DeviceOutOfService	The device has been taken out of service.
DivertedAlwaysFrom	This device has received a call which was diverted away from its intended destination. The call-cause also indicates that the diversion occurred because the intended destination had activated call forward always. The destination which the calling device originally intended to contact is called the “original-destination”.
DivertedAlwaysTo	This device’s outbound call has been diverted from its intended destination to a third device. The diversion occurred because call forward always was active on the intended destination device.
DivertedNoAnswerAway	A call was diverted away from this device. The diversion occurred because this device did not answer and had call forward no answer activated.
DivertedNoAnswerFrom	This device has received a call which was diverted away from its intended destination. The diversion occurred because the intended destination did not answer and had activated call forward no answer.
DivertedNoAnswerTo	This device’s outbound call has been diverted from its intended destination to a third device. The diversion occurred because the intended destination did not answer and had activated call forward no answer.
DivertedOnBusyFrom	This device has received a call which was diverted away from its intended destination. The diversion occurred because the intended destination was busy and had activated call forward when busy.
DivertedOnBusyTo	This device’s outbound call has been diverted from its intended destination to a third device. The diversion occurred because the intended destination was busy and had activated call forward when busy.

Event Cause	Description
DivertedPickedupAway	A call ringing at this device has been answered by another device which invoked the call pick up feature.
ErrorDetected	An attempt to place a call failed due to an error condition.
ExtensionInUse	This device (line) is being used by another device. This cause occurs when a device, which is not the monitored device, is using this extension. This can occur when the other device makes, answers, retrieves, etc. a call on the monitored extension.
GlobalCallIDUpdated	This event occurs when the monitored device is connecting or connected to a remote party via a trunk, and the remote party Global Call ID information changes.
HandOffPush	This device has received a call which was pushed to another PRG member.
HardHeldPartyDropped	This device had a party on hard hold. That party has disconnected.
HardHold	This device was placed on hold by another device.
HardHoldInvoked	This device placed its connected party on hold.
HardHoldRetrieved	
Intrusion	This party and another party are engaged in a call, and a third party has intruded into that call. The MCD feature which allowed this is called executive over-ride (also barge). The result is a conference call between all three parties. Use of the executive override feature requires that the intruder have special privileges assigned to him or her on the MCD .
IntrusionInvoked	This party has intruded into a call between at least two other parties. (see also Intrusion call cause).
InvalidAccountCode	This party provided an invalid account code.
MemberAnswered	
MonitorSet	A device monitor was set against this device. This cause is returned in a MonitorSetEvent to the invoker of the SXMonitor routine. The event reports the initial device state.
NetworkBusy	This will be reported when an external call fails and a busy tone is detected, if tone detection is available and the switch CDE form specifies the action “busy tone and release” for busy tone.
NetworkCongestion	An attempted network call has failed due to a timeout.
NetworkNotObtainable	This will be reported when an external call fails and a reorder tone is detected, if tone detection is available and the switch CDE form specifies the action “reorder tone and release” for reorder tone.
NewCall	This device has just received a new call. This device is now ringing.

Event Cause	Description
NewCallInvoked	This device is initiating a new call.
NoAnswer	This device was attempting to make an outgoing call. The call failed because the destination device did not answer within the MCD allowable time period.
NoSuchNumber	This device has attempted a call to an invalid number.
ONSCallOriginated	A request to make an outbound call from an ONS phone has caused that phone to ring, signalling the user to pick up the handset so that the call can proceed.
OtherDeviceCleared	This device was engaged in a call with another device, when the other device disconnected. This call-cause is reported from CallOriginatedEvent.
OtherDeviceDropped	This device was engaged in a call with another device, when the other device disconnected. This call-cause is reported from DeviceDroppedEvent.
Pickup	A call from this device to another device was answered by a third device, when the third device invoked the MCD call pickup feature.
PickupInvoked	This device answered a call which was ringing at another device by using the MCD call pickup feature.
PrivilegeViolation	This device has attempted to invoke a function which is not available to it.
QueuedPartyDropped	This device had a call waiting (a caller had camped on or been inserted in an ACD call queue). That call has cleared.
RADUnavailable	A monitor on an IP RAD is reporting that the RAD has passed into an unavalable state and must be Cleared (SXClearCall).
RecallHeldParty	This device’s held party is recalling. This occurs when the held party has been on hold for an excessive period.
RedirectedAway	A caller for this device has been redirected to another device.
RedirectedFrom	This device has received a call which was redirected away from its intended destination. The call-cause also indicates that the diversion occurred because this device lacked sufficient privileges to call the other device.
RedirectedHandoff	An MCD acting as a Secondary MCD for a Resilient device has determined that communication with the primary MCD has been re-established, and registration for the device has been passed back to that MCD.
RedirectedOnError	The MCD can be programmed so that calls which fail are rerouted to an error handling number (such as an attendant). In the presence of such programming, failed calls originating at the monitored device result in a CallDeliveredEvent with a RedirectedOnError cause (possibly in addition to a CallFailedEvent). In the absence of such programming only a CallFailedEvent is reported.

Event Cause	Description
RedirectedTo	This device's outbound call has been redirected from its intended destination to another device, because of MCD system programming (such as forced call forward).
RemotePartyUpdated	The information concerning a remote party has been updated due to a change, such as transfer to another DN, or an enhancement, such as remote party name being provided at some time after the original connection. The update is provided via a macro such as SX_ConnectedDevice.
Retrieve	This device was retrieved from hold by the other device.
RetrieveInvoked	This device retrieved its held party.
RoutingDeviceCallAbandoned	When a call queuing on a routing device is abandoned by its caller.
RoutingDeviceCallOverflow	When a call is redirected from a routing device to a new destination provided by HCI Host, or a call is rerouted by default (rerouted on no-answer)from a routing device to a new destination programmed in the MCD.
RoutingDeviceCallQueued	This happens when a new call arrives and queues on a routing device.
RoutingDeviceCallRerouted	If a call is rerouted by MCD system rerouting options (due to always_reroute or reroute-on-DND) from a routing device to a new destination programmed in the MCD.
RoutingDeviceCF	When a call is forwarded via the call forwarding feature to the routing device. There are three conditions for call forwarding: 1. RoutingDeviceCFBUSYCallQueued (only forward on busy). 2. RoutingDeviceCFNOANSCallQueued (only forward call when there is no answer). 3. RoutingDeviceCFALWAYSCallQueued (always forward call).
RoutingDeviceCFBUSYCallQueued	
RoutingDeviceCFNOANSCallQueued	
RoutingDeviceCFALWAYSCallQueued	
RoutingDeviceRadStarted	This happens when the monitored Rad of the routing device starts.
RoutingDeviceReroutedCallQueued	When a call is rerouted via system rerouting features to the routing device.
RoutingDeviceTransCallQueued	When a call is transferred to the routing device.
SecondaryControllerCommsFailed	A Resilient device has no communication with either its primary MCD controller or its secondary MCD controller(s).

Event Cause	Description
SecondaryMonitorSet	ControllerCommsFailed – Communication with the device’s primary or “home” MCD has been broken. SecondaryMonitorSet – Communication with the device’s secondary MCD is up, and the monitor on the device is now established there. SecondaryControllerCommsFailed – There is no communication with either the primary MCD controller or secondary MCD controller(s). RedirectedHandoff – An MCD acting as a Secondary MCD for the device has determined that communication with the primary MCD has been re-established, and registration for the device has been passed back to that MCD.
SupervisedTransfer	This device was connected to another party. The other party transferred the call to a third party.
TapCall	This device, probably an IVR or similar device, has received a call which has been “tapped” from an ACD queue by the use of the SXTapCall() command.
ThisDeviceDropped	This device disconnected.
TransferInvoked	This device has transferred its connected party to another destination. This device is now idle or dialing.
TransferRecallNoAnswer	This device transferred a call to another device. The other device did not answer and the call has now recalled to this device.
TransferRecallOnBusy	This device transferred a call to a another device. The other device was busy and the call has now recalled to this device.
TransferredCall	This device has received a call which has been transferred from a another device. The transfer was unsupervised.
TransferredToBusy	This device was connected to another device. The other device transferred the call to a third device which was busy. This device is now waiting for (camped-on to) the third device.
TrunkDigitsOutpulsed	The outpulsing of all digits for a trunk is complete.
TrunksBusy	An outbound call could not be completed because all available trunks were already in use.
UnsupervisedTransfer	This device was connected to another device. The other device transferred the call to a third device.
WaitingCallRinging	A call which was waiting on the monitored device is now ringing through.
WorkTimerExpired	The ACD work timer has expired or has been canceled.
CallDistributed	
CallAnswered	
CallDisconnected	
CallOverflowed	

Event Cause	Description
CallWaitingPartyDisconnected	
PeerOutOfService	
PeerInService	
EhduAccess	
EhduCallback	
ClosingCallServer	

Call Control Service Event Data – Standard Type

Each event includes data. The data available for a particular event can be broken down as follows:

- **Common Event Data** – data that is common to all events. For detailed information, see Common Event Data Reference.
- **Class-Specific Data** – available for all events of a specific event class. For detailed information, see Class-Specific Data Reference.
- **Event-Specific Data** - available and valid only for certain events types. For detailed information, see Event-Specific Data Reference.

Common Event Data

The common parameters valid for all events are shown below:

Data	Description
EventType	Identifies the event that occurred. This value is used to identify particular event data to be obtained.
EventClass	CallEventClass FeatureEventClass SystemEventClass
EventTime	The time at which the event was received.
objectID	Identifies the monitored object generating the event.
DeviceNamePrivacy	Privacy status of a party involved in a two-party call (**Note**: Name and / or number may be suppressed by MCD Class of Service settings). TRUE (1) indicates that Privacy is "on" for that device.
Conference Member Name Privacy	Privacy status of a member of a Conference call. (**Note**: Name and / or number may be suppressed by MCD Class of Service settings). TRUE (1) indicates that Privacy is "on" for that device.

Class-Specific Data

In addition to the common event data above there are specific event data that is common to each class.

CallEventClass Data

All events of the CallEventClass have additional common data. The data is:

- LocalCallId
- GlobalCallID
- Cause
- State
- FeaturesAllowed
- Device information (calling, called, routing, etc.)

Data	Description
LocalCallId	Identifies the call that was affected. In the case of a ConferenceFeatureEvent, this is a number assigned by the MCD to track a particular conference. **Note 1**: Call events reported by a monitor on an IP RAD do not provide Call ID information. **Note 2**: In general, the Call Id reported for other devices stays the same throughout the life of the call; this facilitates the coordination of events and calls. There are two exceptions (when the MCD is a MCD), however, and applications which depend on Call ID will need to handle them: ***Hard Hold***: When A is talking to B, and puts B on hard hold, and then retrieves it: B's Call ID is unaffected. But A's Call ID goes to zero (0) upon invoking the hard hold, and then goes to a new number upon retrieving the hard-held call. Application handling: After the retrieval, A will report its new Call ID and B's old Call ID (via SX_DeviceCallid(111evice), where 111evice points to B) and B also reports its old Call ID and A's new Call ID. An application which monitors either end can use the consistency of B's Call ID to determine that this is the same call. ***ACD Requeuing***: A change in the Call ID may also be expected when a call comes to a path, and then fails to be delivered and so get requeued to the path. There may be changes in other scenarios, but an application can test as to whether it is the same call by checking the Call ID at the other end of the call (again, via SX_DeviceCallid(111evice), where 111evice points to the other device). If the Call ID for the other device is unchanged, then it is the same call. Hard Hold and ACD requeuing are the most likely scenarios in which a change of call ID may be seen. But a great deal of testing would be required to ensure that there are no changes in any other scenarios, and this list could possibly be altered/added to by ongoing development in the switch. So it will be advisable for an application to check as in the preceding paragraphs, whenever there is a change, and it is required to determine whether it might be the same call.
GlobalCallId	A system wide unique call ID that can be used to track a call through the MCD cluster.
EventCause	Indicates why the event was generated. Causes vary depending on the event type. Refer to the event reference details for type-specific event causes. Call causes are described with respect to the monitored device (i.e., the receiver of the event). Within no distinction is made amongst events which are caused by the MCD, by action on the physical set or by invocation of routines. See the Event Cause Reference section for a detailed description of all event causes.

Data	Description
EventState	Indicates the current state of feature, call, or device. The possible states are dependant on the event type. Refer to the event state reference section for details of type-specific event states.
FeaturesAllowed	Indicates which Call Manipulation operations are now allowed.. AlterWorkTimer AlternateCall AnswerCall AssignCallerId CamponCall CancelConsultationCall ClearCall NewCall ConferenceCall ConsultationCall HoldCall IntrudeCall ListenCall OutpulseDigits RedirectCall RetrieveCall SendCallbackMessage SetAccountCode SetCallMeBack SplitConferenceCall TradeCall TransferCall TapCall
Device information	Identifies the device causing the event, provided in CallEventClass events. Identifies the calling device, provided in CallEventClass events. Identifies a called device that passed the call to the called device, provided in CallEventClass events

FeatureEventClass Data

All events of the FeatureEventClass have additional common data. They are:

- EventFeature
- EventState
- DeviceNumber

Data	Description
EventFeature	The particular features to which the event refers. Features for which this event can apply are listed in the table below:

Feature	Description
ACDAgentFeature	An ACD Agent logged in or logged out
AutoAnswerFeature	Auto answer feature was enabled or disabled
DoNotDisturbFeature	Do not disturb was enabled or disabled
DoNotDisturbAll	All “Do not disturb” has been disabled
MsgWaitingIndicator	Message waiting was enabled or disabled
MsgWaitingIndAll	All message waiting has been disabled
InServiceFeature	Device has gone in or out of service
MakeBusyFeature	ACD device make busy feature was enabled or disabled
MakeBusyStateFeature	A walkaway code, indicating a reason for the make busy, has been set. Use the SX_MakebusyCode(eventp) macro to obtain the code.
CFAlways	Call forwarding always feature was enabled or disabled
CFBusyExternal	Call forwarding busy external feature was enabled or disabled
CFBusyInternal	Call forwarding busy internal feature was enabled or disabled
CFNoAnswerInternal	Call forwarding no answer internal feature was enabled or disabled
CFNoAnswerExternal	Call forwarding no answer external feature was enabled or disabled
CFNoAnswerBusy	Call forwarding no answer and busy features were enabled or disabled
CFAllModes	All call forwarding has been disabled
JoinGroup	
LeaveGroup	
JoinAllAcdGroups (Adv)	
LeaveAllAcdGroups (Adv)	
AcdAgentBlocked (Adv)	
Acd2LoginNoLicenses (Adv)	
Activate	
Forward	
UserId	
HotDeskUser	
Conference	

Data	Description
EventState	The current state a feature with regards to the particular monitored device. This can be FeatureEnabled FeatureDisabled
EventDeviceNumber	The device on which the event occurred.

Event-Specific Data

The data specific to each event is based on the event and state. The device data for a phone includes the Device, Name and Number. For example, the data in the CallReceivedEvent. The OIG WSDL defines what data is provided in what events. See Appendix E on WSDL generated definitions.

OIG Tools

OIG sample aplications will be provided to test OIG functionality. The Tools will be offered on the Mitel MSA web portal.

Appendix A: Design and Error Handling Guidelines

Here are some guidelines when developing applications to use the OIG.

Topic
Session Communication Security
ANI and DNIS Information in Events
DTMF Generation
DTS programmed on a phone key
Error Reporting
Device Data
Identifier for Phone Non-prime Line
Monitoring IP Trunks
Monitoring Limits
Redirecting Call on Line Appearance
General Notes

Session Communication Security

Applications use https to communicate with the OIG. The OIG uses a Secure Socket Layer (SSL) to communicate with an MCD.

ANI and DNIS Information in Events

- Automatic Number Identification (ANI): caller information
- Dialed Number Identification Service (DNIS): called party information

When a call is transferred from any device, the ANI of an incoming PSTN call is preserved but the DNIS is lost. ANI is provided across MSDN trunks with appropriate Class of Service options with Advanced Analog Networking. ANI is not provided across IP Trunks or Non-MSDN trunks. OIG does provide ANI on T1 trunks if the call is transferred to the ACD path using a flash. Some examples are listed here:

Example 1: Remote Transfer (Unsupervised) –

Scenario: Call from MCD **A** to MCD **B**. Phone on **B** transfers incoming call to another phone on **B**. Monitor on second phone on MCD **B**.

Events:

a. CallEstablishedEvent / NewCall / Nil
b. CallEstablishedEvent / NewCall / TransferredCall
c. CallRetrievedEvent / AnswerInvoked / TwoPartyCall
d. ANI is reported as sub DN for events b and c.

Example 2: Remote Transfer (Supervised)

Scenario: Call from MCD **A** to MCD **B** where phone on **B** transfers incoming call to another phone on **B**. Monitor on second phone on MCD **B**.

Events:

a. CallEstablishedEvent / NewCall / Nil
b. CallRetrievedEvent / AnswerInvoked / TwoPartyCall

NO ANI is reported in this scenario

Example 3: Remote Transfer Back to originating switch (Supervised)

Scenario: Call is made to set **A** in switch **X**. Set **A** transfers the call to Set **B** in switch **Y**. Set **B** then transfers the call back to set **C** in switch **X**. Monitor on phone **C** on MCD **X**.

Events:

a. CallEstablishedEvent / NewCall / Nil
b. CallRetrievedEvent / AnswerInvoked / TwoPartyCall
c. CallRetrievedEvent / Transfer / TransferredCall

ANI is reported for event c as main DN.

Example 4: Remote Transfer Back to Originating switch (Unsupervised)

Scenario: Call is made to set **A** in switch **X**. Set **A** transfers the call to Set **B** in switch **Y**. Set **B** then transfers the call back to set **C** in switch **X**. Monitor on phone **C** on MCD **X**.

Events:

a. CallEstablishedEvent / NewCall / Nil
b. CallEstablishedEvent / NewCall / TransferredCall
c. CallRetrievedEvent / AnswerInvoked / TwoPartyCall
d. CallRetrievedEvent / Transfer / TransferredCall

ANI is reported for events b and c as sub DN.
ANI is reported for event d as main DN.

DTMF Generation

An application CAN NOT use outPulseDigits to send DTMF digits to IP phones. The MCD performs the outpulsing but only to trunks and DNI connected phones.

DTS programmed on a phone key/button (Advanced)

An application can not monitor a phone key with DTS. An application can use trunk monitoring to monitor DTS.

Error Reporting

OIG operations return success or an error code on failure. The OIG groups errors into general, communication, and operation-specific errors. **General** errors indicate that an operation cannot be performed. Typically the application is trying to perform an unsupported or unavailable operation. **Communication** errors occur due to a failure in the communication path between the application and the MCD. Some communication errors are recoverable and some are not. For recoverable communication errors, the application should retry the operation. For non-recoverable errors, refer to Call Control Service Error Definition. **Operation-specific** errors are those relating to the failure of an individual operation. These errors include attempts to monitor or manipulate unknown phones, and attempts to perform operations that are not valid. Typically these errors indicate an application programming error.

Device Data

The OIG events contain device data related to a phone within a call event. This information includes the name, number, and originally-dialed digits for both internal and external devices. For external devices this information also includes the name and number of the facilitating trunk. The device number contains Automatic Number Identification (ANI) when ANI is provided by the MCD. If available, ANI is provided in both the CallReceivedEvent and the CallEstablishedEvent. The originally-dialed digits associated with the device contain Dialed Number Information Service (DNIS) when DNIS is provided by the MCD. If available, DNIS is provided in both CallReceivedEvent and CallEstablishedEvent.

- DeviceName
- DeviceNumber
- DeviceTrunkNumber
- DeviceTrunkName
- DeviceANI
- DeviceDNIS
- DeviceDomain
- DeviceNetworkExt

The OIG events also include Device domain information:

- InternalDeviceType
- ExternalDeviceType

Events also provide:

- FirstName
- LastName
- MiddleName
- MainDn
- SubDn

Identifier for Phone Non-prime Line

When a network call is answered by a monitored phone on a local MCD, the event information reported is dependent on the MCD COS option **Non-Prime Public Network Identity**. If Non-Prime Public Network Identity is disabled, a calling non-prime line is identified by the DN of physical phone prime line where the line is calling from. If Non-Prime Public Network Identity is enabled, then a calling non-prime line will be identified by the DN of the non-prime line (i.e., not the DN of physical phone prime line).

Monitoring IP Trunks

IP trunks are not monitorable. If an IP trunk is involved in a call with a monitored phone, a trunk number is provided for the trunk in the monitor call status event. The reported trunk number is not unique and may actually be the same for two different trunk calls in different call status events.

Monitoring Limits

Maximum Of 16 Lines

An application can only monitor 16 line appearances on a phone as the application will only receive information for the first 16 line appearances.

Several Monitors On Same Device

Setting several monitors on the same DN from the same application is not supported.

Redirecting Call on Line Appearance

To redirect a call away from a line appearance of a prime line with a phone monitor, the MCD option 'Display Caller ID on multicall/keylines' must be set to **Yes** for the forward to work. If the application redirects the call to an invalid number an error is generated. A feature not allowed error normally means a MCD COS option violation.

General Notes

1. An application can either poll for events or it can receive events asynchronously. A single event cannot be obtained both through polling and async. Therefore, if an application is receiving an event asynchronously, then it won't receive that event if the application polls for that event.
2. Applications should ignore events the application is not expecting. The application should not stop working when an unexpected event is received.
3. EHDU / EHDA hard hold not supported - When an external hotdesking device places a call on hard hold, the resulting Call Held Event has a cause of ConsHoldInvoked instead of the expected HardHoldInvoked.
4. LocalCallID - LocalCallID is for device side of a call (call reference). If a call has two parties, each party has a call reference to the same global call id (GCID) but a unique localCallID (only unique on a single MCD). In general, the LocalCallID reported for a phone stays the same throughout the life of the call; this facilitates the coordination of events and calls on a single MCD. There are two exceptions:

   Hard Hold: When A is talking to B, puts B on hard hold, and then retrieves the call to B: B's LocalCallId is unaffected. A's LocalCallId goes to zero upon invoking the hard hold, and then goes to a new LocalCallId upon retrieving the hard-held call. After the retrieval A reports a new LocalCallId and B's old LocalCallId. B also reports its old LocalCallId and A's new LocalCallId. An application which monitors either A or B can use the consistency of B's LocalCallId to determine that this is the same call on a single MCD.

