

Mitel Open Integration Gateway

DEVELOPER GUIDE – CALL CONTROL SERVICES

Release 3.0

November 2015



NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (Mitel®). The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

TRADEMARKS

MiVoice Business and MSL are trademarks of Mitel Networks Corporation.

Windows, XP, and Vista are trademarks of Microsoft Corporation.

Salesforce is a trademark of Salesforce Corporation.

Google is a registered trademark of Google Inc.

Other product names mentioned in this document may be trademarks of their respective companies and are hereby acknowledged.

Mitel Open Integration Gateway Developer Guide - Call Control Services Release 3.0 November 2015

®,™ Trademark of Mitel Networks Corporation
© Copyright 2014-2015, Mitel Networks Corporation
All rights reserved

INTRODUCTION	1
Mitel OIG web services	2
Mitel OIG documentation	3
Mitel Solutions Alliance (MSA) portal	4
WSDL Files for Mitel OIG Operations and Events	4
Messaging Formats (SOAP and REST)	4
MITEL OIG CALL CONTROL SERVICE OVERVIEW	5
Standard and Advanced Call Control Services	5
Standard Call Control Service	5
Advanced Call Control Service	5
Call Control Service Key Concepts	5
Overview of Mitel OIG Call Control Service monitoring feature	7
Mitel OIG Call Control Service Components	7
Mitel OIG Call Control Service Device Support	8
MiVoice Business Configuration	8
Get configuration and Get Status Operations now have MiVoice Business information	9
Recent changes affecting the Mitel OIG Call Control Application	9
MITEL OIG SUPPORT FOR MIVOICE BUSINESS IP PHONE RESILIENCY	10
How Mitel OIG supports MiVoice Business IP Phone Resiliency	11
Resilient Scenarios	13
Scenario 1: Application #1 starts a monitor on phone prime line 2000	13
Scenario 2: Monitor on phone prime line 2000 fails over to secondary MiVoice Business	14
Scenario 3: Application starts up when primary MiVoice Business is down and phone 2000 is registered with the secondary MiVoice Business	16
Mitel OIG IP Device Resiliency Impact on Application	18
Applications should be aware of the primary and secondary MiVoice Business	18
Applications that track Device State	18
How to Determine If a Device is Resilient	19
Call Event and Feature Event Changes	19
Starting Monitor on Resilient Device When MiVoice Business is Down	20
MiVoice Business and Mitel OIG Connection Failures	20
MiVoice Business Connection Recovery	20
Event RESILIENT_DEVICE with Cause REDIRECTED_HANDOFF	21
Event RESILIENT_DEVICE with Cause RESILIENT_INFO_UPDATE	21
Mitel OIG Monitor Recovery	21
Mitel OIG Non-Resilient Monitors	22
MITEL OIG SAMPLE SCENARIOS	24
Create Phone Monitor	24

Basic Call Monitoring	25
CALL CONTROL SERVICE OPERATIONS.....	26
Standard Service Operations Overview	26
Advanced Service Operations Overview	30
Call Control Service Operations – Standard Type.....	36
alternateCall	36
answerCall	37
camponCall	37
cancelConsCall	38
clearCall	39
clearCallMeBackMsg	40
conferenceCall	40
consultationCall.....	41
getIcpConfig	42
getCallStatus.....	43
getDeviceConfigurationEx()	43
getDeviceFeatures()	45
getDeviceMembers	46
getEvent	47
getHotDeskUserLoginDevice()	47
getHotDeskUserDn().....	48
getIcpId	49
getLineAppearancId.....	50
getPhoneNumberId	50
holdCall	51
loginExtHotDeskUser()	52
loginHotDeskUser().....	53
logoutExtHotDeskUser()	53
logoutHotDeskUser()	54
makeCall	54
monitorFeatures.....	56
monitorObject.....	57
monitorPRGPresence	58
newCall	58
outPulseDigits	59
pickupCall	60
redirectCall	60
registerEventHandler	62
remoteRetrieveCall	62
retrieveCall	63
sendCallMeBackMsgForCall	64
sendCallMeBackMsgNoCall	65
Mitel OIG gives an error UNSUPPORTED_BY_PBX.setAccountCode	65

setCallMeBack	66
setCFNAExternal()	67
setCFNAInternal()	67
setCFBusyExternal()	68
setCFBusyInternal()	68
setCFAlways()	69
setDeviceDND()	70
setPrgPresence()	70
splitConferenceCall()	71
stopFeatureMonitor	71
stopMonitor	72
stopPRGPresenceMonitor	72
tradeCall()	73
transferCall()	73
verifyHotDeskUserPin()	74
Call Control Service Operations – Advanced Type	75
advACDSilentMonitor	75
advACDRequestHelp	78
advAssignCallerId	78
advCancelACDResezeWorkTimer	79
advCancelACDWorkTimer	80
advCancelTrunkMakeBusy	80
advCoachCall	81
advGetACDAgentDescription	82
advGetACDAgentId	83
advGetACDAgentStatus	84
advGetACDGroupStatus	85
advGetACD2GroupDescription	86
advGetACD2GroupId	87
advGetACD2PathDescription	87
advGetACD2PathId	88
advGetACDXGroupId	89
advGetCallStatus	89
advGetHuntGroupDescription	91
advGetHuntGroupId	92
advGetRingGroupDescription	92
advGetRingGroupId	93
advGetSipPeerDescription	94
advGetSipPeerId	94
advGetSipPeerStatus	96
advGetSystemConferences	97
advGetTdmTrunkId	98
advGetTdmTrunkStatus	98
advIntrudeCall	99

advLoginACDAgent	100
advLoginExternalACDAgent	100
advLogoutACDAgent	101
advLogoutExternalACDAgent	101
advMonitorACDExpress	102
advMonitorACD2AgentFeatures	103
advMonitorAllBridges	104
advMonitorAllTrunkFeatures	104
advMonitorConferences	105
advSetACDAgentMakeBusy	106
advSetACDAgentMakeBusyWithCode	107
advSetACDAgentPresence	107
advSetACD2AgentPresenceAllGroups	108
advSetAutoAnswer	108
advSetGroupPresence	109
advSetMsgWaitingInd	110
advSetTrunkMakeBusy	110
advStopMonitorACD2AgentFeatures	111
advStopMonitorACDExpressObjects	111
advStopMonitorAllBridges	112
advStopMonitorAllTrunkFeatures	113
advStopMonitorConferences	113
advTapCall	114

CALL CONTROL SERVICE EVENTS..... 116

Call Control Service Events – Standard Type	116
Standard Event Classes	116
Events in Standard Call Event Class	116
AccountCodeSetEvent	117
CallClearedEvent	117
CallConferencedEvent	118
CallDeliveredEvent	120
CallDivertedEvent	121
CallEstablishedEvent	122
CallFailedEvent	124
CallHeldEvent	125
CallOriginatedEvent	126
CallQueuedEvent	127
CallReceivedEvent	128
CallRetrievedEvent	131
CallTransferredEvent	132
ConferenceHeldEvent	133
DeviceDroppedEvent	134
ExtensionInUseEvent	135

GroupEvent	136
InformationUpdateEvent.....	137
InServiceEvent	137
MonitorFailedEvent	138
MonitorSetEvent	138
OutOfServiceEvent.....	139
PrgMonitorFailedEvent.....	139
PrgMonitorSetEvent	139
ResilientDeviceEvent	140
RoutingDeviceEvent.....	140
Events in Standard Feature Event Class.....	142
ActivateFeatureEvent	142
AutoAnswerFeatureEvent	143
ForwardFeatureEvent.....	143
HotDeskUserFeatureEvent	144
JoinGroupFeatureEvent	144
LeaveGroupFeatureEvent.....	144
UserIDFeatureEvent.....	145
Events in Standard System Event Class	145
ICPCommunicationEvent	145
OIG GatewayShutdownEvent	146
Call Control Service Events – Advanced Type	146
Advanced Event Classes	146
Events in Advanced Call Event Class	147
ACDExpressGroupEvent.....	147
TrunkDigitsOutpulsedEvent.....	148
WorkTimerExpiredEvent	149
Events in Advanced Feature Event Class	149
ACD_AGENT	149
ACD_AGENT_BLOCKED	151
ACD2_LOGIN_NO_LICENSE.....	151
BRIDGE_CONNECTED	151
BRIDGE_DISABLE_COACHING	152
BRIDGE_DISCONNECTED.....	154
BRIDGE_ENABLE_ALL_COACHING	154
BRIDGE_ENABLE_COACHING	155
BRIDGE_JOIN	156
BRIDGE_LEAVE	158
JOIN_ALL_ACD_GROUPS.....	159
LEAVE_ALL_ACD_Groups.....	159
Events in Advanced System Event Class.....	159
Events in Advanced ACD Event Class	159

ACD2PathEvent.....	160
ACD2GroupEvent	162
Events in Advanced Conference Event Class	163
ConferenceFeatureEvent.....	163
CALL CONTROL SERVICE STATES (FOR BOTH STD. AND ADV.)	166
Feature States	167
Device States	167
Call Control Service Event Causes	167
Call Control Service Event Data.....	177
Common Event Data	177
Class-Specific Data	178
CallEventClass Data.....	178
FeatureEventClass Data.....	179
EventFeature	180
EventState and EventDeviceNumber	181
Event-Specific Data	181
DESIGN AND ERROR HANDLING GUIDELINES.....	182
Call Control Service Error Definition.....	184
General Errors	184
Communication Errors	184
Service-Specific Errors	185
Call Forwarding incorrect event data.....	186
Calling Party Number (CPN) Substitution.....	186
Session Communication Security	187
ANI and DNIS Information in Events	187
Example 1: Remote Transfer (Unsupervised)	187
Example 2: Remote Transfer (Supervised)	187
Example 3: Remote Transfer Back to originating switch (Supervised)	188
Example 4: Remote Transfer Back to Originating switch (Unsupervised).....	188
Call Flow Exceptions for Call Park and Paging	188
Monitor All ACD Agents in One MiVoice Business	188
DTMF Generation	189
DTS Programmed on a Phone Key/Button (Advanced).....	189
Trunk Monitor Events.....	189
Device Data	190
Identifier for Phone Non-prime Line	191
Monitoring IP Trunks.....	191

Monitoring Limits.....	191
Redirecting Call on Line Appearance	191
Missing Called Number in SIP Peer Monitor Events.....	191
Device Attributes in Call Clear Event.....	192
IP RADs (Recorded Announcement Devices)	192
Operation.....	192
Call Received Event to RAD	193
Silent Monitoring an Agent	193
Calling Multicall Groups	194
Identity of Physical Device Answering a Call.....	194
MiVoice Business DPNSS and Feature Indications.....	194
General Notes.....	195
MITEL OIG OPERATION CONSIDERATIONS.....	198
HUNT GROUPS, RING GROUPS, ACD GROUPS AND GROUP PRESENCE....	199
Group Presence.....	199
Group Definitions	199
Hunt Groups	199
Ring Groups	199
Personal Ring Groups	200
Groups and Group Presence Monitoring	201
Ring Group and Hunt Group	201
Personal Ring Groups	201
Group Presence	202
Personal Ring Groups Presence.....	202
External Hot Desk User (EHDU)	202
ACD Express Groups.....	203
Hot Desk ACD Express Agents.....	203
Monitoring Support for ACD Express	203
Interflow	204
Resiliency	204
External Hot Desk Agent (EHDA).....	204
EHDA External Login	205
EHDA Behavior Summary.....	205
Permanent login COS is ignored for an EHDA	207
Programmable Reseize Timer for ACD group calls to an EHDA	207
EHDA must manually answer and hang-up calls.....	208
Answer Confirmation and Auto Logout Interaction	208
Attendant Call Information Display.....	209

Call Forwarding	209
Call Park	209
Call Park Retrieve	209
Call Recognition Services (CRS) Integration.....	209
Campon	209
Class of Restriction	209
Class of Service	210
Direct Page	210
Emergency Call.....	210
Hold.....	210
Interflow Timer Expiry While Ringing An EHDA	210
Message Waiting	210
Music on Hold on Transfer.....	210
Personal Speedcalls	210
PIN Lockout Support.....	210
Devices supported for supervisor (no EHDU supervisor support).....	211
Supervisor and agent must be hosted by the same MiVoice Business.....	211
Existing Monitoring Support for EHDA.....	211
External ACD Agent Login.....	211
Feature Events for EHDA	212
ActivateFeatureEvent	212
ForwardFeatureEvent	213
Group Presence Feature Events	213
UserIDFeatureEvent	213
Application Scenarios for EHDA.....	213
Silent Monitor	213
Request Help	215
Agent Auto Logout on No Answer	216
Unsupported Features for EHDA.....	217
CALL FLOW EXCEPTIONS FOR CALL PARK AND PAGING	220
Loudspeaker Page.....	220
Direct Page	220
Call Park and No Page	220
Call Park and Direct Page	220
Call Park and Loudspeaker Page.....	221
Call Park Retrieve	221
SIP DEVICE SUPPORT	222
GLOSSARY	225

Introduction

The Mitel Open Integration Gateway (Mitel OIG) is a web server that provides each Mitel OIG application a single point of access to web services available within a Mitel communication system.

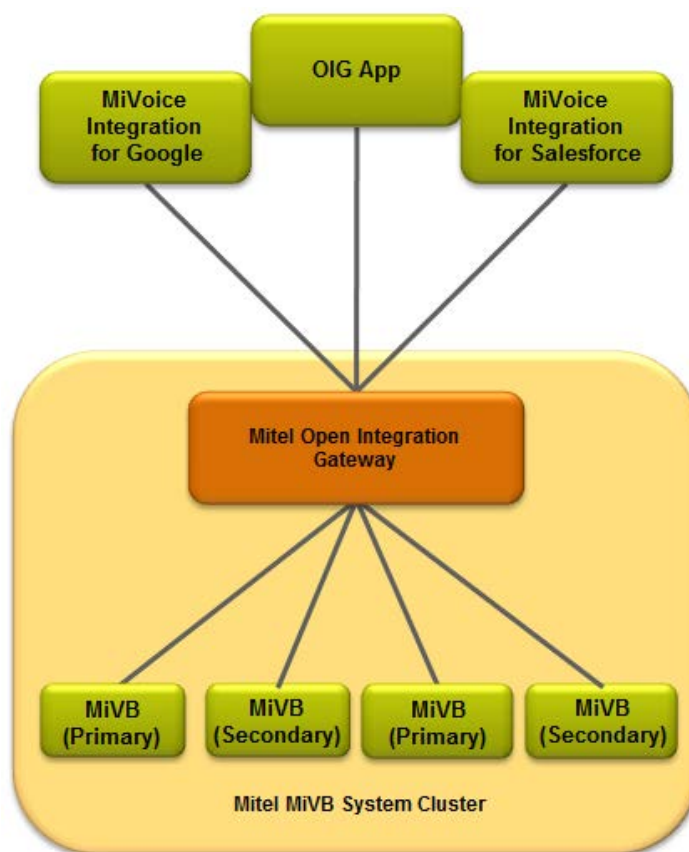
The Mitel OIG uses a services-oriented architecture. A Mitel OIG application opens a communication session with a Mitel OIG by logging in (example: sending a service operation or request to the Mitel OIG). After the Mitel OIG application is authenticated and authorized, the application can use this one communication session to access all Mitel OIG web services the application is authorized to use.

The Mitel OIG allows applications to access features and functionality offered by a MiVoice Business system cluster.



Note: The Mitel OIG can communicate with a single MiVoice Business or a cluster of MiVoice Business controllers. When there are two or more MiVoice Business controllers, the MiVoice Business controllers must be configured in a cluster. Mitel Open Integration Gateway cannot communicate with more than one MiVoice Business cluster. The Mitel OIG assumes the directory number (DN) of a Mitel phone is unique in the MiVoice Business system cluster; two Mitel phones in the system cannot have the same DN.

Figure 1: Mitel OIG system configuration



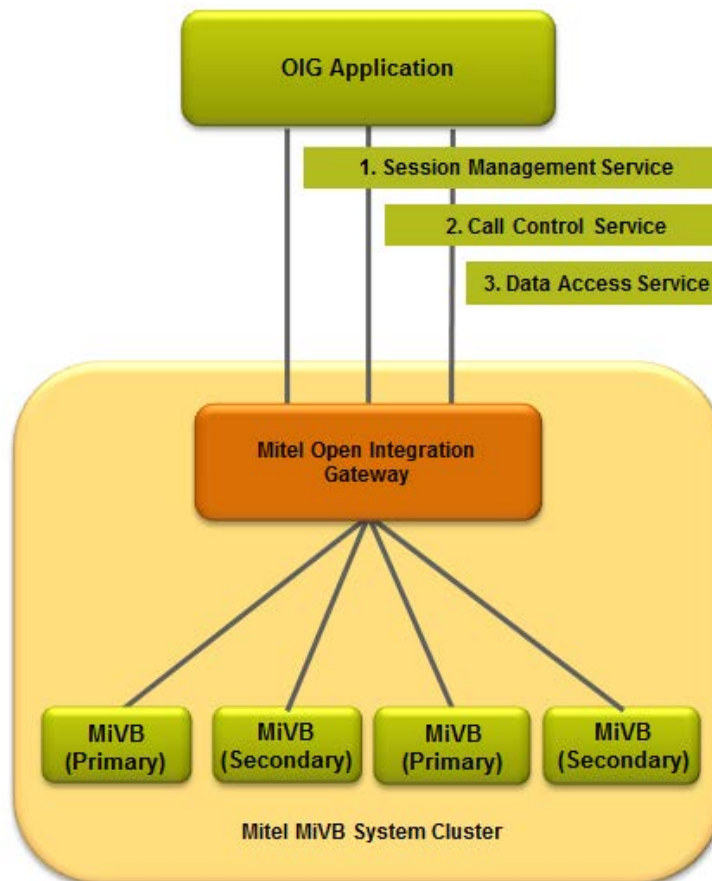
Mitel OIG web services

The Mitel OIG supports the following web services:

- Session Management service - Open communication session with Mitel OIG for services
- Call Control service - Control and monitor CTI behavior in Mitel communication system
- Data Access service - Register for MiVoice Business configuration data change notifications and read and write MiVoice Business configuration data.