   ACD Requeuing: A change in the LocalCallId may also be expected when a call comes to a ACD path and then fails to be delivered and gets requeued to the ACD path. An application can test for the same call by checking the LocalCallId at the other end of the call. If the LocalCallId for the other device is unchanged, then it is the same call.
5. If a phone is left in failed state for 45 seconds (i.e., not cleared using clear call operation) after a make call fails, an outofservice event is followed by an inservice event. The MCD resets a phone left in failed state for 45 seconds.
6. RedirectCall on an ACD2 and ACD Express path follows a call forward no answer (CFNA) model. Meaning the caller remains queued to the current destination if the redirection is unsuccessful because the redirect destination cannot be seized. Redirection does not invoke campon when the nominated destination is busy. Only when the nominated destination is an ACD2 or ACDX path or a Ring group will the call automatically campon. In these cases the destinations do not return busy. ACD Redirection to a busy destination returns feature not allowed, while ACD Interflow to a busy destination will auto-campon. Redirection from a "HCI Reroute Hunt Group" to a busy destination will auto-campon to the busy destination.
7. When a call is auto recall to an ACD Agent the Call Received Event cause is RecallHeldParty.
8. Consider monitoring Hotdesk User 7500 on primary MCD 1 (with secondary MCD 2), physical phone 6001 on MCD 2 (with secondary MCD 1) and physical phone 6500 on MCD 1 (with secondary MCD 2); Application creates a monitor on 7500 on MCD 1. When 7500 is logged into 6500 (using the phone directly), the application receives an in service event for 7500. When 7500 is logged out of 6500, the application receives an out of service event for 7500. When 7500 is logged into 6001 (7500 and 6001 are on two different MCDs), the application does not receive an event. When 7500 is logged out of 6001, the application does not receive an event. Solution is to use featureMonitor operation on Hotdesk User 7500 to detect when a hotdesk user logins

in and out of a phone. When a feature event is generated for hot desk user login in or out, the application can use queryFeature to determine the hotdesk user that logged in and out and on what phone DN.

9. To distinguish the difference between a “PSTN to phone call” (external trunk on the public network) from a “Remote phone on another MCD”, use the calling-line-category information to distinguish the two types of calls. CLC_ORD indicates when the call is from a remote phone. CLC_DASS2 / CLC_PSTN indicates a PSTN call.
10. To allow a application to retrieve group presence on a DN ensure MCD COS settings "Group Presence Control" and "Group Presence Third Party Control" are enabled.
11. An application needs to use the Global Call ID in events to relate a call that is parked and then retrieved. The events reported for the call-park scenario do not indicate explicitly the calls are related to the call-park feature. The global call IDs reported do provide correlation for the parked call and the retrieved call (the Global Call ID in the events are the same). The application should not depend on LocalCallID reporting, which is different from the global call id reporting.
12. The OIG presents different event flow when monitoring devices on one MCD compared to monitoring devices on different MCDs in the same system. Example, the events presented at the holdee for a local recall scenario (i.e., holder gets recalled when both holder and holdee on same MCD).

    Local Recall

    ```
    EVENT: 10.37.196.22-1101   CallHeldEvent  HeldState  ConsHold
    EVENT: 10.37.196.22-1101   CallHeldEvent  HeldState  HardHold
    EVENT: 10.37.196.22-1101   CallDeliveredEvent  DeliveredState  Delivered
    EVENT: 10.37.196.22-1101   CallEstablishedEvent  EstablishedState  Answered
    ```

    The events presented at the holdee for a remote recall scenario (i.e., holder gets recalled when holder and holdee on different MCDs).

    Remote Recall

    ```
    EVENT: 10.112.95.39-1502   CallHeldEvent  HeldState  ConsHold
    EVENT: 10.112.95.39-1502   CallDeliveredEvent  DeliveredState  Delivered
    EVENT: 10.112.95.39-1502   CallEstablishedEvent  EstablishedState  Answered
    ```

    The events presented at the holdee for a local retrievel scenario (i.e., holder retrieves holdee when holder and holdee on same MCD).

    Local Retrieve

    ```
    EVENT: 10.37.196.22-1101   CallHeldEvent  HeldState  ConsHold
    EVENT: 10.37.196.22-1101   CallHeldEvent  HeldState  HardHold
    EVENT: 10.37.196.22-1101   CallEstablishedEvent  EstablishedState  Retrieved
    ```

    The events presented at the holdee for a remote retrievel scenario (i.e., holder retrieves holdee when holder and holdee on different MCD).

    Remote Retrieve

    ```
    EVENT: 10.112.95.39-1502   CallHeldEvent  HeldState  ConsHold
    EVENT: 10.112.95.39-1502   CallEstablishedEvent   EstablishedState  Retrieve
    ```

Appendix B: Operation Considerations

This section describes important operation and design considerations related to the OIG Call Control Service (CCService).

1. Session – An application must login to the OIG to use the Call Control Service. There are two types of login (Standard and Advanced). Standard login uses a “localPassword” defined in the OIG. Standard login provides access to standard services. Advanced login uses a “localPassword” and a “Mitel Certificate”. An Advanced application must also authenticate data with the OIG using the OIG authenticate operation. Advanced login provides access to standard and advanced services. The OIG Session management provides a reset session timer operation to keep the application communication session alive after successful login. Session management allows an application to request service versions provided by a specific OIG (i.e., version of MCDs connected, overall OIG version, version of each OIG service).
2. Event Time Stamps (Future) – The OIG CCService events sent to applications are time stamped with two times; 1) OIG time, and 2) MCD time.
3. Opening an MCD connection – An application requests the OIG to open a connection to a MCD. If two applications request the same connection to the same MCD only one connection will be opened from the OIG to the MCD.
4. Closing an MCD connection – An application does not request an OIG to disconnect a MCD. All successful MCD connections remain open until the OIG is shutdown.
5. CCService Logs – The OIG Admin web interface provides a view of all logs created on the OIG. The OIG is installed using MSL. The MSL server manager also allows an admin to view all logs created by the OIG on the MSL server.
6. AMC Licensing – The OIG is licensed using the Mitel AMC. Refer to the OIG Installation & Maintenance Guide for detailed licensing information.
7. Engineering Limits – The OIG CCService allows a maximum of 5000 monitors on one MCD. The OIG CCService allows a total of 50000 monitors for all MCDs. Refer to the OIG Engineering Guidelines for more information.
8. Monitor Multiplexing – When two or more OIGs request phone monitors on the same phone on one MCD, only one monitor is created in the MCD. When two or more applications ask for phone monitors on same phone, the OIG CCService only requests one monitor from the MCD. (Setting several phone monitors on the same phone from the same application is not supported).
9. IP Resiliency (Future) – The OIG always requests resilient behavior from a MCD. If the primary MCD is off-line, the OIG informs the application that initial connection failed and then attempts to connect to the secondary MCD. The OIG will continually search for a working MCD until an application logs out. For non-resilient monitors, the OIG informs an application when communication with a MCD has dropped. The OIG indicates an OutOfService Event for non-resilient monitors. The OIG indicates InService Events when a MCD connection is restored.

Appendix C: Call Flow Exceptions for Call Park and Paging

Loudspeaker Page

User lifts handset, dials FAC (Feature Access Code) for page, and speaks his message over the loudspeaker. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	NewCallInvoked
CallDeliveredEvent	DeliveredState	RedirectedTo (see Note1)
CallEstablishedEvent	EstablishedState	Answered

Note 1: A cause of **RedirectedTo** is not normally expected immediately after the CallOriginatedEvent. No original destination is provided (as it would be with a DivertedAlwaysTo cause)

Direct Page

User lifts handset, dials DirectPage FAC plus target DN, and leaves his message. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	NewCallInvoked
CallEstablishedEvent	EstablishedState	Answered

Note: There is no CallDeliveredEvent which normally occurs.

Call Park and No Page

User presses Trans/Conf on an incoming call, then dials the CallPark FAC plus target DN. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	ConsCallInvoked
CallClearEvent IdleState	IdleState	CallClearedInvoked (See Note 2)

Note 2: When user hangs up after leaving message. This is a normal sequence, but the app has no indication that a call was parked.

Call Park and Direct Page

User presses Trans/Conf on an incoming call, then the DirectPage FAC plus target DN. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	ConsCallInvoked
CallDeliveredEvent	DeliveredState	RedirectedTo (See Note 3)
CallEstablishedEvent	EstablishedState	Answered

Note 3: A cause of RedirectedTo is not normally expected immediately after the CallOriginatedEvent. No original destination is provided.

Call Park and Loudspeaker Page

User presses Trans/Conf on an incoming call, then dials LoudSpeaker FAC plus target DN, and speaks his message over the loudspeaker. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	ConsCallInvoked
CallDeliveredEvent	DeliveredState	RedirectedTo (See Note 4)
CallEstablishedEvent	EstablishedState	Answered

Note 4: A cause of **RedirectedTo** is not normally expected by apps immediately after the CallOriginatedEvent. No original destination is provided (as it would be with a DivertedAlwaysTo cause).

Call Park Retrieve

User lifts handset, and dials FAC or presses a programmed softkey to pickup the parked call. A monitor on the user's handset will report the following:

Event	State	Cause
CallOriginatedEvent	OriginatedState	NewCallInvoked
CallEstablishedEvent	EstablishedState	Answered

Note: there is no **CallDeliveredEvent** such as normally occurs.

Appendix D: SIP Device Support

The OIG offers limited support for SIP devices. Monitoring and control of SIP devices is similar to monitoring and control of ONS devices. The event flow from a SIP device monitor is different than the event flow from a monitor on a Mitel MiNET device. Application developers should confirm event flow from a SIP device monitor through testing.

Some areas where a SIP device works differently:

1. Hotdesk login to a local SIP phone is not supported.
2. OIG Call Control Service cannot answer or clear a call to a SIP device.
3. Event flow with SIP device as External Hotdesk User is different.
4. Event flow for call features like hold, transfer and conference are different.

Table 1 provides a list of features that work, but work differently in many cases.

Table 1: Device Monitoring

Function	Supported for SIP	Supported for EHDU
SXAlternateCall (Swap)	Yes	Yes
SXAnswerCall	No	No
SXAssignCallerId	No	No
SXCamponCall	Yes	Yes
SXCancelConsCall	Yes	Yes
SXClearCall	Yes	Yes
SXConferenceCall	Yes	Yes
SXConsultationCall	Yes	Yes
SXHoldCall	Yes	Yes
SXIntrudeCall	Yes	Yes
SXListenCall	No	No
SXMakeCall	Yes	Yes
SXNewCall	Yes	Yes
SXOutpulseDigits	Yes	Yes
SXPickupCall	Yes	Yes
SXRedirectCall	Yes	Yes
SXRetrieveCall	Yes	Yes
SXSendCallbackMessage	Yes	Yes
SXSetAccountCode	Yes	Yes

Function	Supported for SIP	Supported for EHDU
SXSetCallMeBack	Yes	Yes
SXSplitConferenceCall	Yes	Yes
SXTapCall	No	No
SXTradeCall	Yes	No
SXTransferCall	Yes	Yes

While the use of stimulus messages can be used to update the displays, states, and LEDs of a MiNET device, this is not possible with SIP and EHDU devices. As such the SIP and EHDU device states may not match the state of the call process. To explain this in more detail, the following sections describe the end user behavior experience when each API is invoked.

Command Descriptions (Supported)

SXAlternateCall

SXAlternateCall is used to alternate between the active call and the call on consultation hold. The active call is placed on consultation hold and the call on consultation hold is made active.

SXCamponCall

SXCamponCall allows a caller to a busy destination to camp on to the busy device. The busy device user may trade to the caller or the camped on (queued) call will ring the busy device as soon as busy device ends the call.

SIP devices get auto campon to a busy destination but as a part of a Callback feature, auto campon will be disabled. In this case the SIP device user must use SXCamponCall to campon to a busy device.

SXCancelConsCall

SXCancelConsCall clears the active call at the specified device and reconnects it to the party on Consultation Hold.

SXClearCall

SXClearCall releases the specified call from the designated device. A SIP device will go into idle state. An EHDU will receive dial tone and be able to make another call.

SXConferenceCall

SXConferenceCall merges two calls into a single conference call. The SIP or EHDU users will hear the conference tone when conference connection has been made.

SXConsultationCall

SXConsultationCall places the active call on consultation hold. If a directory number is provided, a second call is placed to the specified new party.

SXHoldCall

SXHoldCall temporarily suspends communication on the active call at the specified device. This function places a call on hard hold. The SIP and EHDU will hear silience during the period in which they have the call on hard hold.

SXIntrudeCall

SXIntrudeCall allows the caller to conference into an existing conversation after an attempt to call a busy destination. The 3300 must be programmed to allow the caller this privilege.

SXMakeCall

SXMakeCall makes a call from the device identified by a monitor handle or a directory number to any DN recognized as valid in the 3300 numbering plan. It is also possible to set a forced account code with SXMakeCall; see the notes at the end of this section for details.

When SXMakeCall is used for SIP devices, first the calling party starts ringing and when it answers, then the called party is seized and it starts ringing. When called party answers, the call gets established between the two parties. The expected behavior of SXMakeCall API for SIP devices will be same as it is for ONS sets.

SXNewCall

SXNewCall ends a consultation call without retrieving the held call, making it possible for the user to first consult someone else.

SXOutpulseDigits

SXOutpulseDigits does not work when monitored device is connected to an IP phone. The command will indicate success but no DTMF tones are sent to the device. This command does work for DNI and ONS type devices as well as TDM trunk calls.

SXPickupCall

SXPickupCall picks up a call which is ringing at another device. SXPickupCall will first ring SIP device and when answered, the call ringing at the other device gets picked up.

SXRedirectCall

SXRedirectCall redirects (transfers) a ringing call to another device and the SIP or EHDU will cease to ring.

SXRetrieveCall

SXRetrieveCall reconnects an existing held call at the specified holding device. The call was previously suspended using SXHoldCall or the 3300 call hold feature.

SXSendCallbackMessage

SXSendCallbackMessage allows a caller to leave a message at the called set indicating who called. The called party could have been busy on the phone at the time or unable to answer.

SXSetAccountCode

The SXSetAccountCode routine is used to assign a non-verified account code to an established call on any monitored phones. When attempting to use the SXSetAccountCode function with an invalid verified account code, the SXSetAccountCode function does not verify the account code but returns command success even though account code is not valid (i.e., 3300 does not check the account code).

SXSetCallMeBack

SXSetCallMeBack allows the caller to have the 3300 ring the caller with a distinctive pattern as soon as activity at the destination indicates the availability of the callee, and then ring the callee as soon as the caller answers this distinctive ring. This function can be invoked when an attempt to make a call has failed due to the destination being busy or not answering.

SXSplitConferenceCall

Splits a conference into two calls and places one party into the soft held position.

SXTradeCall

When invoked the active call is placed on soft hold and the waiting party is connected to the SIP device. This command is not supported for EHDU as Campon is not supported on the EHDU.

SXTransferCall

The currently connected party is transferred to the party in the soft held position. The SIP device will return to idle state. An EHDU will receive dial tone and be offered the chance to make another call.

Command Descriptions (Unsupported)

SXAnswerCall

Answering an incoming call using is performed by faking keystrokes from a MiNET device. Since this type of control for SIP device is not possible this command is not supported.

SXAssignCallerId

The SXAssignCallerId routine is used to assign the identity of a caller (i.e., name and number) to an incoming or outgoing call from a trunk so that the new identity is displayed on the set. The trunk must be currently participating in the call, and must be monitored by the application. After this assignment occurs the caller’s identity will be associated with the trunk for the life of the call and will be displayed on the sets but not through the events.

This routine is useful when the caller’s identity can be obtained either manually by an attendant or automatically by an interactive voice response (IVR) machine or from ANI supplied by the CO trunk. The IVR machine could for example ask the caller to enter their customer number through their DTMF keypad.

SXListenCall

SXListenCall requests an ACD supervisor to listen-in to an active ACD call. This is not supported functionality on SIP/EHDU.

SxTapCall

Connects a call which is waiting on an ACD queue to a B-channel trunk on Mitel's AFC card, without removing the call from the ACD queue. This is not supported from SIP/EHDU.

SIP Device Display Updates

ANI/DNIS Display on SIP phones

ANI/DNIS information has precedence over caller id update information. When update information is provided to a SIP device with an ANI/DNIS COS option enabled, the update information should contain the appropriate ANI/DNIS information and not the updated caller information. However, the SIP phone is a peer device. The 3300 has no control over what is displayed by the SIP phone. Thus, several ANI/DNIS COS fields can not be applied to SIP phones.

The following COS options affect ANI/DNIS behavior.

- **COS: ANI/ISDN Calling Number Only**. This COS field can not be applied to SIP phones.
- **COS: ANI/DNIS/ISDN Calling/Called Number**. This COS field shall not be applied to SIP phones.
- **COS: DNIS/Called Number Before Digit Modification**. When a MiNet phone makes an outgoing call on a SIP trunk, the display is incorrect on the MiNet phone (when compared to an outgoing call on an ISDN trunk).
- **COS: Dialed Digits During Outgoing Calls**. The SIP phone is a peer device. The 3300 has no control over what is displayed by the SIP phone. Thus, this COS field can not be applied to SIP phones. When a MiNet phone makes an outgoing call on a SIP trunk, the display is incorrect on the MiNet phone (when compared to an outgoing call on an ISDN trunk).

Name Suppression on Outgoing Trunk Calls FAC

Currently, a SIP phone cannot dial the Name Suppression On Outgoing Trunk Call FAC (feature access code). The following scenarios are ***not*** supported:

- A SIP phone user dials the Name Suppression On Outgoing Trunk Call FAC (e.g., **85) and expects to hear dial tone. The SIP user can then proceed to enter the PSTN destination.
- A SIP phone user dials the following in a single digit sequence: the Name Suppression On Outgoing Trunk Call FAC, followed by a '#' (digit_hash) and then a destination address (e.g., **85#96135922122). The expected behavior is to ring the PSTN destination and have name suppression enabled during the call.
- A SIP phone user tries to use a speed call button programmed with the following digit sequence: the Name Suppression On Outgoing Trunk Call FAC, followed by a '#' (digit_hash) and then a destination address (e.g., **85#96135922122). The expected behavior is to ring the PSTN destination and have name suppression enabled during the call.

A SIP phone user tries to dial a system speed call digit string (e.g., 1111) that translates into the following digit sequence: the Name Suppression On Outgoing Trunk Call FAC, followed by a '#' (digit_hash) and then a destination address (e.g., **85#96135922122). The expected behavior is to ring the PSTN destination and have name suppression enabled during the call.

Appendix E: WSDL-generated definitions for OIG Operations and Events

This appendix contains Web Service descriptions for the following services:

Note: These descriptions were auto-generated directly from the OIG WSDL files using a document generation tool.

Web Services

Name	Description
Standardccservice	Standard Call Control Service
SessionService	Session Service
EventHandlerCCService	Event Handler Service in application for receiving asynchronous messages from OIG. The OIG calls this event handler whenever a message needs to be delivered to application.

Standardccservice

See Also

- Complex Types
- Simple Types
- Elements
- Attributes

Complex Types: Standardccservice

Complex Types

Name	Description
ns1:callEventAttributeValue	
ns1:callEventMsg	
ns1:callStatusResult	
ns1:device	
ns1:deviceAttributeValue	
ns2:deviceConfigResult	
ns1:deviceFeaturesResult	
ns1:featureEventMsg	
ns1:forwardingInfo	

Name	Description
ns2:getDeviceConfigurationRequest	
ns2:getIcpIdRequest	
ns2:getLineAppearanceIdRequest	
ns2:getPhoneNumberIdRequest	
ns2:hotDeskUserLoginDeviceResult	
ns2:hotDeskUserLoginResult	
ns1:icpIdResult	
ns2:lineConfig	
ns2:loginExtHotDeskUserRequest	
ns2:loginHotDeskUserRequest	
ns2:monitorResult	
ns1:objectIdResult	
ns1:partyMember	
ns1:result	
ns2:setCFRequest	
ns1:standardEventMsg	
ns1:systemEventMsg	

Complex Type: ns1:callEventAttributeValue

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	ns1:callEventAttributeNames	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Element ns1:callEventAttribute [type callEventMsg]

Complex Type: ns1:callEventMsg

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:localCallId	xs:long	Yes		
ns1:callEventTime	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:type	ns1:callEventType	0..1	
ns1:cause	ns1:eventCauseType	0..1	
ns1:callState	ns1:callState	0..1	
ns1:callEventAttribute	ns1:callEventAttributeValue	0..*	
ns1:featuresAllowed	ns1:featuresAllowed	0..*	
ns1:device	ns1:device	0..*	
ns1:party	ns1:partyMember	0..*	
ns1:accountCode	xs:string	0..1	
ns1:accountCodeType	ns1:accountCodeType	0..1	

Referenced By

- Element ns1:callEvent [type callStatusResult]
- Element ns1:callEvent [type standardEventMsg]

Complex Type: ns1:callStatusResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:callEvent	ns1:callEventMsg	0..1	

Referenced By

- Element tns:return [element getCallStatusResponse]

Complex Type: ns1:device

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:deviceName	ns1:deviceName	0..1	
ns1:deviceAttribute	ns1:deviceAttributeValue	0..*	

Referenced By

- Element ns1:device [type callEventMsg]

Complex Type: ns1:deviceAttributeValue

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	ns1:deviceAttributeNames	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Element ns1:deviceAttribute [type device]

Complex Type: ns2:deviceConfigResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:isResilient	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:line	ns2:lineConfig	0..*	
ns2:primaryIcp	xs:string	0..1	
ns2:secondaryIcp	xs:string	0..1	

Referenced By

- Element tns:return [element getDeviceConfigurationResponse]

Complex Type: ns1:deviceFeaturesResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:DND	xs:boolean	Yes		
ns1:autoAnswer	xs:boolean	Yes		
ns1:msgWaitingLamp	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:hotDeskUserLoggedInDn	xs:string	0..1	
ns1:registrationDn	xs:string	0..1	
ns1:CFNAExt	ns1:forwardingInfo	0..1	
ns1:CFNAInt	ns1:forwardingInfo	0..1	
ns1:CFBusyExt	ns1:forwardingInfo	0..1	
ns1:CFBusyInt	ns1:forwardingInfo	0..1	
ns1:CFAlways	ns1:forwardingInfo	0..1	

Referenced By

- Element tns:return [element getDeviceFeaturesResponse]

Complex Type: ns1:featureEventMsg

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:time	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:state	ns1:callState	0..1	
ns1:devNumber	xs:string	0..1	
ns1:eventType	ns1:featureEventType	0..1	
ns1:userDn	xs:string	0..1	
ns1:groupDn	xs:string	0..1	
ns1:forwardDn	xs:string	0..1	

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Complex Type: ns1:forwardingInfo

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Element ns1:CFAlways [type deviceFeaturesResult]
- Element ns1:CFBusyExt [type deviceFeaturesResult]
- Element ns1:CFBusyInt [type deviceFeaturesResult]
- Element ns1:CFNAExt [type deviceFeaturesResult]
- Element ns1:CFNAInt [type deviceFeaturesResult]

Complex Type: ns2:getDeviceConfigurationRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element ns2:getDeviceConfigurationRequest
- Element tns:request [element getDeviceConfiguration]

Complex Type: ns2:getIcpIdRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:icpIpAddress	xs:string	1..1	

Referenced By

- Element ns2:getIcpIdRequest
- Element tns:request [element getIcpId]