Mitel also offers applications that are integrated with the Mitel OIG:

- MiVoice Integration for Salesforce
- MiVoice Integration for Google

Figure 2: Mitel OIG application, server, and services relationship

Mitel OIG documentation

This developer's guide is specific to the Mitel OIG Call Control Service. The following Mitel OIG documents are also available.

- **OIG Installation and Maintenance Guide:** This document provides details and instructions for installing the Mitel OIG and licensing it for applications and services, including MiVoice Integration applications.
- **OIG Engineering Guidelines:** The Engineering Guidelines provides guidance on network and system level requirements and performance.
- **OIG Developer Guide—Fundamentals:** The fundamentals guide introduces the Mitel OIG application developer environment and general information that applies to developing applications relative for any of the Mitel OIG web services. The fundamentals document also includes a summary of the changes introduced for this release.



Note: Mitel recommends that you become familiar with the content of the *Mitel OIG Developer Guide - Fundamentals* before attempting to create Mitel OIG applications.

- **OIG Developer Guide—Session Management Service:** This developer guide provides application developers detailed requirements for working with the Session Management Service.

- **OIG Developer Guide—Data Access Service:** This developer guide describes the Data Access Service details needed for creating applications that receive MiVoice Business Data.

Mitel Solutions Alliance (MSA) portal

Mitel Solutions Alliance (MSA) is the Mitel developer partner program. MSA delivers resources to enable a wide range of third-party partners (3PPs) to successfully create applications and services that integrate and interoperate with Mitel's core business communications portfolio. The program also helps create awareness of these products and services among Mitel resellers and customers.

The following MSA resources exist for OIG.

- OIG developer documentation
- WSDL files for Mitel OIG web service operations and events
- Mitel OIG Sample Application Package

WSDL Files for Mitel OIG Operations and Events

Mitel provides WSDL files that define the Mitel OIG operations, responses, events, and data for each Mitel OIG Web Service. Refer to the Mitel MSA MOL web portal for the WSDL files related to Mitel OIG Call Control Service.

Messaging Formats (SOAP and REST)

For information about web service messaging formats, refer to the *Mitel OIG Fundamentals Guide*.

In Mitel OIG, the call control service is only offered using SOAP / XML.

Mitel OIG Call Control Service Overview

The Mitel 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 Mitel OIG responds with success/failure and return data. The application must check for the success or failure of each operation. Operations trigger changes in the Mitel OIG and the MiVoice Business system. The changes are reported back to the application using events. An application must check the events returned following an operation on a device before performing more operations on the same device. An application can poll the Mitel OIG for events or provide an asynchronous reporting mechanism. To provide an asynchronous reporting mechanism, the application must register an event handler with the Mitel OIG. The Mitel OIG can then send events to the application asynchronously; this reporting mechanism is described in the `registerEventHandler` service operation.

The Call Control Service allows an application to control and monitor an IP Phone. The application must know information about the MiVoice Business system (i.e., MiVoice Business IP addresses, IP Phone prime line numbers, IP Phone line appearance button numbers, Hotdesk user numbers, and so on). The required information required in the application is defined in each operation within this document.

Standard and Advanced Call Control Services

The Mitel OIG Call Control Service functionality is offered in two types:

Standard Call Control Service

The Mitel OIG Standard Call Control Service allows an application to monitor and control the telephony activity of Mitel physical and logical devices (devices programmed or configured in MiVoice Business) including IP phones, Personal Ring Groups and line appearances on multi-line phones. The Mitel OIG Standard Call Control Service allows applications to control and monitor a Mitel desktop phone similar to a user manually controlling the phone.

Advanced Call Control Service

The Mitel OIG Advanced Call Control Service includes third-party call control capabilities and offers a full suite of functionality from simple call control to contact center monitoring and control. Advanced Call Control Service provides monitoring and control of MiVoice Business functions, e.g., Hot Desk Agent login (Internal and External), Trunking, Ring Groups, Hunt Groups, ACD2, ACD Express. Control relates to functions not normally associated with a specific desktop phone user. Support for MiVoice Business level monitoring (e.g., all conferences within a MiVoice Business) is included. Setting the phone message waiting lamp and auto answer are also only provided in Advanced Call Control.

Call Control Service Key Concepts

Mitel OIG Object IDs – The Mitel OIG allows an application to request an Object ID for each object of interest (i.e., MiVoice Business, IP Phone prime line, IP Phone line appearance button, Hotdesk user). The Mitel 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.

Mitel OIG Operations – The Mitel OIG offers web services. Each service offers an application many possible operations. Operations are commands to the Mitel OIG. For example, an application requests an Object ID for a MiVoice Business using the `getIcpld` operation. The Mitel OIG Call Control Service is divided into Standard and Advanced operations. The Advanced operations offer more complicated operations. Therefore, a deeper knowledge of MiVoice Business behavior is required to use the advanced operations. A developer must specifically request the use of Standard or Advanced services when registering an application with Mitel. Refer to the *Mitel OIG Installation and Maintenance Guide* for instructions on registering an application with Mitel).

Mitel OIG Monitors – To control and monitor an object (e.g., IP Phone), the application must open a monitor (i.e., the application initiates a monitor operation (command) with an IP Phone prime line Object ID). After 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).

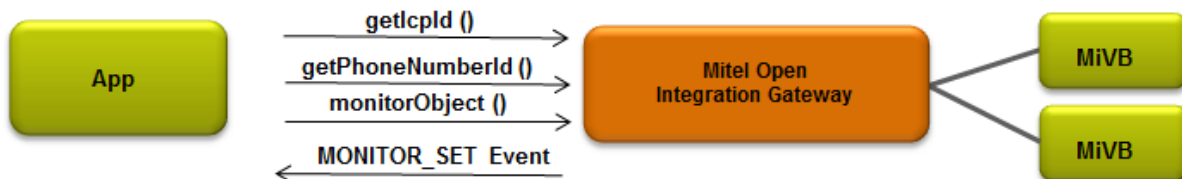
Mitel OIG Events – The Mitel OIG informs an application about status and state changes in objects (e.g., IP Phone prime line) using events. Each event has a name and associated event data. Each event also has a class name that defines the event data to expect. The event data provides specific details about monitored objects. An application uses events to confirm that initiated commands were successful and to determine what commands to initiate next.

Mitel OIG Event Data – Each Mitel 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. A Mitel OIG event also has a time stamp. The time format is `time_t`, the number of seconds elapsed since 00:00 hours, Jan 1, 1970 UTC. The time applied is from the Mitel OIG server and not from an individual MiVoice Business.



Note: The Event Time in 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



Overview of Mitel OIG Call Control Service monitoring feature

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. Applications can also open monitors on phone features and receive events about phone feature state changes, like the phone Do-Not-Disturb (DND) status, for example.

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 a MiVoice Business system cluster, the Mitel OIG provides a Global Call ID in call status events. See the Global Call ID Developer Guide on the MSA web portal for more information.

When using Mitel OIG, an application must provide the IP address of each MiVoice Business that provides devices to be monitored. If the MiVoice Business system supports MiVoice Business IP Phone Resiliency, the application needs to have the IP Addresses of both the primary and secondary MiVoice Business instances. The Mitel OIG must connect to a MiVoice Business to control and monitor the physical and logical devices configured in that MiVoice Business. An application cannot connect to MiVoice Business A to monitor and control a device configured in MiVoice Business B; in this case MiVoice Business A has no knowledge of the device on MiVoice Business B.



Note: MiVoice Business is the Mitel call control software. It can be deployed on the Mitel 3300 ICP, on industry-standard servers, or in a virtual environment. The MiVoice Business software is also used to run the controller instances in MiVoice Business Multi Instance (which runs on Industry Standard Servers).

Mitel OIG Call Control Service Components

The Mitel OIG Call Control Service uses:

- a server component in each MiVoice Business.
- a client component in each Mitel OIG.



Note: Newer releases of server component software in a MiVoice Business support n-1 older releases of the Mitel OIG client component software. However, only the functionality associated with the older Mitel OIG client component version will be available.