Complex Type: ns2:getLineAppearanceIdRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		
ns2:buttonNum	xs:int	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element ns2:getLineAppearanceIdRequest
- Element tns:request [element getLineAppearanceId]

Complex Type: ns2:getPhoneNumberIdRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element ns2:getPhoneNumberIdRequest
- Element tns:request [element getPhoneNumberId]

Complex Type: ns2:hotDeskUserLoginDeviceResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:hotDeskUserObjectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:hotDeskDeviceDn	xs:string	0..1	

Referenced By

- Element tns:return [element getHotDeskUserLoginDeviceResponse]

Complex Type: ns2:hotDeskUserLoginResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:hotDeskDeviceObjectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:userDn	xs:string	0..1	

Referenced By

- Element tns:return [element getHotDeskUserDnResponse]

Complex Type: ns1:icpIdResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:connectionState	ns1:connectionState	0..1	
ns1:icpVersion	xs:string	0..1	

Referenced By

- Element tns:return [element getIcpIdResponse]

Complex Type: ns2:lineConfig

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:objectId	xs:long	Yes		
ns2:buttonNumber	xs:int	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:extNumber	xs:string	0..1	

Referenced By

- Element ns2:line [type deviceConfigResult]

Complex Type: ns2:loginExtHotDeskUserRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	

Referenced By

- Element ns2:loginExtHotDeskUserRequest
- Element tns:request [element loginExtHotDeskUser]

Complex Type: ns2:loginHotDeskUserRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	
ns2:hotDeskUserDn	xs:string	1..1	

Referenced By

- Element ns2:loginHotDeskUserRequest
- Element tns:request [element loginHotDeskUser]

Complex Type: ns2:monitorResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:objectId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	

Referenced By

- Element tns:return [element monitorFeaturesResponse]
- Element tns:return [element monitorObjectResponse]

Complex Type: ns1:objectIdResult

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:return [element getLineAppearanceIdResponse]
- Element tns:return [element getPhoneNumberIdResponse]

Complex Type: ns1:partyMember

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:callId	xs:long	Yes		
ns1:numberPrivacy	xs:boolean	Yes		
ns1:namePrivacy	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:memberNum	xs:string	0..1	
ns1:memberName	xs:string	0..1	
ns1:deviceType	ns1:deviceType	0..1	
ns1:remoteMemberExt	xs:string	0..1	
ns1:remoteMemberName	xs:string	0..1	
ns1:memberNetworkExt	xs:string	0..1	
ns1:peerId	xs:string	0..1	
ns1:peerName	xs:string	0..1	

Referenced By

- Element ns1:party [type callEventMsg]

Complex Type: ns1:result

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:return [element alternateCallResponse]
- Element tns:return [element answerCallResponse]
- Element tns:return [element camponCallResponse]
- Element tns:return [element cancelConsCallResponse]
- Element tns:return [element clearCallMeBackMsgResponse]
- Element tns:return [element clearCallResponse]
- Element tns:return [element conferenceCallResponse]
- Element tns:return [element consultationCallResponse]
- Element tns:return [element holdCallResponse]
- Element tns:return [element loginExtHotDeskUserResponse]
- Element tns:return [element loginHotDeskUserResponse]
- Element tns:return [element logoutExtHotDeskUserResponse]
- Element tns:return [element logoutHotDeskUserResponse]
- Element tns:return [element makeCallResponse]
- Element tns:return [element monitorPRGPresenceResponse]
- Element tns:return [element newCallResponse]
- Element tns:return [element outPulseDigitsResponse]
- Element tns:return [element pickupCallResponse]

- Element tns:return [element redirectCallResponse]
- Element tns:return [element registerEventHandlerResponse]
- Element tns:return [element retrieveCallResponse]
- Element tns:return [element sendCallMeBackMsgForCallResponse]
- Element tns:return [element sendCallMeBackMsgNoCallResponse]
- Element tns:return [element setAccountCodeResponse]
- Element tns:return [element setCallMeBackResponse]
- Element tns:return [element setCFAlwaysResponse]
- Element tns:return [element setCFBusyExternalResponse]
- Element tns:return [element setCFBusyInternalResponse]
- Element tns:return [element setCFNAExternalResponse]
- Element tns:return [element setCFNAInternalResponse]
- Element tns:return [element setDeviceDNDResponse]
- Element tns:return [element setPRGPresenceResponse]
- Element tns:return [element splitConferenceCallResponse]
- Element tns:return [element stopFeatureMonitorResponse]
- Element tns:return [element stopMonitorResponse]
- Element tns:return [element stopPRGPresenceMonitorResponse]
- Element tns:return [element tradeCallResponse]
- Element tns:return [element transferCallResponse]
- Element tns:return [element verifyHotDeskUserPinResponse]

Complex Type: ns2:setCFRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:request [element setCFAlways]
- Element tns:request [element setCFBusyExternal]
- Element tns:request [element setCFBusyInternal]
- Element tns:request [element setCFNAExternal]
- Element tns:request [element setCFNAInternal]
- Element ns2:setCFRequest

Complex Type: ns1:standardEventMsg

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:eventType	ns1:eventType	0..1	
ns1:callEvent	ns1:callEventMsg	0..1	
ns1:featureEvent	ns1:featureEventMsg	0..1	
ns1:systemEvent	ns1:systemEventMsg	0..1	

Referenced By

- Element tns:return [element getEventResponse]

Complex Type: ns1:systemEventMsg

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:icpId	xs:long	Yes		
ns1:time	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:eventType	ns1:systemEventType	0..1	
ns1:connectionState	ns1:connectionState	0..1	

Referenced By

- Element ns1:systemEvent [type standardEventMsg]

Simple Types: Standardccservice

Simple Types

Name	Description
ns1:accountCodeType	
ns1:attributeType	
ns1:callEventAttributeNames	
ns1:callEventType	
ns1:callState	
ns1:connectionState	
ns1:deviceAttributeNames	
ns1:deviceName	
ns1:deviceType	
ns1:eventCauseType	
ns1:eventType	
ns1:featureEventType	
ns1:featuresAllowed	
ns1:systemEventType	

Simple Type: ns1:accountCodeType

Derived By

Restricting xs:string

Enumeration

Value	Description
VERIFIED_ACCOUNT_CODE	
UNVERIFIED_ACCOUNT_CODE	

Referenced By

- Element ns1:accountCodeType [type callEventMsg]

Simple Type: ns1:attributeType

Derived By

Restricting xs:string

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Element ns1:attributeType [type callEventAttributeValue]
- Element ns1:attributeType [type deviceAttributeValue]

Simple Type: ns1:callEventAttributeNames

Derived By

Restricting xs:string

Enumeration

Value	Description
DIALED_DIGITS	
CURRENT_GLOBAL_CALL_ID	
PRIMARY_GLOBAL_CALLID	
SECONDARY_GLOBAL_CALLID	
DISTRIBUTION_RING_GRP_RINGALL	

Value	Description
DISTRIBUTION_RING_GRP_CASCADE	
DISTRIBUTION_HUNT_GRP_CIRCULAR	
DISTRIBUTION_HUNT_GRP_LINEAR	
DISTRIBUTION_PERSONAL_RING_GRP	

Referenced By

- Element ns1:attributeName [type callEventAttributeValue]

Simple Type: ns1:callEventType

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
ACCOUNT_CODE_SET	
ACD2_PATH	
ACD2_GROUP	
GROUP	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CALL_CLEAR	
CALL_CONFERENCED	
CALL_DELIVERED	
CALL_DIVERTED	
CALL_ESTABLISHED	
CALL_FAILED	
CALL_HELD	
CALL_ORIGINATED	
CALL_QUEUED	
CALL_RECEIVED	
CALL_RETRIEVED	
CALL_TRANSFERRED	
CMD_RESPONSE	
CONFERENCE_HELD	
CONFERENCE_FEATURE	

Value	Description
DEVICE_DROPPED	
EXTENSION_IN_USE	
FORWARD_FEATURE	
IN_SERVICE	
MONITOR_SET	
OUT_OF_SERVICE	
REMOTE_PARTY_UPDATE	
RESILIENT_DEVICE	
ROUTING_DEVICE	
TRUNK_DIGITS_OUTPULSED	
WORK_TIMER_EXPIRED	
USERID_FEATURE	
GROUP_PRESENCE_FEATURE	
ACD_EXPRESS_GROUP_EVENT	
MONITOR_FAILED_EVENT	
PRG_MONITOR_SET_EVENT	
PRG_MONITOR_FAILED_EVENT	

Referenced By

- Element ns1:type [type callEventMsg]

Simple Type: ns1:callState

Derived By

Restricting xs:string

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	
HELD	
IDLE	

Value	Description
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Element ns1:callState [type callEventMsg]
- Element ns1:state [type featureEventMsg]

Simple Type: ns1:connectionState

Derived By

Restricting xs:string

Enumeration

Value	Description
NOT_CONNECTED	
CONNECTED	
CONNECTION_FAILURE	
CONNECTION_ATTEMPT	
CONNECTION_INITIALIZING	

Referenced By

- Element ns1:connectionState [type icpIdResult]
- Element ns1:connectionState [type systemEventMsg]

Simple Type: ns1:deviceAttributeNames

Derived By

Restricting xs:string

Enumeration

Value	Description
NUMBER	
NAME	
DEVICE_TYPE	
DIALED_DIGITS	

Value	Description
ADV_MSG	
CALL_ID	
SUITE_PILOT_NUMBER	
LINE_APPEARANCE	
NUMBER_PRIVACY	
NAME_PRIVACY	
SUITE_PILOT_NAME	
PRG_NUMBER	
PRG_NAME	
TRUNK_OUTPULSED_DIGITS	
EHDU_NUMBER	
TRUNK_NUMBER	
TRUNK_NAME	
MFANI_INFO_DIGIT	
CALLING_LINE_CATEGORY	
PEER_NAME	
PEER_ID	

Referenced By

- Element ns1:attributeName [type deviceAttributeValue]

Simple Type: ns1:deviceName

Derived By

Restricting xs:string

Enumeration

Value	Description
CALLING_DEVICE	
CALLING_DEVICE_NETWORK_EXT	
ORIGINAL_DESTINATION_DEVICE	
GROUP_DEVICE	
CALLED_DEVICE	
CONTROLLER_DEVICE	
CONNECTED_DEVICE	
PICKUP_DEVICE	
SPLITTING_DEVICE	

Value	Description
TRANSFER_CONTROLLER_DEVICE	
TRANSFERRED_DEVICE	
NEW_DESTINATION_DEVICE	
DROPPED_DEVICE	
USING_DEVICE	
HOLDING_DEVICE	
WAITING_DEVICE	

Referenced By

- Element ns1:deviceName [type device]

Simple Type: ns1:deviceType

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
SIP_PRIVATE_TRUNK	
SIP_PUBLIC_TRUNK	
MULTICALL_GROUP	
HUNT_GROUP	
KEYLINE	
VOICE_SET_PRIME	
ACD_STRUCTURE	
LINE_APPEARANCE	
DATA_EXTENSION	
ACD2_PATH	
ACD2_AGENT	
ACD2_GROUP	
ROUTING_DEVICE	
ACD2_REMOTE_SUBGROUP	
RING_ALL_GROUP	
RING_CASCADE_GROUP	
ACDX_AGENT	
ACDX_GROUP	

Value	Description
ACDX_PATH	
PERSONAL_RING_GROUP	
PEER	
EXTERNAL_DEVICE_TYPE	
INTERNAL_DEVICE_TYPE	

Referenced By

- Element ns1:deviceType [type partyMember]

Simple Type: ns1:eventCauseType

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
ACCOUNT_CODE_SET	
ACD_AGENT_TIMEOUT	
ACD_CALL_QUEUED	
ACD_CALL_DELIVERED	
ACD_CALL_ANSWERED	
ACD_CALL_CLEARED	
ACD_CALL_OVERFLOWED	
ACD_CALL_ABANDONED	
ACD_CALL_CLEAR_INVOKED	
ACD_CALL_QUEUED_FAILED	
ACD_CALL_QUEUED_OVERFLOWED	
ACD_CALL_REDIRECTED	
ACD_CALL_RECEIVED	
ACD_REMOTE_CALL_QUEUED	
ACD_REMOTE_CALL_DEQUEUED	
ACD_REMOTE_CALL_DISTRIBUTED	
ACD_HELP_CALL	
ACD_INTERFLOWED_DISCONNECTED	
ACD_INTERFLOWED_REROUTED	
ACD_RAD_STARTED	

Value	Description
ACD_REQUEST	
ACD_REQUEST_DISCONNECTED	
ACD_REQUEST_REROUTED	
ACD_REQUEUE_REQUEST	
ACD_REQUEUE_REQUEST_DISCONNECTED	
ACD_REQUEUE_REQUEST_REROUTED	
ACD_SILENT_MONITOR_ESTABLISHED	
ACD_SILENT_MONITOR_DROPPED	
ACD_SILENT_MONITOR	
ACD_SILENT_MONITOR_INVOKED	
ALTERNATE_CALL_INVOKED	
ANSWERED	
ANSWERED_INVOKED	
CALLBACK_INVOKED	
CALLBACK_MATURED	
CALL_QUEUED	
CALL_IS_WAITING	
CLEARED	
CLEAR_INVOKED	
CONFERENCE_CALL_HELD	
CONFERENCE_CALL_RETRIEVED	
CONFERENCE_CONS_HELD	
CONFERENCE_CONS_RETRIEVED	
CONFERENCED	
CONFERENCE_ESTABLISHED	
CONFERENCE_INVOKED	
CONFERENCE_MEMBER_ADDED	
CONFERENCE_MEMBER_DROPPED	
CONFERENCE_MONITOR_SET	
CONFERENCE_REMOTE_UPDATE	
CONFERENCE_SPLIT	
CONFERENCE_SPLIT_INVOKED	
CONFERENCE_TRANSFERRED	
CONF_RETRIEVE_INVOKED	
CONS_HOLD	

Value	Description
CONS_HOLD_INVOKED	
CONS_CALL_INVOKED	
CONS_HELD_PARTY_DROPPED	
CONTROLLER_COMMS_FAILED	
CONTROLLER_COMMS_RESTORED	
DELIVERED	
DESTINATION_BUSY	
DESTINATION_DO_NOT_DISTURB	
DESTINATION_NOT_OBTAINABLE	
DESTINATION_UNAVAILABLE	
DEVICE_IN_SERVICE	
DEVICE_OUT_OF_SERVICE	
DIVERTED_ALWAYS_FROM	
DIVERTED_ALWAYS_TO	
DIVERTED_NO_ANSWER_FROM	
DIVERTED_NO_ANSWER_TO	
DIVERTED_NO_ANSWER_AWAY	
DIVERTED_ON_BUSY_FROM	
DIVERTED_ON_BUSY_TO	
DIVERTED_PICKUP_AWAY	
ERROR_DETECTED	
EXTENSION_IN_USE	
HARD_HELD_PARTY_DROPPED	
HARD_HOLD	
HARD_HOLD_INVOKED	
HARD_HOLD_RETRIEVED	
INTRUSION	
INTRUSION_INVOKED	
INVALID_ACCOUNT_CODE	
MEMBER_ANSWERED	
MONITOR_SET	
NETWORK_BUSY	
NETWORK_CONGESTION	
NETWORK_NOT_OBTAINABLE	
NEW_CALL_INVOKED	

Value	Description
NEW_CALL	
NO_ANSWER	
NO_SUCH_NUMBER	
ONS_CALL_ORIGINATED	
OTHER_DEVICE_CLEARED	
OTHER_DEVICE_DROPPED	
PICKUP	
PICKUP_INVOKED	
PRIVILEGE_VIOLATION	
QUEUED_PARTY_DROPPED	
RAD_UNAVAILABLE	
RECALL_HELD_PARTY	
REDIRECTED_AWAY	
REDIRECTED_FROM	
REDIRECTED_HANDOFF	
REDIRECTED_ON_ERROR	
REDIRECTED_TO	
REMOTE_PARTY_UPDATED	
GLOBAL_CALLID_UPDATED	
CALL_DISTRIBUTED	
CALL_ANSWERED	
CALL_DISCONNECTED	
CALL_OVERFLOWED	
CALL_WAITING_PARTY_DISCONNECTED	
CFNA_CALL_QUEUED	
CFA_CALL_QUEUED	
CFB_CALL_QUEUED	
CFA_DESTINATION_BUSY	
CFB_DESTINATION_BUSY	
RETRIEVE	
RETRIEVE_INVOKED	
ROUTING_DEVICE_CALL_QUEUED	
ROUTING_DEVICE_CALL_OVERFLOW	
ROUTING_DEVICE_CALL_REROUTED	
ROUTING_DEVICE_CALL_ABONDONED	

Value	Description
ROUTING_DEVICE_CFALWAYS_CALL_QUEUED	
ROUTING_DEVICE_CFBUSY_CALL_QUEUED	
ROUTING_DEVICE_CFNOANS_CALL_QUEUED	
ROUTING_DEVICE_RAD_STARTED	
ROUTING_DEVICE_REROUTED_CALL_QUEUED	
ROUTING_DEVICE_TRANS_CALL_QUEUED	
SECONDARY_CONTROLLER_COMMS_FAILED	
SECONDARY_MONITOR_SET	
SUPERVISED_TRANSFER	
TAP_CALL	
THIS_DEVICE_DROPPED	
TRANSFER_INVOKED	
TRANSFER_RECALL_NO_ANSWER	
TRANSFER_RECALL_ON_BUSY	
TRANSFER_TO_BUSY	
TRANSFERRED_CALL	
TRUNKS_BUSY	
TRUNK_DIGITS_OUTPULSED	
UNSUPERVISED_TRANSFER	
WAITING_CALL_RINGING	
WORK_TIMER_EXPIRED	
HANDOFF_PUSH	
PEER_OUT_OF_SERVICE	
PEER_INSERVICE	
EHDU_ACCESS	
EHDU_CALL_BACK	
CLOSING_CALL_SERVER	

Referenced By

- Element ns1:cause [type callEventMsg]

Simple Type: ns1:eventType

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
CALL_EVENT	
FEATURE_EVENT	
ACD_EVENT	
CONFERENCE_EVENT	
SYSTEM_EVENT	

Referenced By

- Element ns1:eventType [type standardEventMsg]

Simple Type: ns1:featureEventType

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
AUTO_ANSWER_FEATURE	
DO_NOT_DISTURB	
MSG_WAITING_INDICATOR	
INSERVICE	
DO_NOT_DISTURB_ALL	
MSG_WAITING_IND_ALL	
MAKE_BUSY	
JOIN_GROUP	
LEAVE_GROUP	
JOIN_ALL_ACD_GROUPS	
MAKE_BUSY_STATE	
LEAVE_All_ACD_GROUPS	
ACD_AGENT	
ACD_AGENT_BLOCKED	
ACD2_LOGIN_NO_LICENSE	
HOT_DESK_USER	
CF_ALWAYS	
CF_BUSY_EXTERNAL	

Value	Description
CF_BUSY_INTERNAL	
CF_NO_ANSWER_INTERNAL	
CF_NO_ANSWER_EXTERNAL	
CF_NO_ANSWER_BUSY	
CF_ALL_MODES	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CONFERENCE_FEATURE	
FORWARD_FEATURE	
USER_ID_FEATURE_EVENT	

Referenced By

- Element ns1:eventType [type featureEventMsg]

Simple Type: ns1:featuresAllowed

Derived By

Restricting xs:string

Enumeration

Value	Description
NOT_SUPPORTED	
ANSWER_CALL	
REDIRECT_CALL	
CAMPON_CALL	
SET_CALL_ME_BACK	
INTRUDE_CALL	
ALTERNATE_CALL	
TRADE_CALL	
CONFERENCE_CALL	
SPLIT_CONFERENCE_CALL	
CONSULTATION_CALL	
CANCEL_CONSULTATION_CALL	
TRANSFER_CALL	
SEND_CALLBACK_MESSAGE	
SET_ACCOUNT_CODE	
NEW_CALL	

Value	Description
CLEAR_CALL	
OUT_PULSE_DIGITS	
HOLD_CALL	
RETRIEVE_CALL	
LISTEN_CALL	
ASSIGN_CALLER_ID	
ALTER_WORK_TIMER	
TAP_CALL	

Referenced By

- Element ns1:featuresAllowed [type callEventMsg]

Simple Type: ns1:systemEventType

Derived By

Restricting xs:string

Enumeration

Value	Description
ICP_COMMUNICATION_EVENT	
OIG_GW_SHUTDOWN	

Referenced By

- Element ns1:eventType [type systemEventMsg]

Elements: Standardccservice

Elements

Name	Description
tns:accCode [element setAccountCode]	
tns:accountCode [element makeCall]	
ns2:accountCode [element makeCallRequest]	
ns2:accountCode [element setAccountCodeRequest]	
ns1:accountCode [type callEventMsg]	
ns1:accountCodeType [type callEventMsg]	
tns:alternateCall	
tns:alternateCallResponse	
tns:answerCall	

Name	Description
tns:answerCallResponse	
ns1:attributeName [type callEventAttributeValue]	
ns1:attributeName [type deviceAttributeValue]	
ns1:attributeType [type callEventAttributeValue]	
ns1:attributeType [type deviceAttributeValue]	
ns1:attributeValue [type callEventAttributeValue]	
ns1:attributeValue [type deviceAttributeValue]	
ns1:callEvent [type callStatusResult]	
ns1:callEvent [type standardEventMsg]	
ns1:callEventAttribute [type callEventMsg]	
ns1:callState [type callEventMsg]	
tns:camponCall	
tns:camponCallResponse	
tns:cancelConsCall	
tns:cancelConsCallResponse	
ns1:cause [type callEventMsg]	
ns1:CFAlways [type deviceFeaturesResult]	
ns1:CFBusyExt [type deviceFeaturesResult]	
ns1:CFBusyInt [type deviceFeaturesResult]	
ns1:CFNAExt [type deviceFeaturesResult]	
ns1:CFNAInt [type deviceFeaturesResult]	
tns:clearCall	
tns:clearCallMeBackMsg	
ns2:clearCallMeBackMsgRequest	
tns:clearCallMeBackMsgResponse	
tns:clearCallResponse	
tns:conferenceCall	
tns:conferenceCallResponse	
ns1:connectionState [type icpIdResult]	
ns1:connectionState [type systemEventMsg]	
tns:consultationCall	
tns:consultationCallResponse	
ns1:device [type callEventMsg]	
ns1:deviceAttribute [type device]	
tns:deviceDn [element clearCallMeBackMsg]	

Name	Description
ns2:deviceDn [element clearCallMeBackMsgRequest]	
ns2:deviceDn [element sendCallMeBackMsgNoCallRequest]	
ns1:deviceName [type device]	
ns1:deviceType [type partyMember]	
ns1:devNumber [type featureEventMsg]	
tns:digits [element outPulseDigits]	
tns:dn [element sendCallMeBackMsgNoCall]	
ns1:dn [type forwardingInfo]	
ns2:dn [type setCFRequest]	
tns:dndState [element setDeviceDND]	
ns2:dtmfDigits [element outPulseDigitsRequest]	
ns1:errorDescription [type callStatusResult]	
ns2:errorDescription [type deviceConfigResult]	
ns1:errorDescription [type deviceFeaturesResult]	
ns2:errorDescription [type hotDeskUserLoginDeviceResult]	
ns2:errorDescription [type hotDeskUserLoginResult]	
ns1:errorDescription [type icpIdResult]	
ns2:errorDescription [type monitorResult]	
ns1:errorDescription [type objectIdResult]	
ns1:errorDescription [type result]	
ns1:errorDescription [type standardEventMsg]	
ns2:eventHandlerURL [element registerEventHandler]	
tns:eventHandlerURL [element registerEventHandler]	
ns1:eventType [type featureEventMsg]	
ns1:eventType [type standardEventMsg]	
ns1:eventType [type systemEventMsg]	
ns2:extNumber [type lineConfig]	
ns1:featureEvent [type standardEventMsg]	
ns1:featuresAllowed [type callEventMsg]	
ns1:forwardDn [type featureEventMsg]	
tns:getCallStatus	
tns:getCallStatusResponse	
tns:getDeviceConfiguration	
ns2:getDeviceConfigurationRequest	

Name	Description
tns:getDeviceConfigurationResponse	
tns:getDeviceFeatures	
tns:getDeviceFeaturesResponse	
tns:getEvent	
tns:getEventResponse	
tns:getHotDeskUserDn	
tns:getHotDeskUserDnResponse	
tns:getHotDeskUserLoginDevice	
tns:getHotDeskUserLoginDeviceResponse	
tns:getIcpId	
ns2:getIcpIdRequest	
tns:getIcpIdResponse	
tns:getLineAppearanceId	
ns2:getLineAppearanceIdRequest	
tns:getLineAppearanceIdResponse	
tns:getPhoneNumberId	
ns2:getPhoneNumberIdRequest	
tns:getPhoneNumberIdResponse	
ns1:groupDn [type featureEventMsg]	
tns:holdCall	
tns:holdCallResponse	
ns2:hotDeskDeviceDn [type hotDeskUserLoginDeviceResult]	
tns:hotDeskDeviceObjectId [element getHotDeskUserDn]	
tns:hotDeskUserDn [element verifyHotDeskUserPin]	
ns2:hotDeskUserDn [type loginHotDeskUserRequest]	
ns1:hotDeskUserLoggedInDn [type deviceFeaturesResult]	
tns:hotDeskUserObjectId [element getHotDeskUserLoginDevice]	
tns:icpId [element monitorPRGPresence]	
tns:icpId [element stopPRGPresenceMonitor]	
ns2:icpIpAddress [type getIcpIdRequest]	
ns1:icpVersion [type icpIdResult]	
tns:invokerDn [element setPRGPresence]	
ns2:line [type deviceConfigResult]	

Name	Description
tns:localCallId [element alternateCall]	
tns:localCallId [element answerCall]	
tns:localCallId [element camponCall]	
tns:localCallId [element cancelConsCall]	
tns:localCallId [element clearCall]	
tns:localCallId [element conferenceCall]	
tns:localCallId [element consultationCall]	
tns:localCallId [element holdCall]	
tns:localCallId [element newCall]	
tns:localCallId [element outPulseDigits]	
tns:localCallId [element redirectCall]	
tns:localCallId [element retrieveCall]	
tns:localCallId [element sendCallMeBackMsgForCall]	
tns:localCallId [element setAccountCode]	
tns:localCallId [element setCallMeBack]	
tns:localCallId [element splitConferenceCall]	
tns:localCallId [element tradeCall]	
tns:localCallId [element transferCall]	
tns:loginExtHotDeskUser	
ns2:loginExtHotDeskUserRequest	
tns:loginExtHotDeskUserResponse	
tns:loginHotDeskUser	
ns2:loginHotDeskUserRequest	
tns:loginHotDeskUserResponse	
tns:logoutExtHotDeskUser	
tns:logoutExtHotDeskUserResponse	
tns:logoutHotDeskUser	
tns:logoutHotDeskUserResponse	
tns:makeCall	
ns2:makeCallRequest	
tns:makeCallResponse	
ns1:memberName [type partyMember]	
ns1:memberNetworkExt [type partyMember]	
ns1:memberNum [type partyMember]	
tns:monitorFeatures	

Name	Description
tns:monitorFeaturesResponse	
tns:monitorObject	
tns:monitorObjectResponse	
tns:monitorPRGPresence	
tns:monitorPRGPresenceResponse	
tns:newCall	
tns:newCallResponse	
tns:number [element consultationCall]	
tns:number [element makeCall]	
ns2:number [element makeCallRequest]	
tns:objectId [element alternateCall]	
tns:objectId [element answerCall]	
tns:objectId [element camponCall]	
tns:objectId [element cancelConsCall]	
tns:objectId [element clearCall]	
tns:objectId [element conferenceCall]	
tns:objectId [element consultationCall]	
tns:objectId [element getCallStatus]	
tns:objectId [element getDeviceFeatures]	
tns:objectId [element holdCall]	
tns:objectId [element logoutExtHotDeskUser]	
tns:objectId [element logoutHotDeskUser]	
tns:objectId [element makeCall]	
tns:objectId [element monitorFeatures]	
tns:objectId [element monitorObject]	
tns:objectId [element newCall]	
tns:objectId [element outPulseDigits]	
tns:objectId [element pickupCall]	
tns:objectId [element redirectCall]	
tns:objectId [element retrieveCall]	
tns:objectId [element setAccountCode]	
tns:objectId [element setCallMeBack]	
tns:objectId [element setDeviceDND]	
tns:objectId [element setPRGPresence]	
tns:objectId [element splitConferenceCall]	

Name	Description
tns:objectId [element stopFeatureMonitor]	
tns:objectId [element stopMonitor]	
tns:objectId [element tradeCall]	
tns:objectId [element transferCall]	
tns:objectId [element verifyHotDeskUserPin]	
tns:objId [element sendCallMeBackMsgForCall]	
tns:objId [element sendCallMeBackMsgNoCall]	
tns:objIdOfDeviceWithMsg [element clearCallMeBackMsg]	
tns:outPulseDigits	
ns2:outPulseDigitsRequest	
tns:outPulseDigitsResponse	
ns1:party [type callEventMsg]	
ns1:peerId [type partyMember]	
ns1:peerName [type partyMember]	
tns:pickupCall	
ns2:pickupCallRequest	
tns:pickupCallResponse	
ns2:pickupDn [element pickupCallRequest]	
tns:pin [element verifyHotDeskUserPin]	
ns2:pin [type loginExtHotDeskUserRequest]	
ns2:pin [type loginHotDeskUserRequest]	
tns:presenceValue [element setPRGPresence]	
tns:PRGDn [element setPRGPresence]	
ns2:PRGDn [element setPRGPresenceRequest]	
ns2:primaryIcp [type deviceConfigResult]	
ns2:primeDn [type getDeviceConfigurationRequest]	
ns2:primeDn [type getLineAppearanceIdRequest]	
ns2:primeDn [type getPhoneNumberIdRequest]	
tns:redirectCall	
ns2:redirectCallRequest	
tns:redirectCallResponse	
tns:redirectDn [element redirectCall]	
ns2:redirectDn [element redirectCallRequest]	
ns2:registerEventHandler	

Name	Description
tns:registerEventHandler	
tns:registerEventHandlerResponse	
ns1:registrationDn [type deviceFeaturesResult]	
ns1:remoteMemberExt [type partyMember]	
ns1:remoteMemberName [type partyMember]	
tns:request [element getDeviceConfiguration]	
tns:request [element getIcpId]	
tns:request [element getLineAppearanceId]	
tns:request [element getPhoneNumberId]	
tns:request [element loginExtHotDeskUser]	
tns:request [element loginHotDeskUser]	
tns:request [element setCFAlways]	
tns:request [element setCFBusyExternal]	
tns:request [element setCFBusyInternal]	
tns:request [element setCFNAExternal]	
tns:request [element setCFNAInternal]	
tns:retrieveCall	
tns:retrieveCallResponse	
tns:return [element alternateCallResponse]	
tns:return [element answerCallResponse]	
tns:return [element camponCallResponse]	
tns:return [element cancelConsCallResponse]	
tns:return [element clearCallMeBackMsgResponse]	
tns:return [element clearCallResponse]	
tns:return [element conferenceCallResponse]	
tns:return [element consultationCallResponse]	
tns:return [element getCallStatusResponse]	
tns:return [element getDeviceConfigurationResponse]	
tns:return [element getDeviceFeaturesResponse]	
tns:return [element getEventResponse]	
tns:return [element getHotDeskUserDnResponse]	
tns:return [element getHotDeskUserLoginDeviceResponse]	
tns:return [element getIcpIdResponse]	
tns:return [element getLineAppearanceIdResponse]	

Name	Description
tns:return [element getPhoneNumberIdResponse]	
tns:return [element holdCallResponse]	
tns:return [element loginExtHotDeskUserResponse]	
tns:return [element loginHotDeskUserResponse]	
tns:return [element logoutExtHotDeskUserResponse]	
tns:return [element logoutHotDeskUserResponse]	
tns:return [element makeCallResponse]	
tns:return [element monitorFeaturesResponse]	
tns:return [element monitorObjectResponse]	
tns:return [element monitorPRGPresenceResponse]	
tns:return [element newCallResponse]	
tns:return [element outPulseDigitsResponse]	
tns:return [element pickupCallResponse]	
tns:return [element redirectCallResponse]	
tns:return [element registerEventHandlerResponse]	
tns:return [element retrieveCallResponse]	
tns:return [element sendCallMeBackMsgForCallResponse]	
tns:return [element sendCallMeBackMsgNoCallResponse]	
tns:return [element setAccountCodeResponse]	
tns:return [element setCallMeBackResponse]	
tns:return [element setCFAlwaysResponse]	
tns:return [element setCFBusyExternalResponse]	
tns:return [element setCFBusyInternalResponse]	
tns:return [element setCFNAExternalResponse]	
tns:return [element setCFNAInternalResponse]	
tns:return [element setDeviceDNDResponse]	
tns:return [element setPRGPresenceResponse]	
tns:return [element splitConferenceCallResponse]	
tns:return [element stopFeatureMonitorResponse]	
tns:return [element stopMonitorResponse]	
tns:return [element stopPRGPresenceMonitorResponse]	
tns:return [element tradeCallResponse]	
tns:return [element transferCallResponse]	

Name	Description
tns:return [element verifyHotDeskUserPinResponse]	
tns:ringDn [element pickupCall]	
ns2:ringDn [element pickupCallRequest]	
ns2:secondaryIcp [type deviceConfigResult]	
tns:sendCallMeBackMsgForCall	
tns:sendCallMeBackMsgForCallResponse	
tns:sendCallMeBackMsgNoCall	
ns2:sendCallMeBackMsgNoCallRequest	
tns:sendCallMeBackMsgNoCallResponse	
tns:sessionId [element alternateCall]	
tns:sessionId [element answerCall]	
tns:sessionId [element camponCall]	
tns:sessionId [element cancelConsCall]	
tns:sessionId [element clearCall]	
tns:sessionId [element clearCallMeBackMsg]	
tns:sessionId [element conferenceCall]	
tns:sessionId [element consultationCall]	
tns:sessionId [element getCallStatus]	
tns:sessionId [element getDeviceFeatures]	
tns:sessionId [element getEvent]	
tns:sessionId [element getHotDeskUserDn]	
tns:sessionId [element getHotDeskUserLoginDevice]	
tns:sessionId [element holdCall]	
tns:sessionId [element logoutExtHotDeskUser]	
tns:sessionId [element logoutHotDeskUser]	
tns:sessionId [element makeCall]	
tns:sessionId [element monitorFeatures]	
tns:sessionId [element monitorObject]	
tns:sessionId [element monitorPRGPresence]	
tns:sessionId [element newCall]	
tns:sessionId [element outPulseDigits]	
tns:sessionId [element pickupCall]	
tns:sessionId [element redirectCall]	
tns:sessionId [element registerEventHandler]	
tns:sessionId [element retrieveCall]	

Name	Description
tns:sessionId [element sendCallMeBackMsgForCall]	
tns:sessionId [element sendCallMeBackMsgNoCall]	
tns:sessionId [element setAccountCode]	
tns:sessionId [element setCallMeBack]	
tns:sessionId [element setDeviceDND]	
tns:sessionId [element setPRGPresence]	
tns:sessionId [element splitConferenceCall]	
tns:sessionId [element stopFeatureMonitor]	
tns:sessionId [element stopMonitor]	
tns:sessionId [element stopPRGPresenceMonitor]	
tns:sessionId [element tradeCall]	
tns:sessionId [element transferCall]	
tns:sessionId [element verifyHotDeskUserPin]	
tns:setAccountCode	
ns2:setAccountCodeRequest	
tns:setAccountCodeResponse	
tns:setCallMeBack	
tns:setCallMeBackResponse	
tns:setCFAlways	
tns:setCFAlwaysResponse	
tns:setCFBusyExternal	
tns:setCFBusyExternalResponse	
tns:setCFBusyInternal	
tns:setCFBusyInternalResponse	
tns:setCFNAExternal	
tns:setCFNAExternalResponse	
tns:setCFNAInternal	
tns:setCFNAInternalResponse	
ns2:setCFRequest	
tns:setDeviceDND	
tns:setDeviceDNDResponse	
tns:setPRGPresence	
ns2:setPRGPresenceRequest	
tns:setPRGPresenceResponse	
tns:splitConferenceCall	

Name	Description
tns:splitConferenceCallResponse	
ns2:standardCCService	
ns1:state [type featureEventMsg]	
tns:stopFeatureMonitor	
tns:stopFeatureMonitorResponse	
tns:stopMonitor	
tns:stopMonitorResponse	
tns:stopPRGPresenceMonitor	
tns:stopPRGPresenceMonitorResponse	
ns1:systemEvent [type standardEventMsg]	
tns:timeout [element getEvent]	
tns:tradeCall	
tns:tradeCallResponse	
tns:transferCall	
tns:transferCallResponse	
ns1:type [type callEventMsg]	
ns1:userDn [type featureEventMsg]	
ns2:userDn [type hotDeskUserLoginResult]	
tns:verifyHotDeskUserPin	
tns:verifyHotDeskUserPinResponse	

Element: tns:accCode [element setAccountCode]

Derived By

Type xs:string

Referenced By

- Element tns:setAccountCode

Element: tns:accountCode [element makeCall]

Derived By

Type xs:string

Referenced By

- Element tns:makeCall

Element: ns2:accountCode [element makeCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:makeCallRequest

Element: ns2:accountCode [element setAccountCodeRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:setAccountCodeRequest

Element: ns1:accountCode [type callEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:accountCodeType [type callEventMsg]

Derived By

Type ns1:accountCodeType

Enumeration

Value	Description
VERIFIED_ACCOUNT_CODE	
UNVERIFIED_ACCOUNT_CODE	

Referenced By

- Complex Type ns1:callEventMsg

Element: tns:alternateCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:alternateCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:answerCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:answerCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:attributeName [type callEventAttributeValue]

Derived By

Type ns1:callEventAttributeNames

Enumeration

Value	Description
DIALED_DIGITS	
CURRENT_GLOBAL_CALL_ID	
PRIMARY_GLOBAL_CALLID	
SECONDARY_GLOBAL_CALLID	
DISTRIBUTION_RING_GRP_RINGALL	
DISTRIBUTION_RING_GRP_CASCADE	
DISTRIBUTION_HUNT_GRP_CIRCULAR	
DISTRIBUTION_HUNT_GRP_LINEAR	
DISTRIBUTION_PERSONAL_RING_GRP	

Referenced By

- Complex Type ns1:callEventAttributeValue

Element: ns1:attributeName [type deviceAttributeValue]

Derived By

Type ns1:deviceAttributeNames

Enumeration

Value	Description
NUMBER	
NAME	

Value	Description
DEVICE_TYPE	
DIALED_DIGITS	
ADV_MSG	
CALL_ID	
SUITE_PILOT_NUMBER	
LINE_APPEARANCE	
NUMBER_PRIVACY	
NAME_PRIVACY	
SUITE_PILOT_NAME	
PRG_NUMBER	
PRG_NAME	
TRUNK_OUTPULSED_DIGITS	
EHDU_NUMBER	
TRUNK_NUMBER	
TRUNK_NAME	
MFANI_INFO_DIGIT	
CALLING_LINE_CATEGORY	
PEER_NAME	
PEER_ID	

Referenced By

- Complex Type ns1:deviceAttributeValue

Element: ns1:attributeType [type callEventAttributeValue]

Derived By

Type ns1:attributeType

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Complex Type ns1:callEventAttributeValue

Element: ns1:attributeType [type deviceAttributeValue]

Derived By

Type ns1:attributeType

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Complex Type ns1:deviceAttributeValue

Element: ns1:attributeValue [type callEventAttributeValue]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:callEventAttributeValue

Element: ns1:attributeValue [type deviceAttributeValue]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:deviceAttributeValue

Element: ns1:callEvent [type callStatusResult]

Derived By

Type ns1:callEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:localCallId	xs:long	Yes		
ns1:callEventTime	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:type	ns1:callEventType	0..1	
ns1:cause	ns1:eventCauseType	0..1	
ns1:callState	ns1:callState	0..1	
ns1:callEventAttribute	ns1:callEventAttributeValue	0..*	
ns1:featuresAllowed	ns1:featuresAllowed	0..*	
ns1:device	ns1:device	0..*	
ns1:party	ns1:partyMember	0..*	
ns1:accountCode	xs:string	0..1	
ns1:accountCodeType	ns1:accountCodeType	0..1	

Referenced By

- Complex Type ns1:callStatusResult

Element: ns1:callEvent [type standardEventMsg]

Derived By

Type ns1:callEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:localCallId	xs:long	Yes		
ns1:callEventTime	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:type	ns1:callEventType	0..1	
ns1:cause	ns1:eventCauseType	0..1	
ns1:callState	ns1:callState	0..1	
ns1:callEventAttribute	ns1:callEventAttributeValue	0..*	
ns1:featuresAllowed	ns1:featuresAllowed	0..*	

Component	Type	Occurs	Description
ns1:device	ns1:device	0..*	
ns1:party	ns1:partyMember	0..*	
ns1:accountCode	xs:string	0..1	
ns1:accountCodeType	ns1:accountCodeType	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:callEventAttribute [type callEventMsg]

Derived By

Type ns1:callEventAttributeValue

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	ns1:callEventAttributeNames	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:callState [type callEventMsg]

Derived By

Type ns1:callState

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	

Value	Description
HELD	
IDLE	
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Complex Type ns1:callEventMsg

Element: tns:camponCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:camponCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:cancelConsCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:cancelConsCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:cause [type callEventMsg]

Derived By

Type ns1:eventCauseType

Enumeration

Value	Description
UNKNOWN	
ACCOUNT_CODE_SET	
ACD_AGENT_TIMEOUT	
ACD_CALL_QUEUED	
ACD_CALL_DELIVERED	
ACD_CALL_ANSWERED	

Value	Description
ACD_CALL_CLEARED	
ACD_CALL_OVERFLOWED	
ACD_CALL_ABANDONED	
ACD_CALL_CLEAR_INVOKED	
ACD_CALL_QUEUED_FAILED	
ACD_CALL_QUEUED_OVERFLOWED	
ACD_CALL_REDIRECTED	
ACD_CALL_RECEIVED	
ACD_REMOTE_CALL_QUEUED	
ACD_REMOTE_CALL_DEQUEUED	
ACD_REMOTE_CALL_DISTRIBUTED	
ACD_HELP_CALL	
ACD_INTERFLOWED_DISCONNECTED	
ACD_INTERFLOWED_REROUTED	
ACD_RAD_STARTED	
ACD_REQUEST	
ACD_REQUEST_DISCONNECTED	
ACD_REQUEST_REROUTED	
ACD_REQUEUE_REQUEST	
ACD_REQUEUE_REQUEST_DISCONNECTED	
ACD_REQUEUE_REQUEST_REROUTED	
ACD_SILENT_MONITOR_ESTABLISHED	
ACD_SILENT_MONITOR_DROPPED	
ACD_SILENT_MONITOR	
ACD_SILENT_MONITOR_INVOKED	
ALTERNATE_CALL_INVOKED	
ANSWERED	
ANSWERED_INVOKED	
CALLBACK_INVOKED	
CALLBACK_MATURED	
CALL_QUEUED	
CALL_IS_WAITING	
CLEARED	
CLEAR_INVOKED	
CONFERENCE_CALL_HELD	

Value	Description
CONFERENCE_CALL_RETRIEVED	
CONFERENCE_CONS_HELD	
CONFERENCE_CONS_RETRIEVED	
CONFERENCED	
CONFERENCE_ESTABLISHED	
CONFERENCE_INVOKED	
CONFERENCE_MEMBER_ADDED	
CONFERENCE_MEMBER_DROPPED	
CONFERENCE_MONITOR_SET	
CONFERENCE_REMOTE_UPDATE	
CONFERENCE_SPLIT	
CONFERENCE_SPLIT_INVOKED	
CONFERENCE_TRANSFERRED	
CONF_RETRIEVE_INVOKED	
CONS_HOLD	
CONS_HOLD_INVOKED	
CONS_CALL_INVOKED	
CONS_HELD_PARTY_DROPPED	
CONTROLLER_COMMS_FAILED	
CONTROLLER_COMMS_RESTORED	
DELIVERED	
DESTINATION_BUSY	
DESTINATION_DO_NOT_DISTURB	
DESTINATION_NOT_OBTAINABLE	
DESTINATION_UNAVAILABLE	
DEVICE_IN_SERVICE	
DEVICE_OUT_OF_SERVICE	
DIVERTED_ALWAYS_FROM	
DIVERTED_ALWAYS_TO	
DIVERTED_NO_ANSWER_FROM	
DIVERTED_NO_ANSWER_TO	
DIVERTED_NO_ANSWER_AWAY	
DIVERTED_ON_BUSY_FROM	
DIVERTED_ON_BUSY_TO	
DIVERTED_PICKUP_AWAY	

Value	Description
ERROR_DETECTED	
EXTENSION_IN_USE	
HARD_HELD_PARTY_DROPPED	
HARD_HOLD	
HARD_HOLD_INVOKED	
HARD_HOLD_RETRIEVED	
INTRUSION	
INTRUSION_INVOKED	
INVALID_ACCOUNT_CODE	
MEMBER_ANSWERED	
MONITOR_SET	
NETWORK_BUSY	
NETWORK_CONGESTION	
NETWORK_NOT_OBTAINABLE	
NEW_CALL_INVOKED	
NEW_CALL	
NO_ANSWER	
NO_SUCH_NUMBER	
ONS_CALL_ORIGINATED	
OTHER_DEVICE_CLEARED	
OTHER_DEVICE_DROPPED	
PICKUP	
PICKUP_INVOKED	
PRIVILEGE_VIOLATION	
QUEUED_PARTY_DROPPED	
RAD_UNAVAILABLE	
RECALL_HELD_PARTY	
REDIRECTED_AWAY	
REDIRECTED_FROM	
REDIRECTED_HANDOFF	
REDIRECTED_ON_ERROR	
REDIRECTED_TO	
REMOTE_PARTY_UPDATED	
GLOBAL_CALLID_UPDATED	
CALL_DISTRIBUTED	