Newer releases of client component software in a Mitel OIG support n-1 older releases of the MiVoice Business server component software. When a newer version of the Mitel OIG client component software is needed, the associated MiVoice Business server component version should be upgraded to align with the Mitel OIG client component software version.

Mitel OIG Call Control Service Device Support

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

- Mitel IP Consoles and Attendant Consoles (not supported)
- Mitel 500X, 520X, and 530X IP Phones (not supported)
- IP trunks (not supported)
- SIP devices have limited support and different function compared to Mitel IP Phones.
- ONS phones have limited support and different function compared to Mitel IP Phones.



Note: See SIP Device Support.

MiVoice Business Configuration

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



Note: If either of these options is disabled for a device, then the Mitel Open Integration Gateway operations on the device does not function correctly.

CLASS OF SERVICE (COS) OPTIONS THAT MUST BE SET FOR EACH MONITORED DEVICE

HCI/CTI/TAPI Call Control Allowed: Yes

HCI/CTI/TAPI Monitor Allowed: Yes

EXAMPLES

Routing Devices use COS #1 only. If you are monitoring a routing device, COS #1 must have options set as indicated above.

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 have options set as indicated above.

When monitoring ACD agents, the COS for each monitored agent must have options set as indicated above.

When monitoring trunks, the COS for each monitored trunk must have options set as indicated above.

When monitoring phones, the COS for each monitored phone must have options set as indicated above.

Get configuration and Get Status Operations now have MiVoice Business information

The responses related to Mitel OIG operations that get configuration and get status information will now include the following information.

- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business Web Service

Recent changes affecting the Mitel OIG Call Control Application

The *Mitel OIG Developers Guide - Fundamentals* provides a summary of all the changes introduced as part of this release.

Any changes in the Mitel OIG 3.0 call control service are included in the Mitel OIG 3.0 WSDL files. Mitel OIG Versions Compatibility

See the *Mitel OIG Engineering Guidelines* for a description of what WSDL versions work with what Mitel OIG versions and what MiVoice Business controller versions.

Mitel OIG Support for MiVoice Business IP Phone Resiliency

The Mitel OIG provides support for MiVoice Business IP Phone Resiliency. For a complete description of MiVoice Business IP Phone Resiliency see the *MiVoice Business Resiliency Guidelines*. This developer guide describes how an application using Mitel OIG can monitor and control MiVoice Business resilient objects (e.g., IP phones, hot desk users, personal ring groups) and non-resilient objects (e.g., SIP trunks).

When using Mitel OIG, an application must have the primary and secondary IP addresses of all MiVoice Business instances of interest in the MiVoice Business system cluster. If an application connection request to a Primary MiVoice Business fails (and the MiVoice Business has resilient objects), the application must know the secondary MiVoice Business IP address to be able to create a connection to that MiVoice Business. The application must have information about which MiVoice Business instances have resilient and non-resilient objects configured. Also, an application must be aware of the additional MiVoice Business resiliency information in Mitel OIG operation responses and Mitel OIG status events.

The Mitel OIG supports MiVoice Business IP Phone resiliency (Standard and Advanced applications) for:

- IP Phone prime line
- Prime Member of PRG
- Hot Desk User/Agent
- External Hot Desk User/Agent



Note: MiVoice Business instances with resilient IP Phones must be configured to use System Data Synchronization (SDS). Refer to the MiVoice Business System Administration Online Help and the *Using SDS Solutions Guide*.

The Mitel OIG does NOT support MiVoice Business IP Phone resiliency for:

- Phone line appearances
- MiVoice Business trunks
- SIP trunks
- ACD2 paths
- Traditional ACD2 Agents
- Groups (Hunt, Ring, voice, voice mail, ACD2)
- Mitel OIG class level monitors. The class level monitors are as follows:
 - monitorPRGPresence operation
Monitoring all PRGs in a MiVoice Business
 - advMonitorConferences operation
Monitoring all Conferences in a MiVoice Business
 - advMonitorACD2AgentFeatures operation
Monitoring the features of all ACD2 Agents in a MiVoice Business

- `advMonitorACDExpress` operation
Monitoring the features of all ACD Express Agents in a MiVoice Business
- `advMonitorAllTrunkFeatures` operation
Monitoring all Trunks on a MiVoice Business

If an application is using class level monitors and the application is monitoring resilient devices, the application should invoke class level monitors on the secondary MiVoice Business as well.



Note: Mitel OIG MiVoice Business IP Phone resiliency solution is provided by default; no additional AMC licenses are needed. An application does NOT need to request or reject MiVoice Business IP Resiliency; MiVoice Business IP Resiliency is always provided starting with Mitel OIG 1.2. No specific Mitel OIG configuration is required to enable MiVoice Business IP Resiliency.

How Mitel OIG supports MiVoice Business IP Phone Resiliency

The Mitel OIG simplifies the monitoring, control and event flow related to resilient MiVoice Business objects. Resilient MiVoice Business objects have a primary MiVoice Business and a secondary MiVoice Business.

When the primary MiVoice Business fails, the resilient objects move to the secondary MiVoice Business. When the primary MiVoice Business recovers the resilient objects move back to the primary MiVoice Business. The Mitel OIG manages some of the monitoring and control of objects during such fail over and fall back scenarios so that application support of MiVoice Business IP Phone Resiliency is easier.

The Mitel OIG reports call status events from both the primary and secondary MiVoice Business instances at the same time. Each event indicates which MiVoice Business originated the event.

When an application is monitoring resilient objects and the Mitel OIG connects to either the primary or secondary MiVoice Business, the Mitel OIG caches in memory the primary MiVoice Business and secondary MiVoice Business IP addresses for the resilient objects. The Mitel OIG memory cache is deleted on Mitel OIG power down or reset, so an application must know the primary and secondary IP addresses of all MiVoice Business instances of interest in the MiVoice Business system cluster. If an application connection request to a Primary MiVoice Business fails (and the MiVoice Business has resilient objects), the application must know the secondary MiVoice Business IP address to be able to create a connection to that MiVoice Business. The application needs to know what MiVoice Business instances have resilient and non-resilient objects configured. The Mitel OIG will auto-recover non-resilient monitors on non-resilient MiVoice Business objects when the primary MiVoice Business recovers.

Mitel OIG provides software logs for errors, warnings, and status messages related to IP resiliency.

The Mitel OIG ensures that the application receives only one object identifier (`objectId`) needed for monitoring and controlling a specific object in a MiVoice Business, regardless of whether the object is registered to the primary or secondary MiVoice Business. The application will always use the one `objectId` provided by a “`getObjectId`” type operation regardless of where the object is currently registered (Primary or Secondary MiVoice Business).



Note: When the Primary MiVoice Business is down, a Mitel OIG application cannot set a monitor on a phone line appearance using the Secondary MiVoice Business. Phone line appearances are not resilient. The Mitel OIG must be able to connect to the Primary MiVoice Business before an application can get an objectID on a phone line appearance when the phone is configured on the Primary MiVoice Business.

Mitel OIG provides support for MiVoice Business IP Phone Resiliency by default. As a result, additional information is provided in operation responses and status events from monitored objects. For example, the get device description operation response for phone prime line, ACD2 Agents, ACD2 Groups, Ring Groups, and Hunt Groups now has the MiVoice Business IP address and MiVoice Business ID for the primary, and in some cases, the secondary MiVoice Business instances. If a non-resilient device, the additional information is only provided for the primary MiVoice Business. If a resilient device, the information is provided for both the primary and secondary MiVoice Business instances. When an application is monitoring a resilient device, there can be Mitel OIG events generated from both the primary and secondary MiVoice Business instances, so an application needs to check event information to determine which MiVoice Business generated the Mitel OIG response or event.

When an application is monitoring a resilient External Hot Desk User (EHDU), and/or an external Hot Desk Agent (EHDA), and a non-resilient SIP Peer (SIP trunk), and the monitors are recovered on the Primary MiVoice Business, there may be no Mitel OIG IN SERVICE events reported from the monitors on the Primary MiVoice Business. The Mitel OIG will recover the monitors on these devices on the Primary MiVoice Business. However, there will be no IN SERVICE events if the monitors are recovered on the Primary MiVoice Business after the EHDU, EHDA and SIP trunk come in service on the primary MiVoice Business. The application can use the Mitel OIG advGetSipPeerStatus operation and the Mitel OIG getResilientStatus operation to determine the in service state of the monitored object.

There may be no Mitel OIG IN SERVICE event on a monitored ring group or hunt group when the group fails back to the primary MiVoice Business and the Mitel OIG recovers the monitors. Consider the following scenario:

1. Application gets icpld of primary MiVoice Business.
2. Application gets objectId of resilient device A,B,C where B is a member of Hunt Group H1 and C is a member of Ring group R1
3. Get ring group ID for R1 and hunt group ID for H1
4. Start monitors on A,B,C, R1, and H1
5. Two monitor set events are returned for A,B and C (Mitel OIG created monitors on both primary MiVoice Business and secondary MiVoice Business for these resilient devices)
6. Only one monitor set event was created for non-resilient H1 and one for non-resilient R1 on the primary MiVoice Business
7. Power down the primary MiVoice Business.
Event type: ICP_COMMUNICATION_EVENT
Connection State : FAILURE for primary ICP is returned
8. OUT OF SERVICE events for A,B,C,R1 and H1 are returned from primary MiVoice Business and IN SERVICE events for A,B and C are returned from secondary MiVoice Business because device failover to secondary MiVoice Business
9. Power up the primary MiVoice Business
Event type: ICP_COMMUNICATION_EVENT
Connection State : CONNECTED for primary MiVoice Business is returned

10. MONITOR SET events for A,B,C,R1 and H1 are returned for primary MiVoice Business
11. OUT OF SERVICE events for A,B,C for secondary MiVoice Business and IN SERVICE events for A,B and C for primary MiVoice Business are returned as devices failback to primary MiVoice Business along with RESILIENT DEVICE REDIRECTED HANDOFF event for resilient devices.
12. There may be IN SERVICE events for ring group R1 and hunt group H1 depending on when the objects come in to service on the primary MiVoice Business. If in service before Mitel OIG creates the associated non-resilient monitors there will be no in service events for R1 and H1.

Resilient Scenarios

The following sections describe a few different scenarios related to Mitel OIG support for MiVoice Business IP Phone Resiliency. There are many resiliency scenarios possible; these few are provided as a brief introduction. Mitel OIG application developers are encouraged to execute different resilient scenarios in a design lab to learn the many Mitel OIG event flows.

Scenario 1: Application #1 starts a monitor on phone prime line 2000

OIG:

- Application gets the ICP ID for Primary MiVoice Business
- Application gets the Object ID for 2000
- OIG stores in memory the device's primary and secondary MiVoice Business details
- Application starts monitor on 2000
- OIG starts monitor on 2000 in primary MiVoice Business
- OIG returns MonitorSet event for primary MiVoice Business
- OIG starts monitor on 2000 in secondary MiVoice Business
- OIG returns MonitorSet event for secondary MiVoice Business

OIG Event Flow:

- GetIcpId was successful: 1
- getPhoneNumberId was successful phone DN: 2000 id: 9
- GetDeviceConfigurationEx was successful isResilient: True
- Monitor set on object id: 9
- Received Event type: CALL_EVENT Call Event Type: MONITOR_SET
- localCallId: 0 ObjectId: 9
- Call State: NIL Event Cause: UNKNOWN
- Attribute Name: ICP_ID Attribute Value: 1
- No devices included.
- Received Event type: CALL_EVENT Call Event Type: MONITOR_SET
- localCallId: 0 ObjectId: 9

- Call State: NIL Event Cause: UNKNOWN
- Attribute Name: ICP_ID Attribute Value: 2
- No devices included.



Note: Phone prime line 2000 will have different object IDs for the primary and secondary MiVoice Business instances, but this is hidden from the application. The application receives the same objectId regardless of which MiVoice Business is active. Even though line 2000 is being monitored on both MiVoice Business instances, the MiVoice Business where the phone is currently registered generates most of the events.

Scenario 2: Monitor on phone prime line 2000 fails over to secondary MiVoice Business

OIG:

- OIG sends application SystemEvent indicating that the connection to the primary MiVoice Business has failed
- Application will start getting events from secondary MiVoice Business once the phone registers with the secondary MiVoice Business
- OIG will try to restore connection to primary MiVoice Business periodically
- OIG sends application SystemEvent indicating that the connection to the primary MiVoice Business is restored
- OIG will restart the monitor on phone 2000 once the connection is restored
- OIG sends MonitorSet event to application for primary MiVoice Business
- OIG will continue sending call control operations to the secondary MiVoice Business until the phone goes into service on the primary MiVoice Business
- When the phone returns to the primary MiVoice Business, the application will receive the following events:
- RESILIENT_DEVICE event Cause REDIRECTED_HANDOFF from the secondary MiVoice Business
- OUT_OF_SERVICE event from the secondary MiVoice Business
- IN_SERVICE event from the primary MiVoice Business

OIG Event Flow:

Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTION_FAILURE

Received Event type: CALL_EVENT Call Event Type: OUT_OF_SERVICE

localCallId: 0 ObjectId: 9

Call State: UNAVAILABLE Event Cause: CONTROLLER_COMMS_FAILED

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

Received Event type: CALL_EVENT Call Event Type: IN_SERVICE

localCallId: 0 ObjectId: 9

Call State: IDLE Event Cause: DEVICE_IN_SERVICE

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTED

Received Event type: CALL_EVENT Call Event Type: MONITOR_SET

localCallId: 0 ObjectId: 22

Call State: NIL Event Cause: UNKNOWN

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

Received Event type: CALL_EVENT Call Event Type: RESILIENT_DEVICE

localCallId: 0 ObjectId: 22

Call State: UNAVAILABLE Event Cause: REDIRECTED_HANDOFF

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event type: CALL_EVENT Call Event Type: OUT_OF_SERVICE

localCallId: 0 ObjectId: 22

Call State: UNAVAILABLE Event Cause: DEVICE_OUT_OF_SERVICE

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event type: CALL_EVENT Call Event Type: IN_SERVICE

localCallId: 0 ObjectId: 22

Call State: IDLE Event Cause: DEVICE_IN_SERVICE

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

Scenario 3: Application starts up when primary MiVoice Business is down and phone 2000 is registered with the secondary MiVoice Business

OIG:

- Application attempts to get the ICP ID for Primary MiVoice Business
- Failure returned
- Application attempts to get the ICP ID for Secondary MiVoice Business
- OIG returns ICP ID for secondary MiVoice Business
- Application gets the Object ID for 2000
- OIG stores in memory the device's primary and secondary MiVoice Business details
- Application starts monitor on 2000
- OIG starts monitor on 2000 in secondary MiVoice Business
- OIG returns MonitorSet event for secondary MiVoice Business
- OIG attempts to open connection to primary MiVoice Business and will continue trying at defined rate
- Once connection is established Mitel OIG starts monitor of 2000 on primary MiVoice Business
- OIG returns MonitorSet event for primary MiVoice Business
- At some point the phone will be redirected back to the primary MiVoice Business

OIG Event Flow:

Failed to get the ICP ID error: Connection not up to MiVoice Business.

GetIcpId was successful: 2

getPhoneNumberId was successful phone DN: 2000 id: 28

GetDeviceConfigurationEx was successful isResilient: True

Adding ICP to list: 1

addEventToQueue invoked.

Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTION_FAILURE

Monitor set on object id: 28

addEventToQueue invoked.

Received Event type: CALL_EVENT Call Event Type: MONITOR_SET

localCallId: 0 ObjectId: 28

Call State: NIL Event Cause: UNKNOWN

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTED

Received Event type: CALL_EVENT Call Event Type: MONITOR_SET

localCallId: 0 ObjectId: 28

Call State: NIL Event Cause: UNKNOWN

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

Received Event type: CALL_EVENT Call Event Type: RESILIENT_DEVICE

localCallId: 0 ObjectId: 28

Call State: UNAVAILABLE Event Cause: REDIRECTED_HANDOFF

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event type: CALL_EVENT Call Event Type: OUT_OF_SERVICE

localCallId: 0 ObjectId: 28

Call State: UNAVAILABLE Event Cause: DEVICE_OUT_OF_SERVICE

Attribute Name: ICP_ID Attribute Value: 2

No devices included.

Received Event type: CALL_EVENT Call Event Type: IN_SERVICE

localCallId: 0 ObjectId: 28

Call State: IDLE Event Cause: DEVICE_IN_SERVICE

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

Mitel OIG IP Device Resiliency Impact on Application

For the most part, applications do not behave any differently when dealing with resilient devices. This section highlights some of the functionality that applications must be aware of.

Applications should be aware of the primary and secondary MiVoice Business

The Mitel OIG will automatically start monitors on the primary and secondary MiVoice Business instances when an application starts a monitor on a resilient device. Therefore, the Mitel OIG generates events from both MiVoice Business instances.

For example, if an application started to monitor a resilient prime line on its primary MiVoice Business, the Mitel OIG automatically starts monitoring the resilient prime line on the secondary MiVoice Business as well. Once the monitor is started on either MiVoice Business, the Mitel OIG sends the application a MONITOR_SET event. Therefore, in this scenario, the application gets two MONITOR_SET events.



Note: The application uses one object ID to identify the phone prime line in all events and to invoke call control operations for both MiVoice Business instances. The application does not need to direct a Mitel OIG operation to the correct MiVoice Business. This is handled by the Mitel OIG.