Value	Description
CALL_ANSWERED	
CALL_DISCONNECTED	
CALL_OVERFLOWED	
CALL_WAITING_PARTY_DISCONNECTED	
CFNA_CALL_QUEUED	
CFA_CALL_QUEUED	
CFB_CALL_QUEUED	
CFA_DESTINATION_BUSY	
CFB_DESTINATION_BUSY	
RETRIEVE	
RETRIEVE_INVOKED	
ROUTING_DEVICE_CALL_QUEUED	
ROUTING_DEVICE_CALL_OVERFLOW	
ROUTING_DEVICE_CALL_REROUTED	
ROUTING_DEVICE_CALL_ABONDONED	
ROUTING_DEVICE_CFALWAYS_CALL_QUEUED	
ROUTING_DEVICE_CFBUSY_CALL_QUEUED	
ROUTING_DEVICE_CFNOANS_CALL_QUEUED	
ROUTING_DEVICE_RAD_STARTED	
ROUTING_DEVICE_REROUTED_CALL_QUEUED	
ROUTING_DEVICE_TRANS_CALL_QUEUED	
SECONDARY_CONTROLLER_COMMS_FAILED	
SECONDARY_MONITOR_SET	
SUPERVISED_TRANSFER	
TAP_CALL	
THIS_DEVICE_DROPPED	
TRANSFER_INVOKED	
TRANSFER_RECALL_NO_ANSWER	
TRANSFER_RECALL_ON_BUSY	
TRANSFER_TO_BUSY	
TRANSFERRED_CALL	
TRUNKS_BUSY	
TRUNK_DIGITS_OUTPULSED	
UNSUPERVISED_TRANSFER	
WAITING_CALL_RINGING	

Value	Description
WORK_TIMER_EXPIRED	
HANDOFF_PUSH	
PEER_OUT_OF_SERVICE	
PEER_INSERVICE	
EHDU_ACCESS	
EHDU_CALL_BACK	
CLOSING_CALL_SERVER	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:CFAlways [type deviceFeaturesResult]

Derived By

Type ns1:forwardingInfo

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns1:CFBusyExt [type deviceFeaturesResult]

Derived By

Type ns1:forwardingInfo

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns1:CFBusyInt [type deviceFeaturesResult]

Derived By

Type ns1:forwardingInfo

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns1:CFNAExt [type deviceFeaturesResult]

Derived By

Type ns1:forwardingInfo

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns1:CFNAInt [type deviceFeaturesResult]

Derived By

Type ns1:forwardingInfo

Attributes

Name	Type	Required?	Default	Description
ns1:fwdOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:dn	xs:string	0..1	

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: tns:clearCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:clearCallMeBackMsg

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objIdOfDeviceWithMsg	xs:long	1..1	
tns:deviceDn	xs:string	0..1	

Element: ns2:clearCallMeBackMsgRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objIdOfDeviceWithMsg	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:deviceDn	xs:string	1..1	

Element: tns:clearCallMeBackMsgResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:clearCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:conferenceCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:conferenceCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:connectionState [type icpIdResult]

Derived By

Type ns1:connectionState

Enumeration

Value	Description
NOT_CONNECTED	
CONNECTED	
CONNECTION_FAILURE	
CONNECTION_ATTEMPT	
CONNECTION_INITIALIZING	

Referenced By

- Complex Type ns1:icpIdResult

Element: ns1:connectionState [type systemEventMsg]

Derived By

Type ns1:connectionState

Enumeration

Value	Description
NOT_CONNECTED	
CONNECTED	
CONNECTION_FAILURE	
CONNECTION_ATTEMPT	
CONNECTION_INITIALIZING	

Referenced By

- Complex Type ns1:systemEventMsg

Element: tns:consultationCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	
tns:number	xs:string	0..1	

Element: tns:consultationCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:device [type callEventMsg]

Derived By

Type ns1:device

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:deviceName	ns1:deviceName	0..1	
ns1:deviceAttribute	ns1:deviceAttributeValue	0..*	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:deviceAttribute [type device]

Derived By

Type ns1:deviceAttributeValue

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	ns1:deviceAttributeNames	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Complex Type ns1:device

Element: tns:deviceDn [element clearCallMeBackMsg]

Derived By

Type xs:string

Referenced By

- Element tns:clearCallMeBackMsg

Element: ns2:deviceDn [element clearCallMeBackMsgRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:clearCallMeBackMsgRequest

Element: ns2:deviceDn [element sendCallMeBackMsgNoCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:sendCallMeBackMsgNoCallRequest

Element: ns1:deviceName [type device]

Derived By

Type ns1:deviceName

Enumeration

Value	Description
CALLING_DEVICE	
CALLING_DEVICE_NETWORK_EXT	
ORIGINAL_DESTINATION_DEVICE	
GROUP_DEVICE	
CALLED_DEVICE	
CONTROLLER_DEVICE	
CONNECTED_DEVICE	
PICKUP_DEVICE	
SPLITTING_DEVICE	
TRANSFER_CONTROLLER_DEVICE	
TRANSFERRED_DEVICE	
NEW_DESTINATION_DEVICE	
DROPPED_DEVICE	
USING_DEVICE	
HOLDING_DEVICE	
WAITING_DEVICE	

Referenced By

- Complex Type ns1:device

Element: ns1:deviceType [type partyMember]

Derived By

Type ns1:deviceType

Enumeration

Value	Description
UNKNOWN	
SIP_PRIVATE_TRUNK	
SIP_PUBLIC_TRUNK	

Value	Description
MULTICALL_GROUP	
HUNT_GROUP	
KEYLINE	
VOICE_SET_PRIME	
ACD_STRUCTURE	
LINE_APPEARANCE	
DATA_EXTENSION	
ACD2_PATH	
ACD2_AGENT	
ACD2_GROUP	
ROUTING_DEVICE	
ACD2_REMOTE_SUBGROUP	
RING_ALL_GROUP	
RING_CASCADE_GROUP	
ACDX_AGENT	
ACDX_GROUP	
ACDX_PATH	
PERSONAL_RING_GROUP	
PEER	
EXTERNAL_DEVICE_TYPE	
INTERNAL_DEVICE_TYPE	

Referenced By

- Complex Type ns1:partyMember

Element: ns1:devNumber [type featureEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:featureEventMsg

Element: tns:digits [element outPulseDigits]

Derived By

Type xs:string

Referenced By

- Element tns:outPulseDigits

Element: tns:dn [element sendCallMeBackMsgNoCall]

Derived By

Type xs:string

Referenced By

- Element tns:sendCallMeBackMsgNoCall

Element: ns1:dn [type forwardingInfo]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:forwardingInfo

Element: ns2:dn [type setCFRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:setCFRequest

Element: tns:dndState [element setDeviceDND]

Derived By

Type xs:boolean

Referenced By

- Element tns:setDeviceDND

Element: ns2:dtmfDigits [element outPulseDigitsRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:outPulseDigitsRequest

Element: ns1:errorDescription [type callStatusResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:callStatusResult

Element: ns2:errorDescription [type deviceConfigResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:deviceConfigResult

Element: ns1:errorDescription [type deviceFeaturesResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns2:errorDescription [type hotDeskUserLoginDeviceResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:hotDeskUserLoginDeviceResult

Element: ns2:errorDescription [type hotDeskUserLoginResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:hotDeskUserLoginResult

Element: ns1:errorDescription [type icpIdResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:icpIdResult

Element: ns2:errorDescription [type monitorResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:monitorResult

Element: ns1:errorDescription [type objectIdResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:objectIdResult

Element: ns1:errorDescription [type result]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:result

Element: ns1:errorDescription [type standardEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns2:eventHandlerURL [element registerEventHandler]

Derived By

Type xs:string

Referenced By

- Element ns2:registerEventHandler

Element: tns:eventHandlerURL [element registerEventHandler]

Derived By

Type xs:string

Referenced By

- Element tns:registerEventHandler

Element: ns1:eventType [type featureEventMsg]

Derived By

Type ns1:featureEventType

Enumeration

Value	Description
UNKNOWN	
AUTO_ANSWER_FEATURE	
DO_NOT_DISTURB	
MSG_WAITING_INDICATOR	
INSERVICE	
DO_NOT_DISTURB_ALL	
MSG_WAITING_IND_ALL	
MAKE_BUSY	
JOIN_GROUP	
LEAVE_GROUP	
JOIN_ALL_ACD_GROUPS	
MAKE_BUSY_STATE	
LEAVE_All_ACD_GROUPS	
ACD_AGENT	
ACD_AGENT_BLOCKED	
ACD2_LOGIN_NO_LICENSE	
HOT_DESK_USER	
CF_ALWAYS	
CF_BUSY_EXTERNAL	
CF_BUSY_INTERNAL	
CF_NO_ANSWER_INTERNAL	
CF_NO_ANSWER_EXTERNAL	
CF_NO_ANSWER_BUSY	
CF_ALL_MODES	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CONFERENCE_FEATURE	
FORWARD_FEATURE	
USER_ID_FEATURE_EVENT	

Referenced By

- Complex Type ns1:featureEventMsg

Element: ns1:eventType [type standardEventMsg]

Derived By

Type ns1:eventType

Enumeration

Value	Description
UNKNOWN	
CALL_EVENT	
FEATURE_EVENT	
ACD_EVENT	
CONFERENCE_EVENT	
SYSTEM_EVENT	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:eventType [type systemEventMsg]

Derived By

Type ns1:systemEventType

Enumeration

Value	Description
ICP_COMMUNICATION_EVENT	
OIG_GW_SHUTDOWN	

Referenced By

- Complex Type ns1:systemEventMsg

Element: ns2:extNumber [type lineConfig]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:lineConfig

Element: ns1:featureEvent [type standardEventMsg]

Derived By

Type ns1:featureEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:time	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:state	ns1:callState	0..1	
ns1:devNumber	xs:string	0..1	
ns1:eventType	ns1:featureEventType	0..1	
ns1:userDn	xs:string	0..1	
ns1:groupDn	xs:string	0..1	
ns1:forwardDn	xs:string	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:featuresAllowed [type callEventMsg]

Derived By

Type ns1:featuresAllowed

Enumeration

Value	Description
NOT_SUPPORTED	
ANSWER_CALL	
REDIRECT_CALL	
CAMPON_CALL	
SET_CALL_ME_BACK	
INTRUDE_CALL	
ALTERNATE_CALL	
TRADE_CALL	
CONFERENCE_CALL	
SPLIT_CONFERENCE_CALL	
CONSULTATION_CALL	

Value	Description
CANCEL_CONSULTATION_CALL	
TRANSFER_CALL	
SEND_CALLBACK_MESSAGE	
SET_ACCOUNT_CODE	
NEW_CALL	
CLEAR_CALL	
OUT_PULSE_DIGITS	
HOLD_CALL	
RETRIEVE_CALL	
LISTEN_CALL	
ASSIGN_CALLER_ID	
ALTER_WORK_TIMER	
TAP_CALL	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:forwardDn [type featureEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:featureEventMsg

Element: tns:getCallStatus

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:getCallStatusResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:callStatusResult	0..1	

Element: tns:getDeviceConfiguration

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:getDeviceConfigurationRequest	0..1	

Element: ns2:getDeviceConfigurationRequest

Derived By

Type ns2:getDeviceConfigurationRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Element: tns:getDeviceConfigurationResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns2:deviceConfigResult	0..1	

Element: tns:getDeviceFeatures

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:getDeviceFeaturesResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:deviceFeaturesResult	0..1	

Element: tns:getEvent

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:timeout	xs:int	1..1	

Element: tns:getEventResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:standardEventMsg	0..1	

Element: tns:getHotDeskUserDn

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:hotDeskDeviceObjectId	xs:long	1..1	

Element: tns:getHotDeskUserDnResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns2:hotDeskUserLoginResult	0..1	

Element: tns:getHotDeskUserLoginDevice

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:hotDeskUserObjectId	xs:long	1..1	

Element: tns:getHotDeskUserLoginDeviceResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns2:hotDeskUserLoginDeviceResult	0..1	

Element: tns:getIcpId

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:getIcpIdRequest	0..1	

Element: ns2:getIcpIdRequest

Derived By

Type ns2:getIcpIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:icpIpAddress	xs:string	1..1	

Element: tns:getIcpIdResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:icpIdResult	0..1	

Element: tns:getLineAppearanceId

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:getLineAppearanceIdRequest	0..1	

Element: ns2:getLineAppearanceIdRequest

Derived By

Type ns2:getLineAppearanceIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		
ns2:buttonNum	xs:int	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Element: tns:getLineAppearanceIdResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:objectIdResult	0..1	

Element: tns:getPhoneNumberId

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:getPhoneNumberIdRequest	0..1	

Element: ns2:getPhoneNumberIdRequest

Derived By

Type ns2:getPhoneNumberIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Element: tns:getPhoneNumberIdResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:objectIdResult	0..1	

Element: ns1:groupDn [type featureEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:featureEventMsg

Element: tns:holdCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:holdCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns2:hotDeskDeviceDn [type hotDeskUserLoginDeviceResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:hotDeskUserLoginDeviceResult

Element: tns:hotDeskDeviceObjectId [element getHotDeskUserDn]

Derived By

Type xs:long

Referenced By

- Element tns:getHotDeskUserDn

Element: tns:hotDeskUserDn [element verifyHotDeskUserPin]

Derived By

Type xs:string

Referenced By

- Element tns:verifyHotDeskUserPin

Element: ns2:hotDeskUserDn [type loginHotDeskUserRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:loginHotDeskUserRequest

Element: ns1:hotDeskUserLoggedInDn [type deviceFeaturesResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: tns:hotDeskUserObjectId [element getHotDeskUserLoginDevice]

Derived By

Type xs:long

Referenced By

- Element tns:getHotDeskUserLoginDevice

Element: tns:icpId [element monitorPRGPresence]

Derived By

Type xs:long

Referenced By

- Element tns:monitorPRGPresence

Element: tns:icpId [element stopPRGPresenceMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopPRGPresenceMonitor

Element: ns2:icpIpAddress [type getIcpIdRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:getIcpIdRequest

Element: ns1:icpVersion [type icpIdResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:icpIdResult

Element: tns:invokerDn [element setPRGPresence]

Derived By

Type xs:string

Referenced By

- Element tns:setPRGPresence

Element: ns2:line [type deviceConfigResult]

Derived By

Type ns2:lineConfig

Attributes

Name	Type	Required?	Default	Description
ns2:objectId	xs:long	Yes		
ns2:buttonNumber	xs:int	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:extNumber	xs:string	0..1	

Referenced By

- Complex Type ns2:deviceConfigResult

Element: tns:localCallId [element alternateCall]

Derived By

Type xs:long

Referenced By

- Element tns:alternateCall

Element: tns:localCallId [element answerCall]

Derived By

Type xs:long

Referenced By

- Element tns:answerCall

Element: tns:localCallId [element camponCall]

Derived By

Type xs:long

Referenced By

- Element tns:camponCall

Element: tns:localCallId [element cancelConsCall]

Derived By

Type xs:long

Referenced By

- Element tns:cancelConsCall

Element: tns:localCallId [element clearCall]

Derived By

Type xs:long

Referenced By

- Element tns:clearCall

Element: tns:localCallId [element conferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:conferenceCall

Element: tns:localCallId [element consultationCall]

Derived By

Type xs:long

Referenced By

- Element tns:consultationCall

Element: tns:localCallId [element holdCall]

Derived By

Type xs:long

Referenced By

- Element tns:holdCall

Element: tns:localCallId [element newCall]

Derived By

Type xs:long

Referenced By

- Element tns:newCall

Element: tns:localCallId [element outPulseDigits]

Derived By

Type xs:long

Referenced By

- Element tns:outPulseDigits

Element: tns:localCallId [element redirectCall]

Derived By

Type xs:long

Referenced By

- Element tns:redirectCall

Element: tns:localCallId [element retrieveCall]

Derived By

Type xs:long

Referenced By

- Element tns:retrieveCall

Element: tns:localCallId [element sendCallMeBackMsgForCall]

Derived By

Type xs:long

Referenced By

- Element tns:sendCallMeBackMsgForCall

Element: tns:localCallId [element setAccountCode]

Derived By

Type xs:long

Referenced By

- Element tns:setAccountCode

Element: tns:localCallId [element setCallMeBack]

Derived By

Type xs:long

Referenced By

- Element tns:setCallMeBack

Element: tns:localCallId [element splitConferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:splitConferenceCall

Element: tns:localCallId [element tradeCall]

Derived By

Type xs:long

Referenced By

- Element tns:tradeCall

Element: tns:localCallId [element transferCall]

Derived By

Type xs:long

Referenced By

- Element tns:transferCall

Element: tns:loginExtHotDeskUser

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:loginExtHotDeskUserRequest	0..1	

Element: ns2:loginExtHotDeskUserRequest

Derived By

Type ns2:loginExtHotDeskUserRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	

Element: tns:loginExtHotDeskUserResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:loginHotDeskUser

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:loginHotDeskUserRequest	0..1	

Element: ns2:loginHotDeskUserRequest

Derived By

Type ns2:loginHotDeskUserRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	
ns2:hotDeskUserDn	xs:string	1..1	

Element: tns:loginHotDeskUserResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:logoutExtHotDeskUser

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:logoutExtHotDeskUserResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:logoutHotDeskUser

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:logoutHotDeskUserResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:makeCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:number	xs:string	0..1	
tns:accountCode	xs:string	0..1	

Element: ns2:makeCallRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:number	xs:string	0..1	
ns2:accountCode	xs:string	0..1	

Element: tns:makeCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:memberName [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: ns1:memberNetworkExt [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: ns1:memberNum [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: tns:monitorFeatures

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:monitorFeaturesResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns2:monitorResult	0..1	

Element: tns:monitorObject

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:monitorObjectResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns2:monitorResult	0..1	

Element: tns:monitorPRGPresence

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:icpId	xs:long	1..1	

Element: tns:monitorPRGPresenceResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:newCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:newCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:number [element consultationCall]

Derived By

Type xs:string

Referenced By

- Element tns:consultationCall

Element: tns:number [element makeCall]

Derived By

Type xs:string

Referenced By

- Element tns:makeCall

Element: ns2:number [element makeCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:makeCallRequest

Element: tns:objectId [element alternateCall]

Derived By

Type xs:long

Referenced By

- Element tns:alternateCall

Element: tns:objectId [element answerCall]

Derived By

Type xs:long

Referenced By

- Element tns:answerCall

Element: tns:objectId [element camponCall]

Derived By

Type xs:long

Referenced By

- Element tns:camponCall

Element: tns:objectId [element cancelConsCall]

Derived By

Type xs:long

Referenced By

- Element tns:cancelConsCall

Element: tns:objectId [element clearCall]

Derived By

Type xs:long

Referenced By

- Element tns:clearCall

Element: tns:objectId [element conferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:conferenceCall

Element: tns:objectId [element consultationCall]

Derived By

Type xs:long

Referenced By

- Element tns:consultationCall

Element: tns:objectId [element getCallStatus]

Derived By

Type xs:long

Referenced By

- Element tns:getCallStatus

Element: tns:objectId [element getDeviceFeatures]

Derived By

Type xs:long

Referenced By

- Element tns:getDeviceFeatures

Element: tns:objectId [element holdCall]

Derived By

Type xs:long

Referenced By

- Element tns:holdCall

Element: tns:objectId [element logoutExtHotDeskUser]

Derived By

Type xs:long

Referenced By

- Element tns:logoutExtHotDeskUser

Element: tns:objectId [element logoutHotDeskUser]

Derived By

Type xs:long

Referenced By

- Element tns:logoutHotDeskUser

Element: tns:objectId [element makeCall]

Derived By

Type xs:long

Referenced By

- Element tns:makeCall

Element: tns:objectId [element monitorFeatures]

Derived By

Type xs:long

Referenced By

- Element tns:monitorFeatures

Element: tns:objectId [element monitorObject]

Derived By

Type xs:long

Referenced By

- Element tns:monitorObject

Element: tns:objectId [element newCall]

Derived By

Type xs:long

Referenced By

- Element tns:newCall

Element: tns:objectId [element outPulseDigits]

Derived By

Type xs:long

Referenced By

- Element tns:outPulseDigits

Element: tns:objectId [element pickupCall]

Derived By

Type xs:long

Referenced By

- Element tns:pickupCall

Element: tns:objectId [element redirectCall]

Derived By

Type xs:long

Referenced By

- Element tns:redirectCall

Element: tns:objectId [element retrieveCall]

Derived By

Type xs:long

Referenced By

- Element tns:retrieveCall

Element: tns:objectId [element setAccountCode]

Derived By

Type xs:long

Referenced By

- Element tns:setAccountCode

Element: tns:objectId [element setCallMeBack]

Derived By

Type xs:long

Referenced By

- Element tns:setCallMeBack

Element: tns:objectId [element setDeviceDND]

Derived By

Type xs:long

Referenced By

- Element tns:setDeviceDND

Element: tns:objectId [element setPRGPresence]

Derived By

Type xs:long

Referenced By

- Element tns:setPRGPresence

Element: tns:objectId [element splitConferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:splitConferenceCall

Element: tns:objectId [element stopFeatureMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopFeatureMonitor

Element: tns:objectId [element stopMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopMonitor

Element: tns:objectId [element tradeCall]

Derived By

Type xs:long

Referenced By

- Element tns:tradeCall

Element: tns:objectId [element transferCall]

Derived By

Type xs:long

Referenced By

- Element tns:transferCall

Element: tns:objectId [element verifyHotDeskUserPin]

Derived By

Type xs:long

Referenced By

- Element tns:verifyHotDeskUserPin

Element: tns:objId [element sendCallMeBackMsgForCall]

Derived By

Type xs:long

Referenced By

- Element tns:sendCallMeBackMsgForCall

Element: tns:objId [element sendCallMeBackMsgNoCall]

Derived By

Type xs:long

Referenced By

- Element tns:sendCallMeBackMsgNoCall

Element: tns:objIdOfDeviceWithMsg [element clearCallMeBackMsg]

Derived By

Type xs:long

Referenced By

- Element tns:clearCallMeBackMsg

Element: tns:outPulseDigits

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	
tns:digits	xs:string	0..1	

Element: ns2:outPulseDigitsRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dtmfDigits	xs:string	1..1	

Element: tns:outPulseDigitsResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:party [type callEventMsg]

Derived By

Type ns1:partyMember

Attributes

Name	Type	Required?	Default	Description
ns1:callId	xs:long	Yes		
ns1:numberPrivacy	xs:boolean	Yes		
ns1:namePrivacy	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:memberNum	xs:string	0..1	
ns1:memberName	xs:string	0..1	
ns1:deviceType	ns1:deviceType	0..1	
ns1:remoteMemberExt	xs:string	0..1	
ns1:remoteMemberName	xs:string	0..1	
ns1:memberNetworkExt	xs:string	0..1	
ns1:peerId	xs:string	0..1	
ns1:peerName	xs:string	0..1	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:peerId [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: ns1:peerName [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: tns:pickupCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:ringDn	xs:string	0..1	

Element: ns2:pickupCallRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

	Type	Occurs	Description
SEQUENCE		1..1	
ns2:ringDn	xs:string	1..1	
ns2:pickupDn	xs:string	1..1	

Element: tns:pickupCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns2:pickupDn [element pickupCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:pickupCallRequest

Element: tns:pin [element verifyHotDeskUserPin]

Derived By

Type xs:string

Referenced By

- Element tns:verifyHotDeskUserPin

Element: ns2:pin [type loginExtHotDeskUserRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:loginExtHotDeskUserRequest

Element: ns2:pin [type loginHotDeskUserRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:loginHotDeskUserRequest

Element: tns:presenceValue [element setPRGPresence]

Derived By

Type xs:boolean

Referenced By

- Element tns:setPRGPresence

Element: tns:PRGDn [element setPRGPresence]

Derived By

Type xs:string

Referenced By

- Element tns:setPRGPresence

Element: ns2:PRGDn [element setPRGPresenceRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:setPRGPresenceRequest

Element: ns2:primaryIcp [type deviceConfigResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:deviceConfigResult

Element: ns2:primeDn [type getDeviceConfigurationRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:getDeviceConfigurationRequest

Element: ns2:primeDn [type getLineAppearanceIdRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:getLineAppearanceIdRequest

Element: ns2:primeDn [type getPhoneNumberIdRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:getPhoneNumberIdRequest

Element: tns:redirectCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	
tns:redirectDn	xs:string	0..1	

Element: ns2:redirectCallRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:session	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:localCallId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:redirectDn	xs:string	1..1	

Element: tns:redirectCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:redirectDn [element redirectCall]

Derived By

Type xs:string

Referenced By

- Element tns:redirectCall

Element: ns2:redirectDn [element redirectCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:redirectCallRequest

Element: ns2:registerEventHandler

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:eventHandlerURL	xs:string	1..1	

Element: tns:registerEventHandler

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:eventHandlerURL	xs:string	0..1	

Element: tns:registerEventHandlerResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:registrationDn [type deviceFeaturesResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:deviceFeaturesResult

Element: ns1:remoteMemberExt [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: ns1:remoteMemberName [type partyMember]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:partyMember

Element: tns:request [element getDeviceConfiguration]

Derived By

Type ns2:getDeviceConfigurationRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element tns:getDeviceConfiguration

Element: tns:request [element getIcpId]

Derived By

Type ns2:getIcpIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:icpIpAddress	xs:string	1..1	

Referenced By

- Element tns:getIcpId

Element: tns:request [element getLineAppearanceId]

Derived By

Type ns2:getLineAppearanceIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		
ns2:buttonNum	xs:int	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element tns:getLineAppearanceId

Element: tns:request [element getPhoneNumberId]

Derived By

Type ns2:getPhoneNumberIdRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:primeDn	xs:string	1..1	

Referenced By

- Element tns:getPhoneNumberId

Element: tns:request [element loginExtHotDeskUser]

Derived By

Type ns2:loginExtHotDeskUserRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	

Referenced By

- Element tns:loginExtHotDeskUser

Element: tns:request [element loginHotDeskUser]

Derived By

Type ns2:loginHotDeskUserRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:pin	xs:string	1..1	
ns2:hotDeskUserDn	xs:string	1..1	

Referenced By

- Element tns:loginHotDeskUser

Element: tns:request [element setCFAlways]

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:setCFAlways

Element: tns:request [element setCFBusyExternal]

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:setCFBusyExternal

Element: tns:request [element setCFBusyInternal]

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:setCFBusyInternal

Element: tns:request [element setCFNAExternal]

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:setCFNAExternal

Element: tns:request [element setCFNAInternal]

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Referenced By

- Element tns:setCFNAInternal

Element: tns:retrieveCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:retrieveCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:return [element alternateCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:alternateCallResponse

Element: tns:return [element answerCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:answerCallResponse

Element: tns:return [element camponCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:camponCallResponse

Element: tns:return [element cancelConsCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:cancelConsCallResponse

Element: tns:return [element clearCallMeBackMsgResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:clearCallMeBackMsgResponse

Element: tns:return [element clearCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:clearCallResponse

Element: tns:return [element conferenceCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:conferenceCallResponse

Element: tns:return [element consultationCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:consultationCallResponse

Element: tns:return [element getCallStatusResponse]

Derived By

Type ns1:callStatusResult

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:callEvent	ns1:callEventMsg	0..1	

Referenced By

- Element tns:getCallStatusResponse

Element: tns:return [element getDeviceConfigurationResponse]

Derived By

Type ns2:deviceConfigResult

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:isResilient	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:line	ns2:lineConfig	0..*	
ns2:primaryIcp	xs:string	0..1	
ns2:secondaryIcp	xs:string	0..1	

Referenced By

- Element tns:getDeviceConfigurationResponse

Element: tns:return [element getDeviceFeaturesResponse]

Derived By

Type ns1:deviceFeaturesResult

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:DND	xs:boolean	Yes		
ns1:autoAnswer	xs:boolean	Yes		
ns1:msgWaitingLamp	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:hotDeskUserLoggedInDn	xs:string	0..1	
ns1:registrationDn	xs:string	0..1	
ns1:CFNAExt	ns1:forwardingInfo	0..1	
ns1:CFNAInt	ns1:forwardingInfo	0..1	
ns1:CFBusyExt	ns1:forwardingInfo	0..1	
ns1:CFBusyInt	ns1:forwardingInfo	0..1	
ns1:CFAlways	ns1:forwardingInfo	0..1	

Referenced By

- Element tns:getDeviceFeaturesResponse

Element: tns:return [element getEventResponse]

Derived By

Type ns1:standardEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:eventType	ns1:eventType	0..1	
ns1:callEvent	ns1:callEventMsg	0..1	
ns1:featureEvent	ns1:featureEventMsg	0..1	
ns1:systemEvent	ns1:systemEventMsg	0..1	

Referenced By

- Element tns:getEventResponse

Element: tns:return [element getHotDeskUserDnResponse]

Derived By

Type ns2:hotDeskUserLoginResult

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:Boolean	Yes		
ns2:hotDeskDeviceObjectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:userDn	xs:string	0..1	

Referenced By

- Element tns:getHotDeskUserDnResponse

Element: tns:return [element getHotDeskUserLoginDeviceResponse]

Derived By

Type ns2:hotDeskUserLoginDeviceResult

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:hotDeskUserObjectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	
ns2:hotDeskDeviceDn	xs:string	0..1	

Referenced By

- Element tns:getHotDeskUserLoginDeviceResponse

Element: tns:return [element getIcpIdResponse]

Derived By

Type ns1:icpIdResult

Attributes

Name	Type	Required?	Default	Description
ns1:result	Xs:boolean	Yes		
ns1:icpId	Xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:connectionState	ns1:connectionState	0..1	
ns1:icpVersion	xs:string	0..1	

Referenced By

- Element tns:getIcpIdResponse

Element: tns:return [element getLineAppearanceIdResponse]

Derived By

Type ns1:objectIdResult

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:getLineAppearanceIdResponse

Element: tns:return [element getPhoneNumberIdResponse]

Derived By

Type ns1:objectIdResult

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:getPhoneNumberIdResponse

Element: tns:return [element holdCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:holdCallResponse

Element: tns:return [element loginExtHotDeskUserResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:loginExtHotDeskUserResponse

Element: tns:return [element loginHotDeskUserResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:loginHotDeskUserResponse

Element: tns:return [element logoutExtHotDeskUserResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:logoutExtHotDeskUserResponse

Element: tns:return [element logoutHotDeskUserResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:logoutHotDeskUserResponse

Element: tns:return [element makeCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:makeCallResponse

Element: tns:return [element monitorFeaturesResponse]

Derived By

Type ns2:monitorResult

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:objectId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	

Referenced By

- Element tns:monitorFeaturesResponse

Element: tns:return [element monitorObjectResponse]

Derived By

Type ns2:monitorResult

Attributes

Name	Type	Required?	Default	Description
ns2:result	xs:boolean	Yes		
ns2:objectId	xs:long	Yes		
ns2:icpId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:errorDescription	xs:string	0..1	

Referenced By

- Element tns:monitorObjectResponse

Element: tns:return [element monitorPRGPresenceResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:monitorPRGPresenceResponse

Element: tns:return [element newCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:newCallResponse

Element: tns:return [element outPulseDigitsResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:outPulseDigitsResponse

Element: tns:return [element pickupCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:pickupCallResponse

Element: tns:return [element redirectCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:redirectCallResponse

Element: tns:return [element registerEventHandlerResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:registerEventHandlerResponse

Element: tns:return [element retrieveCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:retrieveCallResponse

Element: tns:return [element sendCallMeBackMsgForCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:sendCallMeBackMsgForCallResponse

Element: tns:return [element sendCallMeBackMsgNoCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:sendCallMeBackMsgNoCallResponse

Element: tns:return [element setAccountCodeResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setAccountCodeResponse

Element: tns:return [element setCallMeBackResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCallMeBackResponse

Element: tns:return [element setCFAlwaysResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCFAlwaysResponse

Element: tns:return [element setCFBusyExternalResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCFBusyExternalResponse

Element: tns:return [element setCFBusyInternalResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCFBusyInternalResponse

Element: tns:return [element setCFNAExternalResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCFNAExternalResponse

Element: tns:return [element setCFNAInternalResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setCFNAInternalResponse

Element: tns:return [element setDeviceDNDResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setDeviceDNDResponse

Element: tns:return [element setPRGPresenceResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:setPRGPresenceResponse

Element: tns:return [element splitConferenceCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:splitConferenceCallResponse

Element: tns:return [element stopFeatureMonitorResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:Boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:stopFeatureMonitorResponse

Element: tns:return [element stopMonitorResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	Xs:string	0..1	

Referenced By

- Element tns:stopMonitorResponse

Element: tns:return [element stopPRGPresenceMonitorResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	Xs:string	0..1	

Referenced By

- Element tns:stopPRGPresenceMonitorResponse

Element: tns:return [element tradeCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	Xs:string	0..1	

Referenced By

- Element tns:tradeCallResponse

Element: tns:return [element transferCallResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	Xs:string	0..1	

Referenced By

- Element tns:transferCallResponse

Element: tns:return [element verifyHotDeskUserPinResponse]

Derived By

Type ns1:result

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:verifyHotDeskUserPinResponse

Element: tns:ringDn [element pickupCall]

Derived By

Type xs:string

Referenced By

- Element tns:pickupCall

Element: ns2:ringDn [element pickupCallRequest]

Derived By

Type xs:string

Referenced By

- Element ns2:pickupCallRequest

Element: ns2:secondaryIcp [type deviceConfigResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:deviceConfigResult

Element: tns:sendCallMeBackMsgForCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:sendCallMeBackMsgForCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:sendCallMeBackMsgNoCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objId	xs:long	1..1	
tns:dn	xs:string	0..1	

Element: ns2:sendCallMeBackMsgNoCallRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objIdOfDeviceToMsg	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:deviceDn	xs:string	1..1	

Element: tns:sendCallMeBackMsgNoCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:sessionId [element alternateCall]

Derived By

Type xs:long

Referenced By

- Element tns:alternateCall

Element: tns:sessionId [element answerCall]

Derived By

Type xs:long

Referenced By

- Element tns:answerCall

Element: tns:sessionId [element camponCall]

Derived By

Type xs:long

Referenced By

- Element tns:camponCall

Element: tns:sessionId [element cancelConsCall]

Derived By

Type xs:long

Referenced By

- Element tns:cancelConsCall

Element: tns:sessionId [element clearCall]

Derived By

Type xs:long

Referenced By

- Element tns:clearCall

Element: tns:sessionId [element clearCallMeBackMsg]

Derived By

Type xs:long

Referenced By

- Element tns:clearCallMeBackMsg

Element: tns:sessionId [element conferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:conferenceCall

Element: tns:sessionId [element consultationCall]

Derived By

Type xs:long

Referenced By

- Element tns:consultationCall

Element: tns:sessionId [element getCallStatus]

Derived By

Type xs:long

Referenced By

- Element tns:getCallStatus

Element: tns:sessionId [element getDeviceFeatures]

Derived By

Type xs:long

Referenced By

- Element tns:getDeviceFeatures

Element: tns:sessionId [element getEvent]

Derived By

Type xs:long

Referenced By

- Element tns:getEvent

Element: tns:sessionId [element getHotDeskUserDn]

Derived By

Type xs:long

Referenced By

- Element tns:getHotDeskUserDn

Element: tns:sessionId [element getHotDeskUserLoginDevice]

Derived By

Type xs:long

Referenced By

- Element tns:getHotDeskUserLoginDevice

Element: tns:sessionId [element holdCall]

Derived By

Type xs:long

Referenced By

- Element tns:holdCall

Element: tns:sessionId [element logoutExtHotDeskUser]

Derived By

Type xs:long

Referenced By

- Element tns:logoutExtHotDeskUser

Element: tns:sessionId [element logoutHotDeskUser]

Derived By

Type xs:long

Referenced By

- Element tns:logoutHotDeskUser

Element: tns:sessionId [element makeCall]

Derived By

Type xs:long

Referenced By

- Element tns:makeCall

Element: tns:sessionId [element monitorFeatures]

Derived By

Type xs:long

Referenced By

- Element tns:monitorFeatures

Element: tns:sessionId [element monitorObject]

Derived By

Type xs:long

Referenced By

- Element tns:monitorObject

Element: tns:sessionId [element monitorPRGPresence]

Derived By

Type xs:long

Referenced By

- Element tns:monitorPRGPresence

Element: tns:sessionId [element newCall]

Derived By

Type xs:long

Referenced By

- Element tns:newCall

Element: tns:sessionId [element outPulseDigits]

Derived By

Type xs:long

Referenced By

- Element tns:outPulseDigits

Element: tns:sessionId [element pickupCall]

Derived By

Type xs:long

Referenced By

- Element tns:pickupCall

Element: tns:sessionId [element redirectCall]

Derived By

Type xs:long

Referenced By

- Element tns:redirectCall

Element: tns:sessionId [element registerEventHandler]

Derived By

Type xs:long

Referenced By

- Element tns:registerEventHandler

Element: tns:sessionId [element retrieveCall]

Derived By

Type xs:long

Referenced By

- Element tns:retrieveCall

Element: tns:sessionId [element sendCallMeBackMsgForCall]

Derived By

Type xs:long

Referenced By

- Element tns:sendCallMeBackMsgForCall

Element: tns:sessionId [element sendCallMeBackMsgNoCall]

Derived By

Type xs:long

Referenced By

- Element tns:sendCallMeBackMsgNoCall

Element: tns:sessionId [element setAccountCode]

Derived By

Type xs:long

Referenced By

- Element tns:setAccountCode

Element: tns:sessionId [element setCallMeBack]

Derived By

Type xs:long

Referenced By

- Element tns:setCallMeBack

Element: tns:sessionId [element setDeviceDND]

Derived By

Type xs:long

Referenced By

- Element tns:setDeviceDND

Element: tns:sessionId [element setPRGPresence]

Derived By

Type xs:long

Referenced By

- Element tns:setPRGPresence

Element: tns:sessionId [element splitConferenceCall]

Derived By

Type xs:long

Referenced By

- Element tns:splitConferenceCall

Element: tns:sessionId [element stopFeatureMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopFeatureMonitor

Element: tns:sessionId [element stopMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopMonitor

Element: tns:sessionId [element stopPRGPresenceMonitor]

Derived By

Type xs:long

Referenced By

- Element tns:stopPRGPresenceMonitor

Element: tns:sessionId [element tradeCall]

Derived By

Type xs:long

Referenced By

- Element tns:tradeCall

Element: tns:sessionId [element transferCall]

Derived By

Type xs:long

Referenced By

- Element tns:transferCall

Element: tns:sessionId [element verifyHotDeskUserPin]

Derived By

Type xs:long

Referenced By

- Element tns:verifyHotDeskUserPin

Element: tns:setAccountCode

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	
tns:accCode	xs:string	0..1	

Element: ns2:setAccountCodeRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:localCallId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:accountCode	xs:string	1..1	

Element: tns:setAccountCodeResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCallMeBack

Derived *By*

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:setCallMeBackResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCFAlways

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:setCFRequest	0..1	

Element: tns:setCFAlwaysResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCFBusyExternal

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:setCFRequest	0..1	

Element: tns:setCFBusyExternalResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCFBusyInternal

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:setCFRequest	0..1	

Element: tns:setCFBusyInternalResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCFNAExternal

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:setCFRequest	0..1	

Element: tns:setCFNAExternalResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setCFNAInternal

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:request	ns2:setCFRequest	0..1	

Element: tns:setCFNAInternalResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns2:setCFRequest

Derived By

Type ns2:setCFRequest

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:cfOn	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:dn	xs:string	1..1	

Element: tns:setDeviceDND

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:dndState	xs:boolean	1..1	

Element: tns:setDeviceDNDResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:setPRGPresence

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:presenceValue	xs:boolean	1..1	
tns:PRGDn	xs:string	0..1	
tns:invokerDn	xs:string	0..1	

Element: ns2:setPRGPresenceRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns2:sessionId	xs:long	Yes		
ns2:objectId	xs:long	Yes		
ns2:presenceValue	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns2:PRGDn	xs:string	1..1	

Element: tns:setPRGPresenceResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:splitConferenceCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:splitConferenceCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns2:standardCCService

Derived By

Restricting xs:anyType

Content Model

Always empty.

Element: ns1:state [type featureEventMsg]

Derived By

Type ns1:callState

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	
HELD	
IDLE	
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Complex Type ns1:featureEventMsg

Element: tns:stopFeatureMonitor

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:stopFeatureMonitorResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:stopMonitor

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	

Element: tns:stopMonitorResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:stopPRGPresenceMonitor

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:icpId	xs:long	1..1	

Element: tns:stopPRGPresenceMonitorResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:systemEvent [type standardEventMsg]

Derived By

Type ns1:systemEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:icpId	xs:long	Yes		
ns1:time	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:eventType	ns1:systemEventType	0..1	
ns1:connectionState	ns1:connectionState	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: tns:timeout [element getEvent]

Derived By

Type xs:int

Referenced By

- Element tns:getEvent

Element: tns:tradeCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:tradeCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: tns:transferCall

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:localCallId	xs:long	1..1	

Element: tns:transferCallResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Element: ns1:type [type callEventMsg]

Derived By

Type ns1:callEventType

Enumeration

Value	Description
UNKNOWN	

Value	Description
ACCOUNT_CODE_SET	
ACD2_PATH	
ACD2_GROUP	
GROUP	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CALL_CLEAR	
CALL_CONFERENCED	
CALL_DELIVERED	
CALL_DIVERTED	
CALL_ESTABLISHED	
CALL_FAILED	
CALL_HELD	
CALL_ORIGINATED	
CALL_QUEUED	
CALL_RECEIVED	
CALL_RETRIEVED	
CALL_TRANSFERRED	
CMD_RESPONSE	
CONFERENCE_HELD	
CONFERENCE_FEATURE	
DEVICE_DROPPED	
EXTENSION_IN_USE	
FORWARD_FEATURE	
IN_SERVICE	
MONITOR_SET	
OUT_OF_SERVICE	
REMOTE_PARTY_UPDATE	
RESILIENT_DEVICE	
ROUTING_DEVICE	
TRUNK_DIGITS_OUTPULSED	
WORK_TIMER_EXPIRED	
USERID_FEATURE	
GROUP_PRESENCE_FEATURE	
ACD_EXPRESS_GROUP_EVENT	

Value	Description
MONITOR_FAILED_EVENT	
PRG_MONITOR_SET_EVENT	
PRG_MONITOR_FAILED_EVENT	

Referenced By

- Complex Type ns1:callEventMsg

Element: ns1:userDn [type featureEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:featureEventMsg

Element: ns2:userDn [type hotDeskUserLoginResult]

Derived By

Type xs:string

Referenced By

- Complex Type ns2:hotDeskUserLoginResult

Element: tns:verifyHotDeskUserPin

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	
tns:objectId	xs:long	1..1	
tns:hotDeskUserDn	xs:string	0..1	
tns:pin	xs:string	0..1	

Element: tns:verifyHotDeskUserPinResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:result	0..1	

Attributes: Standardccservice

Attributes

Name	Description
ns1:autoAnswer [type deviceFeaturesResult]	
ns2:buttonNum [type getLineAppearanceIdRequest]	
ns2:buttonNumber [type lineConfig]	
ns1:callEventTime [type callEventMsg]	
ns1:callId [type partyMember]	
ns2:cfOn [type setCFRequest]	
ns1:DND [type deviceFeaturesResult]	
ns1:fwdOn [type forwardingInfo]	
ns2:hotDeskDeviceObjectId [type hotDeskUserLoginResult]	
ns2:hotDeskUserObjectId [type hotDeskUserLoginDeviceResult]	
ns2:icpId [type getDeviceConfigurationRequest]	
ns2:icpId [type getLineAppearanceIdRequest]	
ns2:icpId [type getPhoneNumberIdRequest]	
ns1:icpId [type icpIdResult]	
ns2:icpId [type monitorResult]	
ns1:icpId [type systemEventMsg]	
ns2:isResilient [type deviceConfigResult]	
ns2:localCallId [element redirectCallRequest]	
ns2:localCallId [element setAccountCodeRequest]	
ns1:localCallId [type callEventMsg]	

Name	Description
ns1:msgWaitingLamp [type deviceFeaturesResult]	
ns1:namePrivacy [type partyMember]	
ns1:numberPrivacy [type partyMember]	
ns2:objectId [element makeCallRequest]	
ns2:objectId [element outPulseDigitsRequest]	
ns2:objectId [element pickupCallRequest]	
ns2:objectId [element redirectCallRequest]	
ns2:objectId [element setAccountCodeRequest]	
ns2:objectId [element setPRGPresenceRequest]	
ns1:objectId [type callEventMsg]	
ns1:objectId [type featureEventMsg]	
ns2:objectId [type lineConfig]	
ns2:objectId [type loginExtHotDeskUserRequest]	
ns2:objectId [type loginHotDeskUserRequest]	
ns2:objectId [type monitorResult]	
ns1:objectId [type objectIdResult]	
ns2:objectId [type setCFRequest]	
ns2:objIdOfDeviceToMsg [element sendCallMeBackMsgNoCallRequest]	
ns2:objIdOfDeviceWithMsg [element clearCallMeBackMsgRequest]	
ns2:presenceValue [element setPRGPresenceRequest]	
ns1:result [type callStatusResult]	
ns2:result [type deviceConfigResult]	
ns1:result [type deviceFeaturesResult]	
ns2:result [type hotDeskUserLoginDeviceResult]	
ns2:result [type hotDeskUserLoginResult]	
ns1:result [type icpIdResult]	
ns2:result [type monitorResult]	