Applications that track Device State

Applications that track device state need to track the device state on both the primary and secondary MiVoice Business instances and not rely on timing of events. It's possible that the application could get a phone IN_SERVICE event on the primary MiVoice Business followed by an OUT_OF_SERVICE event for the same phone from the secondary MiVoice Business. Each event indicates which MiVoice Business generated the event. An application must be aware of the primary and secondary MiVoice Business instances.

A Mitel OIG operation called "getResilientStatus" can be used by applications to determine primary and secondary MiVoice Business IDs, MiVoice Business IP addresses, object connection status to the Mitel OIG, object state and which MiVoice Business is active (i.e. where object is currently connected).

How to Determine If a Device is Resilient

Applications can use one of the following operations to determine if a device is resilient:

getDeviceConfigurationEx:

Input parameters:

- long sessionId
- long icpld
- String primeDn
- Returns:
- Boolean result -- Operation result
- String errorDescription -- Error description if operation failed
- Boolean isResilient
- String primaryIcpAddress
- Long primaryIcpId
- String secondaryIcpAddress
- Long secondaryIcpId
- LineConfig array describing lines configured on device

getResilientStatus:

Input Parameters:

- Long sessionId
- Long objectId
- Output Parameters:
- Boolean result -- Operation result
- String errorDescription -- Error description if operation failed
- Boolean isResilient
- Long primaryIcpId
- String primaryIcpAddress
- ConnectionState primaryConnectionState
- CallState primaryCallState
- Long secondaryIcpId
- String secondaryIcpAddress
- ConnectionState secondaryConnectionState
- CallState secondaryCallState
- Long ActiveIcpId

Call Event and Feature Event Changes

All call events and feature events include the attribute ICP_ID indicating which MiVoice Business generated the event. Duplicate feature events (same event with different ICP_ID) are observed when monitoring features on a resilient phone. For example, if an application is monitoring

features on a resilient phone and the phone is put into DND, a feature event is generated from the primary and secondary MiVoice Business instances.

Starting Monitor on Resilient Device When MiVoice Business is Down

When an application starts monitoring a resilient device and its secondary MiVoice Business is down, the Mitel OIG tries periodically to connect to the secondary MiVoice Business. If the Mitel OIG successfully connects to the secondary MiVoice Business, it starts a monitor on the device, and sends a `MONITOR_SET` event to the application. The reverse is true if the application started a monitor on a device's secondary MiVoice Business when the primary MiVoice Business is down.



Note: This is a situation where the application must be aware of the primary and secondary MiVoice Business instances. If the application fails to get the ID of the primary MiVoice Business because the primary MiVoice Business is down, the application should try to get the ID of the secondary MiVoice Business and use it to start any required monitors.

MiVoice Business and Mitel OIG Connection Failures

When a connection failure occurs between the Mitel OIG and a MiVoice Business, the application is sent a system event `ICP_COMMUNICATION_EVENT` which includes the MiVoice Business `lcpId` and the connection state `CONNECTION_FAILURE`. The application receives call events of event type `OUT_OF_SERVICE` for all device monitors that connected to the failed MiVoice Business.



Note: `OUT_OF_SERVICE` events for phones on a MiVoice Business are generated by the Mitel OIG.. If there's a network problem that prevents the Mitel OIG from accessing a MiVoice Business, but the phones can still access the MiVoice Business, the phones may still be `IN_SERVICE`. This is not a common situation, but it can occur. In most cases when the MiVoice Business connection fails, the phones become active on the secondary MiVoice Business and the application receives an `IN_SERVICE` call event from the secondary. After any MiVoice Business connection failure, the Mitel OIG tries periodically to reconnect to the MiVoice Business and recover all monitors.

MiVoice Business Connection Recovery

After monitors are started on a MiVoice Business, if the connection to the MiVoice Business fails or the MiVoice Business is restarted, the Mitel OIG tries periodically to reconnect to the MiVoice Business and recover all monitors. Monitors are only stopped when an application invokes the `stopMonitor` operation and there are no other sessions monitoring that device or the session is terminated due to inactivity and no other sessions are monitoring that device.

When a connection to a MiVoice Business is recovered, the application receives a system event `ICP_COMMUNICATION_EVENT` with a connection state set to `CONNECTED`. When the monitors are recovered the application receives a `MONITOR_SET` event.

If both connections to a device's primary and secondary MiVoice Business controllers fail, the Mitel OIG tries to recover both connections and restart the monitors.

Event RESILIENT_DEVICE with Cause REDIRECTED_HANDOFF

The Mitel OIG supports a RESILIENT_DEVICE event, which is generated when a device is connected to its secondary MiVoice Business and is being directed back to its primary MiVoice Business. When a device switches MiVoice Business instances there can be a small period of time where call control operations are not functional. After the application receives the IN_SERVICE event on the primary or secondary MiVoice Business, call control operations will be functional again.

Event RESILIENT_DEVICE with Cause RESILIENT_INFO_UPDATE

The Mitel OIG supports the RESILIENT_DEVICE event caused by RESILIENT_INFO_UPDATE. This event and cause are generated when a device being monitored has its resilient configuration changed on the MiVoice Business. Possible changes are:

- making the device resilient by assigning a secondary MiVoice Business
- changing the devices secondary MiVoice Business to a different MiVoice Business
- removing the secondary MiVoice Business (turning off resiliency).
- This event and cause contains the following information:
- Call Event Type: RESILIENT_DEVICE
- Event Cause: RESILIENT_INFO_UPDATE
- Object Id
- Event Time
- If secondary is assigned the secondary ICP_ID

When an application receives this event it should use the “getResilientStatus” operation to determine what has changed. The Mitel OIG will perform the appropriate actions based on the configuration change (i.e. start/stop monitors on the old/new secondary MiVoice Business instances).



Note: If resiliency is being enabled on a secondary MiVoice Business, the application receives a MONITOR_SET event on the new secondary MiVoice Business.

Mitel OIG Monitor Recovery

Depending on the failure scenario and timing of events, an application may not have an accurate view of a device’s state after some connection failures. In this case, an application must use the “getResilientStatus” operation. Typically this operation is used after a MONITOR_SET event.

For example: An application is monitoring a resilient device and the primary MiVoice Business goes down. The phone goes into service on the secondary MiVoice Business. At the same time, the Mitel OIG will be trying to recover its connection and monitors on the primary MiVoice Business. Then the connection fails to the secondary MiVoice Business and the Mitel OIG recovers connection and monitors on the primary MiVoice Business. The application is still receiving MONITOR_SET events on the primary MiVoice Business, but the application does not yet know the state of the phones until it uses the “getResilientStatus” operation. For this example, the “getResilientStatus” operation returns the following:

- Boolean result -- true

- String errorDescription -- NULL
- Boolean isResilient -- true
- Long primaryIcpId -- 2
- String primaryIcpAddress -- 10.38.103.20
- ConnectionState primary -- CONNECTION_FAILURE
- CallState primaryCallState -- NIL
- Long secondaryIcpId -- 3
- String secondaryIcpAddress -- 10.37.195.20
- ConnectionState secondary -- CONNECTED
- CallState secondaryCallState -- IDLE
- Long ActiveIcpId -- 3

From this information, the application can determine that the phone is still registered and active on the secondary MiVoice Business.

Mitel OIG Non-Resilient Monitors

If a non-resilient device is being monitored and the MiVoice Business goes down, or its network connection fails, the Mitel OIG tries periodically to recover its connection to the MiVoice Business. When it recovers its connection, it restarts the monitor. The following event flow illustrates what events the application receives:

Non-resilient Monitor Recovery Event Flow:

Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTION_FAILURE

Received Event type: CALL_EVENT Call Event Type: OUT_OF_SERVICE

localCallId: 0 ObjectId: 29

Call State: UNAVAILABLE Event Cause: CONTROLLER_COMMS_FAILED

Attribute Name: ICP_ID Attribute Value: 1

No devices included. Received Event Type: SYSTEM_EVENT System Event Type: ICP_COMMUNICATION_EVENT

ICPId: 1 Connection State: CONNECTED

Received Event type: CALL_EVENT Call Event Type: MONITOR_SET

localCallId: 0 ObjectId: 29

Call State: NIL Event Cause: UNKNOWN

Attribute Name: ICP_ID Attribute Value: 1

No devices included.

From the above events, the application could determine that the connection to the MiVoice Business was down and was subsequently recovered, along with the monitor. After receiving the MONITOR_SET event, the application could use either `getResilientStatus` or `getCallStatus` operations to determine the state of the device.