Name	Description
ns1:result [type objectIdResult]	
ns1:result [type result]	
ns1:result [type standardEventMsg]	
ns2:session [element redirectCallRequest]	
ns2:sessionId [element clearCallMeBackMsgRequest]	
ns2:sessionId [element makeCallRequest]	
ns2:sessionId [element outPulseDigitsRequest]	
ns2:sessionId [element pickupCallRequest]	
ns2:sessionId [element registerEventHandler]	
ns2:sessionId [element sendCallMeBackMsgNoCallRequest]	
ns2:sessionId [element setAccountCodeRequest]	
ns2:sessionId [element setPRGPresenceRequest]	
ns2:sessionId [type getDeviceConfigurationRequest]	
ns2:sessionId [type getIcpIdRequest]	
ns2:sessionId [type getLineAppearanceIdRequest]	
ns2:sessionId [type getPhoneNumberIdRequest]	
ns2:sessionId [type loginExtHotDeskUserRequest]	
ns2:sessionId [type loginHotDeskUserRequest]	
ns2:sessionId [type setCFRequest]	
ns1:time [type featureEventMsg]	
ns1:time [type systemEventMsg]	

Attribute: ns1:autoAnswer [type deviceFeaturesResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:deviceFeaturesResult

Attribute: ns2:buttonNum [type getLineAppearanceIdRequest]

Derived By

Type xs:int

Referenced By

- Complex Type ns2:getLineAppearanceIdRequest

Attribute: ns2:buttonNumber [type lineConfig]

Derived By

Type xs:int

Referenced By

- Complex Type ns2:lineConfig

Attribute: ns1:callEventTime [type callEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:callEventMsg

Attribute: ns1:callId [type partyMember]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:partyMember

Attribute: ns2:cfOn [type setCFRequest]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:setCFRequest

Attribute: ns1:DND [type deviceFeaturesResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:deviceFeaturesResult

Attribute: ns1:fwdOn [type forwardingInfo]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:forwardingInfo

Attribute: ns2:hotDeskDeviceObjectId [type hotDeskUserLoginResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:hotDeskUserLoginResult

Attribute: ns2:hotDeskUserObjectId [type hotDeskUserLoginDeviceResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:hotDeskUserLoginDeviceResult

Attribute: ns2:icpId [type getDeviceConfigurationRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getDeviceConfigurationRequest

Attribute: ns2:icpId [type getLineAppearanceIdRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getLineAppearanceIdRequest

Attribute: ns2:icpId [type getPhoneNumberIdRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getPhoneNumberIdRequest

Attribute: ns1:icpId [type icpIdResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:icpIdResult

Attribute: ns2:icpId [type monitorResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:monitorResult

Attribute: ns1:icpId [type systemEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:systemEventMsg

Attribute: ns2:isResilient [type deviceConfigResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:deviceConfigResult

Attribute: ns2:localCallId [element redirectCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:redirectCallRequest

Attribute: ns2:localCallId [element setAccountCodeRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:setAccountCodeRequest

Attribute: ns1:localCallId [type callEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:callEventMsg

Attribute: ns1:msgWaitingLamp [type deviceFeaturesResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:deviceFeaturesResult

Attribute: ns1:namePrivacy [type partyMember]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:partyMember

Attribute: ns1:numberPrivacy [type partyMember]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:partyMember

Attribute: ns2:objectId [element makeCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:makeCallRequest

Attribute: ns2:objectId [element outPulseDigitsRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:outPulseDigitsRequest

Attribute: ns2:objectId [element pickupCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:pickupCallRequest

Attribute: ns2:objectId [element redirectCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:redirectCallRequest

Attribute: ns2:objectId [element setAccountCodeRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:setAccountCodeRequest

Attribute: ns2:objectId [element setPRGPresenceRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:setPRGPresenceRequest

Attribute: ns1:objectId [type callEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:callEventMsg

Attribute: ns1:objectId [type featureEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:featureEventMsg

Attribute: ns2:objectId [type lineConfig]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:lineConfig

Attribute: ns2:objectId [type loginExtHotDeskUserRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:loginExtHotDeskUserRequest

Attribute: ns2:objectId [type loginHotDeskUserRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:loginHotDeskUserRequest

Attribute: ns2:objectId [type monitorResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:monitorResult

Attribute: ns1:objectId [type objectIdResult]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:objectIdResult

Attribute: ns2:objectId [type setCFRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:setCFRequest

Attribute: ns2:objIdOfDeviceToMsg [element sendCallMeBackMsgNoCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:sendCallMeBackMsgNoCallRequest

Attribute: ns2:objIdOfDeviceWithMsg [element clearCallMeBackMsgRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:clearCallMeBackMsgRequest

Attribute: ns2:presenceValue [element setPRGPresenceRequest]

Derived By

Type xs:boolean

Referenced By

- Element ns2:setPRGPresenceRequest

Attribute: ns1:result [type callStatusResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:callStatusResult

Attribute: ns2:result [type deviceConfigResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:deviceConfigResult

Attribute: ns1:result [type deviceFeaturesResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:deviceFeaturesResult

Attribute: ns2:result [type hotDeskUserLoginDeviceResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:hotDeskUserLoginDeviceResult

Attribute: ns2:result [type hotDeskUserLoginResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:hotDeskUserLoginResult

Attribute: ns1:result [type icpIdResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:icpIdResult

Attribute: ns2:result [type monitorResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns2:monitorResult

Attribute: ns1:result [type objectIdResult]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:objectIdResult

Attribute: ns1:result [type result]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:result

Attribute: ns1:result [type standardEventMsg]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:standardEventMsg

Attribute: ns2:session [element redirectCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:redirectCallRequest

Attribute: ns2:sessionId [element clearCallMeBackMsgRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:clearCallMeBackMsgRequest

Attribute: ns2:sessionId [element makeCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:makeCallRequest

Attribute: ns2:sessionId [element outPulseDigitsRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:outPulseDigitsRequest

Attribute: ns2:sessionId [element pickupCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:pickupCallRequest

Attribute: ns2:sessionId [element registerEventHandler]

Derived By

Type xs:long

Referenced By

- Element ns2:registerEventHandler

Attribute: ns2:sessionId [element sendCallMeBackMsgNoCallRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:sendCallMeBackMsgNoCallRequest

Attribute: ns2:sessionId [element setAccountCodeRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:setAccountCodeRequest

Attribute: ns2:sessionId [element setPRGPresenceRequest]

Derived By

Type xs:long

Referenced By

- Element ns2:setPRGPresenceRequest

Attribute: ns2:sessionId [type getDeviceConfigurationRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getDeviceConfigurationRequest

Attribute: ns2:sessionId [type getIcpIdRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getIcpIdRequest

Attribute: ns2:sessionId [type getLineAppearanceIdRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getLineAppearanceIdRequest

Attribute: ns2:sessionId [type getPhoneNumberIdRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:getPhoneNumberIdRequest

Attribute: ns2:sessionId [type loginExtHotDeskUserRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:loginExtHotDeskUserRequest

Attribute: ns2:sessionId [type loginHotDeskUserRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:loginHotDeskUserRequest

Attribute: ns2:sessionId [type setCFRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns2:setCFRequest

Attribute: ns1:time [type featureEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:featureEventMsg

Attribute: ns1:time [type systemEventMsg]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:systemEventMsg

SessionService Web Service

Description

This document contains Web Service descriptions for the following services.

Web Services

Name	Description
SessionService	

SessionService Web Service

Type

SOAP

Style

Document

See Also

- Methods
- Complex Types
- Elements
- Attributes
- Methods: SessionService

Methods

Name	Description
authenticate	
login	
logout	
resetSessionTimer	
serviceVersions	

Method: authenticate

Action

urn:authenticate

Style

Document

Input (Literal)

The input of this method is the document element tns:authenticate having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:req	ns1:authenticateSessionRequest	0..1	

Output (Literal)

The output of this method is the document element tns:authenticateResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:authenticateSessionResponse	0..1	

Method: login

Action

urn:login

Style

Document

Input (Literal)

The input of this method is the document element tns:login having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:req	ns1:loginRequest	0..1	

Output (Literal)

The output of this method is the document element tns:loginResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:loginResponse	0..1	

Method: logout

Action

urn:logout

Style

Document

Input (Literal)

The input of this method is the document element tns:logout having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Output (Literal)

The output of this method is the document element tns:logoutResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:boolean	1..1	

Method: resetSessionTimer

Action

urn:resetSessionTimer

Style

Document

Input (Literal)

The input of this method is the document element tns:resetSessionTimer having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Output (Literal)

The output of this method is the document element tns:resetSessionTimerResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:boolean	1..1	

Method: serviceVersions

Action

urn:serviceVersions

Style

Document

Input (Literal)

The input of this method is the document element tns:serviceVersions having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Output (Literal)

The output of this method is the document element tns:serviceVersionsResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:versionResponse	0..1	

Complex Types: SessionService

Complex Types

Name	Description
ns1:authenticateSessionRequest	
ns1:authenticateSessionResponse	
ns1:loginRequest	
ns1:loginResponse	
ns1:logoutResponse	
ns1:resetSessionTimerResponse	
ns1:versionResponse	

Complex Type: ns1:authenticateSessionRequest

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:signedAuthenticationData	xs:string	1..1	

Referenced By

- Element ns1:authenticateSessionRequest
- Element tns:req [element authenticate]

Complex Type: ns1:authenticateSessionResponse

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:sessionId	xs:long	Yes		
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:return [element authenticateResponse]

Complex Type: ns1:loginRequest

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:applicationName	xs:string	1..1	
ns1:companyName	xs:string	1..1	
ns1:localPassword	xs:string	1..1	
ns1:applicationPassword	xs:string	0..1	
ns1:certificate	xs:string	0..1	

Referenced By

- Element ns1:loginRequest
- Element tns:req [element login]

Complex Type: ns1:loginResponse

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	
ns1:authenticationData	xs:string	0..1	

Referenced By

Element tns:return [element loginResponse]

Complex Type: ns1:logoutResponse

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	

Complex Type: ns1:resetSessionTimerResponse

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	

Complex Type: ns1:versionResponse

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	
ns1:serviceVersions	xs:string	0..1	

Referenced By

- Element tns:return [element serviceVersionsResponse]

Elements: SessionService

Elements

Name	Description
ns1:applicationName [type loginRequest]	
ns1:applicationPassword [type loginRequest]	
tns:authenticate	
tns:authenticateResponse	
ns1:authenticateSessionRequest	
ns1:authenticationData [type loginResponse]	
ns1:certificate [type loginRequest]	
ns1:companyName [type loginRequest]	
ns1:errorDesc [type loginResponse]	
ns1:errorDesc [type logoutResponse]	
ns1:errorDesc [type resetSessionTimerResponse]	
ns1:errorDesc [type versionResponse]	
ns1:errorDescription [type authenticateSessionResponse]	
ns1:localPassword [type loginRequest]	
tns:login	
ns1:loginRequest	
tns:loginResponse	
tns:logout	
tns:logoutResponse	
tns:req [element authenticate]	
tns:req [element login]	
tns:resetSessionTimer	
tns:resetSessionTimerResponse	
tns:return [element authenticateResponse]	
tns:return [element loginResponse]	
tns:return [element logoutResponse]	
tns:return [element resetSessionTimerResponse]	

Name	Description
tns:return [element serviceVersionsResponse]	
ns1:serviceVersions [type versionResponse]	
tns:serviceVersions	
tns:serviceVersionsResponse	
tns:sessionId [element logout]	
tns:sessionId [element resetSessionTimer]	
tns:sessionId [element serviceVersions]	
ns1:signedAuthenticationData [type authenticateSessionRequest]	

Element: ns1:applicationName [type loginRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginRequest

Element: ns1:applicationPassword [type loginRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginRequest

Element: tns:authenticate

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:req	ns1:authenticateSessionRequest	0..1	

Element: tns:authenticateResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:authenticateSessionResponse	0..1	

Element: ns1:authenticateSessionRequest

Derived By

Type ns1:authenticateSessionRequest

Attributes

Name	Type	Required?	Default	Description
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:signedAuthenticationData	xs:string	1..1	

Element: ns1:authenticationData [type loginResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginResponse

Element: ns1:certificate [type loginRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginRequest

Element: ns1:companyName [type loginRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginRequest

Element: ns1:errorDesc [type loginResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginResponse

Element: ns1:errorDesc [type logoutResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:logoutResponse

Element: ns1:errorDesc [type resetSessionTimerResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:resetSessionTimerResponse

Element: ns1:errorDesc [type versionResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:versionResponse

Element: ns1:errorDescription [type authenticateSessionResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:authenticateSessionResponse

Element: ns1:localPassword [type loginRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:loginRequest

Element: tns:login

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:req	ns1:loginRequest	0..1	

Element: ns1:loginRequest

Derived By

Type ns1:loginRequest

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:applicationName	xs:string	1..1	
ns1:companyName	xs:string	1..1	
ns1:localPassword	xs:string	1..1	
ns1:applicationPassword	xs:string	0..1	
ns1:certificate	xs:string	0..1	

Element: tns:loginResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:loginResponse	0..1	

Element: tns:logout

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Element: tns:logoutResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:boolean	1..1	

Element: tns:req [element authenticate]

Derived By

Type ns1:authenticateSessionRequest

Attributes

Name	Type	Required?	Default	Description
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:signedAuthenticationData	xs:string	1..1	

Referenced By

- Element tns:authenticate

Element: tns:req [element login]

Derived By

Type ns1:loginRequest

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:applicationName	xs:string	1..1	
ns1:companyName	xs:string	1..1	
ns1:localPassword	xs:string	1..1	
ns1:applicationPassword	xs:string	0..1	
ns1:certificate	xs:string	0..1	

Referenced By

- Element tns:login

Element: tns:resetSessionTimer

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Element: tns:resetSessionTimerResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:Boolean	1..1	

Element: tns:return [element authenticateResponse]

Derived By

Type ns1:authenticateSessionResponse

Attributes

Name	Type	Required?	Default	Description
ns1:sessionId	xs:long	Yes		
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	

Referenced By

- Element tns:authenticateResponse

Element: tns:return [element loginResponse]

Derived By

Type ns1:loginResponse

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	
ns1:authenticationData	xs:string	0..1	

Referenced By

- Element tns:loginResponse

Element: tns:return [element logoutResponse]

Derived By

Type xs:boolean

Referenced By

- Element tns:logoutResponse

Element: tns:return [element resetSessionTimerResponse]

Derived By

Type xs:boolean

Referenced By

- Element tns:resetSessionTimerResponse

Element: tns:return [element serviceVersionsResponse]

Derived By

Type ns1:versionResponse

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		
ns1:sessionId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDesc	xs:string	0..1	
ns1:serviceVersions	xs:string	0..1	

Referenced By

- Element tns:serviceVersionsResponse

Element: ns1:serviceVersions [type versionResponse]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:versionResponse

Element: tns:serviceVersions

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:sessionId	xs:long	1..1	

Element: tns:serviceVersionsResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	ns1:versionResponse	0..1	

Element: tns:sessionId [element logout]

Derived By

Type xs:long

Referenced By

- Element tns:logout

Element: tns:sessionId [element resetSessionTimer]

Derived By

Type xs:long

Referenced By

- Element tns:resetSessionTimer

Element: tns:sessionId [element serviceVersions]

Derived By

Type xs:long

Referenced By

- Element tns:serviceVersions

Element: ns1:signedAuthenticationData [type authenticateSessionRequest]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:authenticateSessionRequest

Attributes: SessionService

Attributes

Name	Description
ns1:result [type authenticateSessionResponse]	
ns1:result [type loginResponse]	
ns1:result [type logoutResponse]	
ns1:result [type resetSessionTimerResponse]	
ns1:result [type versionResponse]	
ns1:sessionId [type authenticateSessionRequest]	
ns1:sessionId [type authenticateSessionResponse]	
ns1:sessionId [type loginResponse]	
ns1:sessionId [type versionResponse]	

Attribute: ns1:result [type authenticateSessionResponse]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:authenticateSessionResponse

Attribute: ns1:result [type loginResponse]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:loginResponse

Attribute: ns1:result [type logoutResponse]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:logoutResponse

Attribute: ns1:result [type resetSessionTimerResponse]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:resetSessionTimerResponse

Attribute: ns1:result [type versionResponse]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:versionResponse

Attribute: ns1:sessionId [type authenticateSessionRequest]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:authenticateSessionRequest

Attribute: ns1:sessionId [type authenticateSessionResponse]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:authenticateSessionResponse

Attribute: ns1:sessionId [type loginResponse]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:loginResponse

Attribute: ns1:sessionId [type versionResponse]

Derived By

Type xs:long

Referenced By

- Complex Type ns1:versionResponse

EventHandlerCCService Web Service

Description

This document contains Web Service descriptions for the following services.

Web Services

Name	Description
EventHandlerCCService	

EventHandlerCCService Web Service

Type

SOAP

Style

Document

See Also

- Methods
- Complex Types
- Simple Types
- Elements
- Attributes

Methods: EventHandlerCCService

Methods

Name	Description
handleEvent	

Method: handleEvent

Action

urn:handleEvent

Style

Document

Input (Literal)

The input of this method is the document element tns:handleEvent having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:event	ns1:standardEventMsg	0..1	

Output (Literal)

The output of this method is the document element tns:handleEventResponse having the structure defined by the following table.

Element	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:boolean	1..1	

Complex Types: EventHandlerCCService

Complex Types

Name	Description
ns1:standardEventMsg	

Complex Type: ns1:standardEventMsg

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:eventType	xs:string (restriction)	0..1	
ns1:callEvent	xs:anyType (restriction)	0..1	
ns1:featureEvent	xs:anyType (restriction)	0..1	
ns1:systemEvent	xs:anyType (restriction)	0..1	

Referenced By

- Element tns:event [element handleEvent]

Simple Types: EventHandlerCCService

Simple Types

Name	Description
ns1:attributeType	
ns1:callState	
ns1:featuresAllowed	

Simple Type: ns1:attributeType

Derived By

Restricting xs:string

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Element ns1:attributeType [type standardEventMsg/callEvent/callEventAttribute]
- Element ns1:attributeType [type standardEventMsg/callEvent/device/deviceAttribute]

Simple Type: ns1:callState

Derived By

Restricting xs:string

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	
HELD	
IDLE	
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Element ns1:callState [type standardEventMsg/callEvent]
- Element ns1:state [type standardEventMsg/featureEvent]

Simple Type: ns1:featuresAllowed

Derived By

Restricting xs:string

Enumeration

Value	Description
NOT_SUPPORTED	
ANSWER_CALL	
REDIRECT_CALL	
CAMPON_CALL	
SET_CALL_ME_BACK	
INTRUDE_CALL	
ALTERNATE_CALL	
TRADE_CALL	
CONFERENCE_CALL	
SPLIT_CONFERENCE_CALL	
CONSULTATION_CALL	
CANCEL_CONSULTATION_CALL	
TRANSFER_CALL	
SEND_CALLBACK_MESSAGE	
SET_ACCOUNT_CODE	
NEW_CALL	
CLEAR_CALL	
OUT_PULSE_DIGITS	
HOLD_CALL	
RETRIEVE_CALL	
LISTEN_CALL	
ASSIGN_CALLER_ID	
ALTER_WORK_TIMER	
TAP_CALL	

Referenced By

- Element ns1:featuresAllowed [type standardEventMsg/callEvent]

Elements: EventHandlerCCService

Elements

Name	Description
ns1:accountCode [type standardEventMsg/callEvent]	
ns1:accountCodeType [type standardEventMsg/callEvent]	
ns1:attributeName [type standardEventMsg/callEvent/callEventAttribute]	
ns1:attributeName [type standardEventMsg/callEvent/device/deviceAttribute]	
ns1:attributeType [type standardEventMsg/callEvent/callEventAttribute]	
ns1:attributeType [type standardEventMsg/callEvent/device/deviceAttribute]	
ns1:attributeValue [type standardEventMsg/callEvent/callEventAttribute]	
ns1:attributeValue [type standardEventMsg/callEvent/device/deviceAttribute]	
ns1:callEvent [type standardEventMsg]	
ns1:callEventAttribute [type standardEventMsg/callEvent]	
ns1:callState [type standardEventMsg/callEvent]	
ns1:cause [type standardEventMsg/callEvent]	
ns1:connectionState [type standardEventMsg/systemEvent]	
ns1:device [type standardEventMsg/callEvent]	
ns1:deviceAttribute [type standardEventMsg/callEvent/device]	
ns1:deviceName [type standardEventMsg/callEvent/device]	
ns1:deviceType [type standardEventMsg/callEvent/party]	
ns1:devNumber [type standardEventMsg/featureEvent]	
ns1:errorDescription [type standardEventMsg]	
tns:event [element handleEvent]	
ns1:eventType [type standardEventMsg/featureEvent]	
ns1:eventType [type standardEventMsg/systemEvent]	

Name	Description
ns1:eventType [type standardEventMsg]	
ns1:featureEvent [type standardEventMsg]	
ns1:featuresAllowed [type standardEventMsg/callEvent]	
ns1:forwardDn [type standardEventMsg/featureEvent]	
ns1:groupDn [type standardEventMsg/featureEvent]	
tns:handleEvent	
tns:handleEventResponse	
ns1:memberName [type standardEventMsg/callEvent/party]	
ns1:memberNetworkExt [type standardEventMsg/callEvent/party]	
ns1:memberNum [type standardEventMsg/callEvent/party]	
ns1:party [type standardEventMsg/callEvent]	
ns1:peerId [type standardEventMsg/callEvent/party]	
ns1:peerName [type standardEventMsg/callEvent/party]	
ns1:remoteMemberExt [type standardEventMsg/callEvent/party]	
ns1:remoteMemberName [type standardEventMsg/callEvent/party]	
tns:return [element handleEventResponse]	
ns1:state [type standardEventMsg/featureEvent]	
ns1:systemEvent [type standardEventMsg]	
ns1:type [type standardEventMsg/callEvent]	
ns1:userDn [type standardEventMsg/featureEvent]	

Element: ns1:accountCode [type standardEventMsg/callEvent]

Derived By

Type xs:string

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:accountCodeType [type standardEventMsg/callEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
VERIFIED_ACCOUNT_CODE	
UNVERIFIED_ACCOUNT_CODE	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:attributeName [type standardEventMsg/callEvent/callEventAttribute]

Derived By

Restricting xs:string

Enumeration

Value	Description
DIALED_DIGITS	
CURRENT_GLOBAL_CALL_ID	
PRIMARY_GLOBAL_CALLID	
SECONDARY_GLOBAL_CALLID	
DISTRIBUTION_RING_GRP_RINGALL	
DISTRIBUTION_RING_GRP_CASCADE	
DISTRIBUTION_HUNT_GRP_CIRCULAR	
DISTRIBUTION_HUNT_GRP_LINEAR	
DISTRIBUTION_PERSONAL_RING_GRP	

Referenced By

- Element ns1:callEventAttribute [type standardEventMsg/callEvent]

Element: ns1:attributeName [type standardEventMsg/callEvent/device/deviceAttribute]

Derived By

Restricting xs:string

Enumeration

Value	Description
NUMBER	
NAME	
DEVICE_TYPE	
DIALED_DIGITS	
ADV_MSG	
CALL_ID	
SUITE_PILOT_NUMBER	
LINE_APPEARANCE	
NUMBER_PRIVACY	
NAME_PRIVACY	
SUITE_PILOT_NAME	
PRG_NUMBER	
PRG_NAME	
TRUNK_OUTPULSED_DIGITS	
EHDU_NUMBER	
TRUNK_NUMBER	
TRUNK_NAME	
MFANI_INFO_DIGIT	
CALLING_LINE_CATEGORY	
PEER_NAME	
PEER_ID	

Referenced By

- Element ns1:deviceAttribute [type standardEventMsg/callEvent/device]

Element: ns1:attributeType [type standardEventMsg/callEvent/callEventAttribute]

Derived By

Type ns1:attributeType

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Element ns1:callEventAttribute [type standardEventMsg/callEvent]

Element: ns1:attributeType [type standardEventMsg/callEvent/device/deviceAttribute]

Derived By

Type ns1:attributeType

Enumeration

Value	Description
STRING	
INTEGER	
LONG_INTEGER	
BOOLEAN	

Referenced By

- Element ns1:deviceAttribute [type standardEventMsg/callEvent/device]

Element: ns1:attributeValue [type standardEventMsg/callEvent/callEventAttribute]

Derived By

Type xs:string

Referenced By

- Element ns1:callEventAttribute [type standardEventMsg/callEvent]

Element: ns1:attributeValue [type standardEventMsg/callEvent/device/deviceAttribute]

Derived By

Type xs:string

Referenced By

- Element ns1:deviceAttribute [type standardEventMsg/callEvent/device]

Element: ns1:callEvent [type standardEventMsg]

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:localCallId	xs:long	Yes		
ns1:callEventTime	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:type	xs:string (restriction)	0..1	
ns1:cause	xs:string (restriction)	0..1	
ns1:callState	ns1:callState	0..1	
ns1:callEventAttribute	xs:anyType (restriction)	0..*	
ns1:featuresAllowed	ns1:featuresAllowed	0..*	
ns1:device	xs:anyType (restriction)	0..*	
ns1:party	xs:anyType (restriction)	0..*	
ns1:accountCode	xs:string	0..1	
ns1:accountCodeType	xs:string (restriction)	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:callEventAttribute [type standardEventMsg/callEvent]

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	xs:string (restriction)	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:callState [type standardEventMsg/callEvent]

Derived By

Type ns1:callState

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	
HELD	
IDLE	
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:cause [type standardEventMsg/callEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
ACCOUNT_CODE_SET	
ACD_AGENT_TIMEOUT	
ACD_CALL_QUEUED	
ACD_CALL_DELIVERED	
ACD_CALL_ANSWERED	
ACD_CALL_CLEARED	
ACD_CALL_OVERFLOWED	
ACD_CALL_ABANDONED	
ACD_CALL_CLEAR_INVOKED	
ACD_CALL_QUEUED_FAILED	
ACD_CALL_QUEUED_OVERFLOWED	
ACD_CALL_REDIRECTED	
ACD_CALL_RECEIVED	
ACD_REMOTE_CALL_QUEUED	
ACD_REMOTE_CALL_DEQUEUED	
ACD_REMOTE_CALL_DISTRIBUTED	
ACD_HELP_CALL	
ACD_INTERFLOWED_DISCONNECTED	
ACD_INTERFLOWED_REROUTED	
ACD_RAD_STARTED	
ACD_REQUEST	
ACD_REQUEST_DISCONNECTED	
ACD_REQUEST_REROUTED	
ACD_REQUEUE_REQUEST	
ACD_REQUEUE_REQUEST_DISCONNECTED	
ACD_REQUEUE_REQUEST_REROUTED	
ACD_SILENT_MONITOR_ESTABLISHED	
ACD_SILENT_MONITOR_DROPPED	
ACD_SILENT_MONITOR	

Value	Description
ACD_SILENT_MONITOR_INVOKED	
ALTERNATE_CALL_INVOKED	
ANSWERED	
ANSWERED_INVOKED	
CALLBACK_INVOKED	
CALLBACK_MATURED	
CALL_QUEUED	
CALL_IS_WAITING	
CLEARED	
CLEAR_INVOKED	
CONFERENCE_CALL_HELD	
CONFERENCE_CALL_RETRIEVED	
CONFERENCE_CONS_HELD	
CONFERENCE_CONS_RETRIEVED	
CONFERENCED	
CONFERENCE_ESTABLISHED	
CONFERENCE_INVOKED	
CONFERENCE_MEMBER_ADDED	
CONFERENCE_MEMBER_DROPPED	
CONFERENCE_MONITOR_SET	
CONFERENCE_REMOTE_UPDATE	
CONFERENCE_SPLIT	
CONFERENCE_SPLIT_INVOKED	
CONFERENCE_TRANSFERRED	
CONF_RETRIEVE_INVOKED	
CONS_HOLD	
CONS_HOLD_INVOKED	
CONS_CALL_INVOKED	
CONS_HELD_PARTY_DROPPED	
CONTROLLER_COMMS_FAILED	
CONTROLLER_COMMS_RESTORED	
DELIVERED	
DESTINATION_BUSY	
DESTINATION_DO_NOT_DISTURB	
DESTINATION_NOT_OBTAINABLE	

Value	Description
DESTINATION_UNAVAILABLE	
DEVICE_IN_SERVICE	
DEVICE_OUT_OF_SERVICE	
DIVERTED_ALWAYS_FROM	
DIVERTED_ALWAYS_TO	
DIVERTED_NO_ANSWER_FROM	
DIVERTED_NO_ANSWER_TO	
DIVERTED_NO_ANSWER_AWAY	
DIVERTED_ON_BUSY_FROM	
DIVERTED_ON_BUSY_TO	
DIVERTED_PICKUP_AWAY	
ERROR_DETECTED	
EXTENSION_IN_USE	
HARD_HELD_PARTY_DROPPED	
HARD_HOLD	
HARD_HOLD_INVOKED	
HARD_HOLD_RETRIEVED	
INTRUSION	
INTRUSION_INVOKED	
INVALID_ACCOUNT_CODE	
MEMBER_ANSWERED	
MONITOR_SET	
NETWORK_BUSY	
NETWORK_CONGESTION	
NETWORK_NOT_OBTAINABLE	
NEW_CALL_INVOKED	
NEW_CALL	
NO_ANSWER	
NO_SUCH_NUMBER	
ONS_CALL_ORIGINATED	
OTHER_DEVICE_CLEARED	
OTHER_DEVICE_DROPPED	
PICKUP	
PICKUP_INVOKED	
PRIVILEGE_VIOLATION	

Value	Description
QUEUED_PARTY_DROPPED	
RAD_UNAVAILABLE	
RECALL_HELD_PARTY	
REDIRECTED_AWAY	
REDIRECTED_FROM	
REDIRECTED_HANDOFF	
REDIRECTED_ON_ERROR	
REDIRECTED_TO	
REMOTE_PARTY_UPDATED	
GLOBAL_CALLID_UPDATED	
CALL_DISTRIBUTED	
CALL_ANSWERED	
CALL_DISCONNECTED	
CALL_OVERFLOWED	
CALL_WAITING_PARTY_DISCONNECTED	
CFNA_CALL_QUEUED	
CFA_CALL_QUEUED	
CFB_CALL_QUEUED	
CFA_DESTINATION_BUSY	
CFB_DESTINATION_BUSY	
RETRIEVE	
RETRIEVE_INVOKED	
ROUTING_DEVICE_CALL_QUEUED	
ROUTING_DEVICE_CALL_OVERFLOW	
ROUTING_DEVICE_CALL_REROUTED	
ROUTING_DEVICE_CALL_ABONDONED	
ROUTING_DEVICE_CFALWAYS_CALL_QUEUED	
ROUTING_DEVICE_CFBUSY_CALL_QUEUED	
ROUTING_DEVICE_CFNOANS_CALL_QUEUED	
ROUTING_DEVICE_RAD_STARTED	
ROUTING_DEVICE_REROUTED_CALL_QUEUED	
ROUTING_DEVICE_TRANS_CALL_QUEUED	
SECONDARY_CONTROLLER_COMMS_FAILED	
SECONDARY_MONITOR_SET	
SUPERVISED_TRANSFER	

Value	Description
TAP_CALL	
THIS_DEVICE_DROPPED	
TRANSFER_INVOKED	
TRANSFER_RECALL_NO_ANSWER	
TRANSFER_RECALL_ON_BUSY	
TRANSFER_TO_BUSY	
TRANSFERRED_CALL	
TRUNKS_BUSY	
TRUNK_DIGITS_OUTPULSED	
UNSUPERVISED_TRANSFER	
WAITING_CALL_RINGING	
WORK_TIMER_EXPIRED	
HANDOFF_PUSH	
PEER_OUT_OF_SERVICE	
PEER_INSERVICE	
EHDU_ACCESS	
EHDU_CALL_BACK	
CLOSING_CALL_SERVER	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:connectionState [type standardEventMsg/systemEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
NOT_CONNECTED	
CONNECTED	
CONNECTION_FAILURE	
CONNECTION_ATTEMPT	
CONNECTION_INITIALIZING	

Referenced By

- Element ns1:systemEvent [type standardEventMsg]

Element: ns1:device [type standardEventMsg/callEvent]

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:deviceName	xs:string (restriction)	0..1	
ns1:deviceAttribute	xs:anyType (restriction)	0..*	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:deviceAttribute [type standardEventMsg/callEvent/device]

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:attributeName	xs:string (restriction)	0..1	
ns1:attributeValue	xs:string	0..1	
ns1:attributeType	ns1:attributeType	0..1	

Referenced By

- Element ns1:device [type standardEventMsg/callEvent]

Element: ns1:deviceName [type standardEventMsg/callEvent/device]

Derived By

Restricting xs:string

Enumeration

Value	Description
CALLING_DEVICE	
CALLING_DEVICE_NETWORK_EXT	
ORIGINAL_DESTINATION_DEVICE	
GROUP_DEVICE	
CALLED_DEVICE	
CONTROLLER_DEVICE	
CONNECTED_DEVICE	
PICKUP_DEVICE	
SPLITTING_DEVICE	
TRANSFER_CONTROLLER_DEVICE	
TRANSFERRED_DEVICE	
NEW_DESTINATION_DEVICE	
DROPPED_DEVICE	
USING_DEVICE	
HOLDING_DEVICE	
WAITING_DEVICE	

Referenced By

- Element ns1:device [type standardEventMsg/callEvent]

Element: ns1:deviceType [type standardEventMsg/callEvent/party]

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
SIP_PRIVATE_TRUNK	
SIP_PUBLIC_TRUNK	
MULTICALL_GROUP	
HUNT_GROUP	
KEYLINE	
VOICE_SET_PRIME	
ACD_STRUCTURE	
LINE_APPEARANCE	
DATA_EXTENSION	
ACD2_PATH	
ACD2_AGENT	
ACD2_GROUP	
ROUTING_DEVICE	
ACD2_REMOTE_SUBGROUP	
RING_ALL_GROUP	
RING_CASCADE_GROUP	
ACDX_AGENT	
ACDX_GROUP	
ACDX_PATH	
PERSONAL_RING_GROUP	
PEER	
EXTERNAL_DEVICE_TYPE	
INTERNAL_DEVICE_TYPE	

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:devNumber [type standardEventMsg/featureEvent]

Derived By

Type xs:string

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Element: ns1:errorDescription [type standardEventMsg]

Derived By

Type xs:string

Referenced By

- Complex Type ns1:standardEventMsg

Element: tns:event [element handleEvent]

Derived By

Type ns1:standardEventMsg

Attributes

Name	Type	Required?	Default	Description
ns1:result	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:errorDescription	xs:string	0..1	
ns1:eventType	xs:string (restriction)	0..1	
ns1:callEvent	xs:anyType (restriction)	0..1	
ns1:featureEvent	xs:anyType (restriction)	0..1	
ns1:systemEvent	xs:anyType (restriction)	0..1	

Referenced By

- Element tns:handleEvent

Element: ns1:eventType [type standardEventMsg/featureEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
AUTO_ANSWER_FEATURE	
DO_NOT_DISTURB	
MSG_WAITING_INDICATOR	
INSERVICE	
DO_NOT_DISTURB_ALL	
MSG_WAITING_IND_ALL	
MAKE_BUSY	
JOIN_GROUP	
LEAVE_GROUP	
JOIN_ALL_ACD_GROUPS	
MAKE_BUSY_STATE	
LEAVE_All_ACD_GROUPS	
ACD_AGENT	
ACD_AGENT_BLOCKED	
ACD2_LOGIN_NO_LICENSE	
HOT_DESK_USER	
CF_ALWAYS	
CF_BUSY_EXTERNAL	
CF_BUSY_INTERNAL	
CF_NO_ANSWER_INTERNAL	
CF_NO_ANSWER_EXTERNAL	
CF_NO_ANSWER_BUSY	
CF_ALL_MODES	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CONFERENCE_FEATURE	
FORWARD_FEATURE	
USER_ID_FEATURE_EVENT	

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Element: ns1:eventType [type standardEventMsg/systemEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
ICP_COMMUNICATION_EVENT	
OIG_GW_SHUTDOWN	

Referenced By

- Element ns1:systemEvent [type standardEventMsg]

Element: ns1:eventType [type standardEventMsg]

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
CALL_EVENT	
FEATURE_EVENT	
ACD_EVENT	
CONFERENCE_EVENT	
SYSTEM_EVENT	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:featureEvent [type standardEventMsg]

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:time	xs:long	Yes		
ns1:objectId	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:state	ns1:callState	0..1	
ns1:devNumber	xs:string	0..1	
ns1:eventType	xs:string (restriction)	0..1	
ns1:userDn	xs:string	0..1	
ns1:groupDn	xs:string	0..1	
ns1:forwardDn	xs:string	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:featuresAllowed [type standardEventMsg/callEvent]

Derived By

Type ns1:featuresAllowed

Enumeration

Value	Description
NOT_SUPPORTED	
ANSWER_CALL	
REDIRECT_CALL	
CAMPON_CALL	
SET_CALL_ME_BACK	
INTRUDE_CALL	
ALTERNATE_CALL	
TRADE_CALL	
CONFERENCE_CALL	
SPLIT_CONFERENCE_CALL	
CONSULTATION_CALL	
CANCEL_CONSULTATION_CALL	
TRANSFER_CALL	
SEND_CALLBACK_MESSAGE	
SET_ACCOUNT_CODE	
NEW_CALL	
CLEAR_CALL	
OUT_PULSE_DIGITS	
HOLD_CALL	
RETRIEVE_CALL	
LISTEN_CALL	
ASSIGN_CALLER_ID	
ALTER_WORK_TIMER	
TAP_CALL	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:forwardDn [type standardEventMsg/featureEvent]

Derived By

Type xs:string

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Element: ns1:groupDn [type standardEventMsg/featureEvent]

Derived By

Type xs:string

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Element: tns:handleEvent

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:event	ns1:standardEventMsg	0..1	

Element: tns:handleEventResponse

Derived By

Restricting xs:anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
tns:return	xs:boolean	1..1	

Element: ns1:memberName [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:memberNetworkExt [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:memberNum [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:party [type standardEventMsg/callEvent]

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:callId	xs:long	Yes		
ns1:numberPrivacy	xs:boolean	Yes		
ns1:namePrivacy	xs:boolean	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:memberNum	xs:string	0..1	
ns1:memberName	xs:string	0..1	
ns1:deviceType	xs:string (restriction)	0..1	
ns1:remoteMemberExt	xs:string	0..1	
ns1:remoteMemberName	xs:string	0..1	
ns1:memberNetworkExt	xs:string	0..1	
ns1:peerId	xs:string	0..1	
ns1:peerName	xs:string	0..1	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:peerId [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:peerName [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:remoteMemberExt [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: ns1:remoteMemberName [type standardEventMsg/callEvent/party]

Derived By

Type xs:string

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Element: tns:return [element handleEventResponse]

Derived By

Type xs:boolean

Referenced By

- Element tns:handleEventResponse

Element: ns1:state [type standardEventMsg/featureEvent]

Derived By

Type ns1:callState

Enumeration

Value	Description
NIL	
DELIVERED	
ESTABLISHED	
EXTENSION_IN_USE	
FAILED	
FEATURE_DISABLED	
FEATURE_ENABLED	
HELD	
IDLE	
ORIGINATED	
OUT_OF_SERVICE	
QUEUED	
RECEIVED	
UNAVAILABLE	

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Element: ns1:systemEvent [type standardEventMsg]

Derived By

Restricting xs:anyType

Attributes

Name	Type	Required?	Default	Description
ns1:icpId	xs:long	Yes		
ns1:time	xs:long	Yes		

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
SEQUENCE		1..1	
ns1:eventType	xs:string (restriction)	0..1	
ns1:connectionState	xs:string (restriction)	0..1	

Referenced By

- Complex Type ns1:standardEventMsg

Element: ns1:type [type standardEventMsg/callEvent]

Derived By

Restricting xs:string

Enumeration

Value	Description
UNKNOWN	
ACCOUNT_CODE_SET	
ACD2_PATH	
ACD2_GROUP	
GROUP	
ACD_AGENT_FEATURE	
ACTIVATE_FEATURE	
CALL_CLEAR	
CALL_CONFERENCED	
CALL_DELIVERED	
CALL_DIVERTED	
CALL_ESTABLISHED	
CALL_FAILED	
CALL_HELD	
CALL_ORIGINATED	
CALL_QUEUED	
CALL_RECEIVED	
CALL_RETRIEVED	
CALL_TRANSFERRED	
CMD_RESPONSE	
CONFERENCE_HELD	
CONFERENCE_FEATURE	
DEVICE_DROPPED	
EXTENSION_IN_USE	
FORWARD_FEATURE	
IN_SERVICE	
MONITOR_SET	
OUT_OF_SERVICE	
REMOTE_PARTY_UPDATE	
RESILIENT_DEVICE	

Value	Description
ROUTING_DEVICE	
TRUNK_DIGITS_OUTPULSED	
WORK_TIMER_EXPIRED	
USERID_FEATURE	
GROUP_PRESENCE_FEATURE	
ACD_EXPRESS_GROUP_EVENT	
MONITOR_FAILED_EVENT	
PRG_MONITOR_SET_EVENT	
PRG_MONITOR_FAILED_EVENT	

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Element: ns1:userDn [type standardEventMsg/featureEvent]

Derived By

Type xs:string

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Attributes: EventHandlerCCService

Attributes

Name	Description
ns1:callEventTime [type standardEventMsg/callEvent]	
ns1:callId [type standardEventMsg/callEvent/party]	
ns1:icpId [type standardEventMsg/systemEvent]	
ns1:localCallId [type standardEventMsg/callEvent]	
ns1:namePrivacy [type standardEventMsg/callEvent/party]	
ns1:numberPrivacy [type standardEventMsg/callEvent/party]	
ns1:objectId [type standardEventMsg/callEvent]	
ns1:objectId [type standardEventMsg/featureEvent]	
ns1:result [type standardEventMsg]	
ns1:time [type standardEventMsg/featureEvent]	
ns1:time [type standardEventMsg/systemEvent]	

Attribute: ns1:callEventTime [type standardEventMsg/callEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Attribute: ns1:callId [type standardEventMsg/callEvent/party]

Derived By

Type xs:long

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Attribute: ns1:icpId [type standardEventMsg/systemEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:systemEvent [type standardEventMsg]

Attribute: ns1:localCallId [type standardEventMsg/callEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Attribute: ns1:namePrivacy [type standardEventMsg/callEvent/party]

Derived By

Type xs:boolean

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Attribute: ns1:numberPrivacy [type standardEventMsg/callEvent/party]

Derived By

Type xs:boolean

Referenced By

- Element ns1:party [type standardEventMsg/callEvent]

Attribute: ns1:objectId [type standardEventMsg/callEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:callEvent [type standardEventMsg]

Attribute: ns1:objectId [type standardEventMsg/featureEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Attribute: ns1:result [type standardEventMsg]

Derived By

Type xs:boolean

Referenced By

- Complex Type ns1:standardEventMsg

Attribute: ns1:time [type standardEventMsg/featureEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:featureEvent [type standardEventMsg]

Attribute: ns1:time [type standardEventMsg/systemEvent]

Derived By

Type xs:long

Referenced By

- Element ns1:systemEvent [type standardEventMsg]

Glossary

ACD	Automatic Call Distribution
ACL	Access Control List
AMC	Applications Management Center (licensing server)
API	Application Programming Interface
CCS	Call Control Service
COS	Class of Service
DLL	Dynamic Link Library
DMZ	De-Militarized Zone
DNS	Domain Name Server
ICP	IP Communications Platform
IP	Internet Protocol
IVR	Interactive Voice Response
LAN	Local Area Network
MCD	Mitel Communications Director
MCS	Mitel Certificate Server
MiTAI	Mitel Telephony Application Interface
MOL	Mitel OnLine
MSA	Mitel Solutions Alliance (Mitel developer partner program)
MSL	Mitel Standard Linux (operating system)
MSP	Media Service Provider
OIG	Open Integration Gateway
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
TDM	Time Division Multiplexing
VOIP	Voice over IP
vLAN	Virtual Local Area Network
WAN	Wide Area Network
WSDL	Web Service Description Language

GLOBAL HEADQUARTERS	U.S.	EMEA	CALA	ASIA PACIFIC
Tel: +1(613) 592-2122 Fax: +1(613) 592-4784	Tel: +1(480) 961-9000 Fax: +1(480) 961-1370	Tel: +44(0)1291-430000 Fax: +44(0)1291-430400	Tel: +1(613) 592-2122 Fax: +1(613) 592-7825	Tel: +61(0) 2 9023 9500 Fax: +61(0) 2 9023 9501

FOR MORE INFORMATION ON OUR WORLDWIDE OFFICE LOCATIONS, VISIT OUR WEBSITE AT **MITEL.COM/OFFICES**

mitel.com