Mitel OIG Sample Scenarios

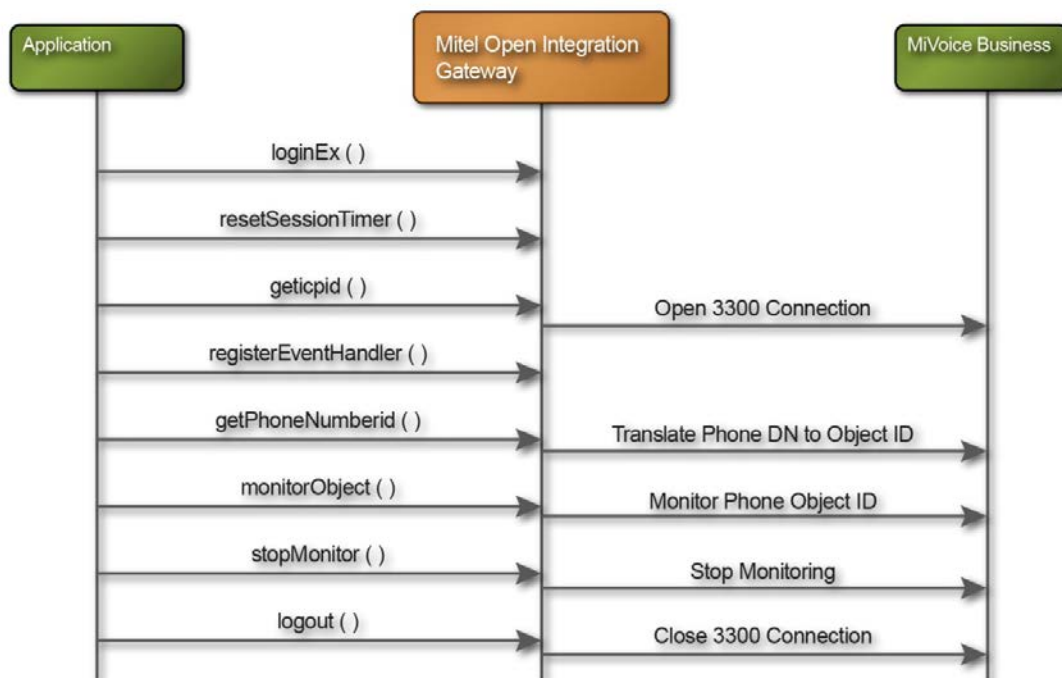
This section defines a few scenarios to introduce typical Mitel OIG behavior from an application viewpoint. See operation descriptions within this document and Mitel OIG WSDL for complete details.

To learn how to create an application that uses Mitel OIG Advanced Call Control Service using Mitel certificates, refer to the Mitel OIG Sample Applications Package. Included in the package are a sample application and an associated document. The document explains how to use the application and provides guidelines on how to create a similar application.

Create Phone Monitor

In this scenario, an application uses Session Service to open a communication session with the Mitel OIG. Next, the application opens a connection to a MiVoice Business and requests asynchronous event reporting. Then the application opens and closes a monitor on a phone.

Figure 4: Create phone monitor



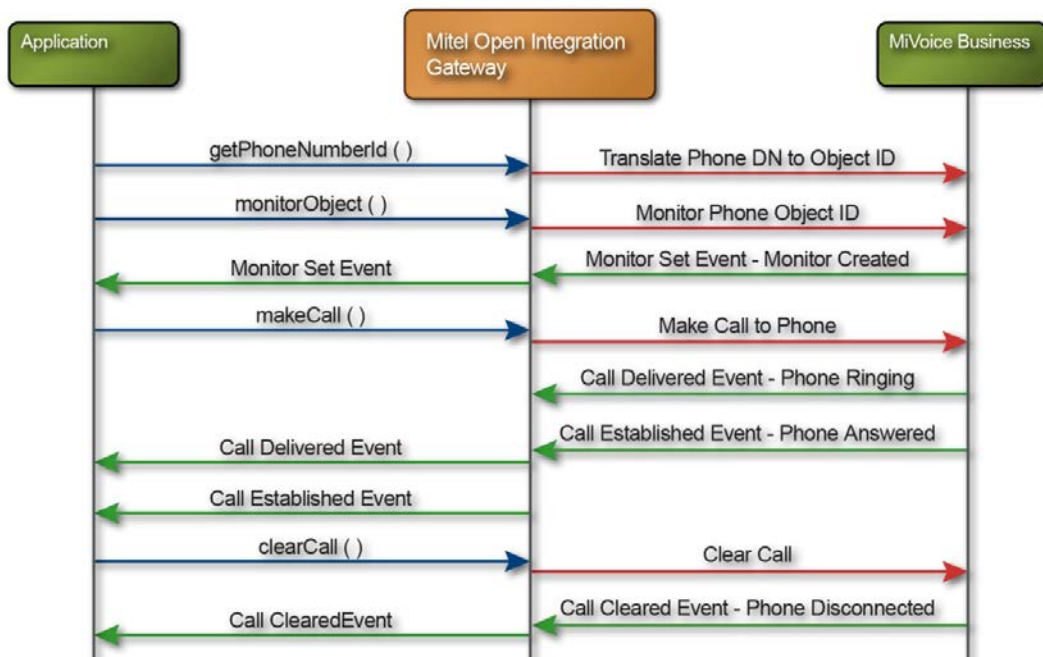
IP1775

Finally, the application closes the communication session with the Mitel OIG by logging out. Events associated with the operations are not shown.

Basic Call Monitoring

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

Figure 5: Basic call monitoring



IP1776

Call Control Service Operations

Call Control Service Operations allow an application to monitor and control MiVoice Business objects. There are two classes of operations.

- Standard service class operations
- Advanced service class operations

Standard Service Operations Overview

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

Table 1: Standard Call Control service operations

alternateCall	This operation 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 the device to the party on Consultation Hold.
clearCall	This operation releases the active 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 a MiVoice Business HCI routing device. To create an HCI Routing Device use the MiVoice Business Configuration UI and select “Users and Devices”, Group Programming, Hunt Groups, and then create a device with type HCIRoute.
conferenceCall	This operation allows a device to merge two calls (one active and one on consultation hold) into a single conference call.
consultationCall	This operation puts the active call on consultation hold and places a call to a provided phone number. This operation is usually the first step in performing a transfer or conference call.
getCallStatus	This operation allows the application to request the call state of a monitored device.
getIcpConfig	This operation is used to retrieve the E.164 settings configured in the Mitel OIG. This operation can be used after the application has successfully logged in to the OIG using standard call control service.
getDeviceConfigurationEx	This operation gets device configuration information about the monitored device. This includes the line appearance button numbers.

getDeviceFeatures	This operation gets the feature settings (call forwarding, Do-Not-Disturb, auto answer) for a phone DN. This operation can also be applied to a Traditional Agent, Internal hotdesk user (IHdu), Internal hotdesk agent (IHDA), External hotdesk user (EHdu), and External hotdesk agent (EHDA). For a complete description of the values returned in the operation response for hotDeskUserLoggedInDn and registrationDn see the getDeviceFeatures operation description later in the document.
getDeviceMembers	The "getDeviceMembers" operation can be used to determine if there are line appearances of a Prime DN on other telephones in the system.
getEvent	This operation is used by the application to synchronously call for events held in the Mitel OIG.. This operation is only used if the application polls for events. The registerEventHandler operation must be used when the application wants to receive events from the Mitel OIG asynchronously. The operation returns immediately if an event is pending. If no events are available, the operation waits for a set period of time and then returns with no event.
getHotDeskUserDn	This operation gets hotdesk user DN based on phone DN Object ID. If no hotdesk user is logged in the hotdesk user DN is blank.
getHotDeskUserLoginDevice	This operation gets device DN based on hotdesk user DN Object ID. If hotdesk user is not logged in the phone DN is blank.
getIcpld	This operation provides the object ID for a MiVoice Business. The application uses this MiVoice Business object ID in other operations to specifically identify a MiVoice Business to the Mitel OIG.. The application calls this operation to open a connection to a MiVoice Business. Note: An error condition occurs when an invalid MiVoice Business IP address is given.
getLineAppearanceld	This operation provides the object ID for a MiVoice Business phone line appearance. The application uses this line appearance object ID in other operations to specifically identify a MiVoice Business phone line appearance to the Mitel 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 MiVoice Business phone prime line. The application uses this phone number object ID in other operations to specifically identify a MiVoice Business phone prime line DN to the Mitel 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 logs 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 phone number provided. An application can also set a forced account code; see makeCall operation description for details.
monitorFeatures	This operation creates a device feature monitor. Device features include DND, auto answer, call forwarding, hotdesk user login and logout. The provided object ID identifies the object to be monitored. This operation instructs the MiVoice Business to report feature events when a device feature changes. The object ID also identifies the MiVoice Business 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 MiVoice Business 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 MiVoice Business. The object ID also identifies the MiVoice Business 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 class level monitor on a specific MiVoice Business. The resulting monitor reports all presence activities for all Personal Ring Groups configured on the MiVoice Business.
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 back in dialing state with a party still on hold). 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 Established State. Tones are not heard at the monitored device. An application cannot 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 pick up a call which has landed at a Routing Device; see the pickupCall operation description 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 Mitel OIG. The Mitel OIG uses the URL to send events asynchronously to the application. The application must process the https request from the Mitel 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.
remoteRetrieveCall	The remoteRetrieveCall operation is used to retrieve a call put on hold by one telephone to a different telephone. The COS option "Call Hold Remote Retrieve" must be set to "yes" and the feature access code "Call Hold – Remote Retrieve" must be set.

retrieveCall	This operation reconnects an existing held call at the specified holding device. The call was previously suspended using the holdCall or the MiVoice Business 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. When the held call is reconnected to this monitored device a new the local Call ID is created.
sendCallMeBackMsgNoCall	This operation allows a caller to leave a message at a device identified by deviceDn when the target device is not in a call..
setAccountCode	This operation is used to assign a non-verified account code to an established call on a monitored phone. 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 the account code is not valid (MiVoice Business does not check the account code).
setCallMeBack	This operation instructs a MiVoice Business to ring the monitored device with a distinctive ring pattern as soon as the called device that is busy becomes available. When the called device becomes available the MiVoice Business rings the monitored device. 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 monitored device setting the call me back and the local Call ID for call to busy device.
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 monitored device on or off. This operation includes the DN to be used when forwarding call.
setCFBusyExternal	This operation sets the call forwarding for a monitored device on or off related to an external call. If the monitored device 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 monitored device on or off related to an internal call. If the monitored device 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 monitored device on or off related to an external call. If the monitored device 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 monitored device on or off related to an internal call. If the monitored device does not answer when an internal call is received, the call is forwarded to the specified destination.
setDeviceDND	This operation sets or clears the monitored 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 class level monitor on a MiVoice Business that reports all presence activities 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 a party on Consultation Hold and the other party being called or in an active call with this monitored device. The requesting device ends its participation in both calls (held and active). Initiating this operation before the called party answers is referred to as unsupervised transfer. Initiating this operation after the called party answers is referred to as supervised transfer.
verifyHotDeskUserPin	This operation verifies a PIN used to log in a hotdesk user.

Advanced Service Operations Overview

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

Table 2: Advanced Call Control service operations

advGetTdmTrunkId	This operation provides the object ID for a MiVoice Business TDM trunk number. The application uses this trunk object ID in other operations to specifically identify a MiVoice Business trunk to the Mitel OIG. An application must call this operation before setting a monitor on the trunk.
advGetCallStatus	The “advGetCallStatus” operation is used by advanced applications to determine the call state of an object. If the object is in a call, then the operation provides details about the call such as localCallId, globalCallId, and trunkCallId if the call is using a SIP trunk.
advGetSipPeerId	This operation provides the object ID for an incoming MiVoice Business SIP Trunk. The application uses this SIP Peer object ID in other operations to specifically identify a MiVoice Business SIP trunk to the Mitel OIG. An application must call this operation before setting a monitor on the SIP Trunk.
advGetACD2PathId	This operation provides the object ID for a MiVoice Business ACD2 path. The application uses this path object ID in other operations to specifically identify a MiVoice Business path to the Mitel OIG. An application must call this operation before setting a monitor on the path.
advGetACD2GroupId	This operation provides the object ID for a MiVoice Business ACD2 group. The application uses this group object ID in other operations to specifically identify a MiVoice Business ACD2 group to the Mitel OIG. An application must call this operation before

setting a monitor on the group.

advGetACDAgentId	This operation provides the object ID for a MiVoice Business ACD2 agent. The application uses this ACD2 agent object ID in other operations to specifically identify a MiVoice Business ACD2 agent to the Mitel OIG. An application must call this operation before setting a monitor on the ACD2 agent.
advGetACDXGroupId	This operation provides the object ID for a MiVoice Business ACD Express Group. The application uses this ACD Express group object ID in other operations to specifically identify a MiVoice Business Express group to the Mitel OIG. An application must call this operation before setting a monitor on the ACD Express group.
advGetRingGroupId	This operation provides the object ID for a MiVoice Business Ring Group. The application uses this Ring Group object ID in other operations to specifically identify a MiVoice Business ring group to the Mitel OIG. An application must call this operation before setting a monitor on the ring group.
advGetHuntGroupId	This operation provides the object ID for a MiVoice Business Hunt Group. The application uses this Hunt Group object ID in other operations to specifically identify a MiVoice Business Hunt Group to the Mitel OIG. An application must call this operation before setting a monitor on the Hunt Group.
advMonitorConferences	This operation allows an application to create a MiVoice Business level monitor on a specific MiVoice Business for the purpose of receiving events about all conferences created on the MiVoice Business. This allows an application to create one monitor in place of having many monitors on the many objects involved in conferences.
advGetEvent	This operation is used by the application to synchronously call for events held in the Mitel OIG. This operation is only used if the application polls for events. The registerEventHandler operation must be used when the application wants to receive events from the Mitel OIG asynchronously. The operation returns immediately if an event is pending. If no events are available, the operation waits for a set period of time and then returns with no event.
advAssignCallerId	This operation is used by the application 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 called/calling phone. The trunk must be currently participating in the call and must be monitored by the application. After this assignment occurs the changed caller identity is displayed on the phone for the life of the trunk call. The caller identity is displayed on the phone but not provided in the Mitel OIG 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 on an incoming trunk call to enter their customer number through their DTMF keypad.
advIntrudeCall	This operation allows the caller to be conferenced into an existing conversation after an attempt to call one of the endpoints in the call results in failure due to the endpoint being busy. The MiVoice Business must be programmed to allow the caller this privilege.

Note that this operation is invoked when the calling device is in the Call Failed state.

advTapCall	This operation connects a call that is waiting on an ACD path to another device, without removing the call from the ACD queue. This operation can be used to have the caller listen to messages and/or interact with an IVR while waiting in the queue. If the caller reaches the head of the queue during this activity, the “tap” call is aborted and the caller is connected to the available agent. Tap call destinations can be softphone ports on a MiVoice Business.
advSetMsgWaiting	This operation is used to set the state of a phone Message Waiting lamp / indicator. The application can turn on and turn off the indication. The application can use the getDeviceFeatures operation to confirm the status of the message waiting lamp / indicator.
advSetACD2AgentPresenceAllGroups	This operation allows an application to set an ACD agent absent or present in all ACD groups on a specific MiVoice Business. The agent has to be configured in the ACD groups.
advSetACDAgentPresence	This operation allows an application to set an ACD group member (agent) absent or present in a particular ACD group.
advLoginACDAgent	This operation allows an application to log in an agent using a MiVoice Business phone enabled for such behavior. There are two types of Agents supported with this operation; 1) Traditional Agent and 2) Internal Hotdesk Agent. For traditional agent the phone remains in service and the Agent ID and the phone prime line DN can be called. For Internal Hotdesk Agent, the phone becomes out-of-service and the ACD agent becomes in-service. This operation requires an ACD agent DN and PIN.
advLoginExternalACDAgent	This operation allows an application to log in an External Hotdesk agent using a MiVoice Business EHDA that is configured to route calls to an external mobile device. This operation requires an ACD agent DN.
advLogoutACDAgent	This operation allows an application to log out a traditional or internal hotdesk agent.
advACDSilentMonitor	This operation allows an ACD supervisor to silent monitor an ACD agent call depending on the options enabled in the MiVoice Business. This operation has changed with MiVoice Business 7.0. In MiVoice Business 6.0 a CallConferencedEvent is generated to each of the three monitored parties (agent, caller, and supervisor) involved. Conference member list is presented to the application. The silently monitored ACD agent phone will be the last item in the conference member list. The supervisor must be on the same MiVoice Business as the ACD agent. In MiVoice Business 7.0 several bridge events are generated when a supervisor creates a silent monitor on an ACD Hot Desk Agent; the bridge events are listed in this document and include bridge connected, bridge disconnected, bridge join, and bridge leave.
advACDRequestHelp	This operation allows an ACD agent to request help from an ACD supervisor (i.e., listen-in on his active ACD call). This operation is equivalent to that available via the Request Help softkey on a MiVoice Business phone. An application monitoring an agent cannot cancel an agent request to have a supervisor listen-in (the

	agent must press the cancel button on his phone).																						
advSetACDAgentMakeBusy	This operation allows an application to set an agent status to busy.																						
advLogoutExternalACDAgent	This operation allows an application to log out an external hotdesk agent.																						
advSetAutoAnswer	This operation allows an application to enable auto answer on a phone.																						
advCancelACDWorkTimer	This operation allows an application to cancel a work timer for an ACD agent.																						
advCancelACDResezeWorkTimer	An application can Cancel Reseze Timer using similar to Cancel Work Timer. A device monitor is required on an EHDA to Cancel Reseze Timer. An Application cannot START a reseize timer or a work timer. An application can only cancel a timer.																						
advSetTrunkMakeBusy	This operation allows an application to set a TDM trunk status to busy. Available modes are MakebusyForced, MakebusyCourtesy, and CancelFeature.																						
advMonitorAllTrunkFeatures advStopMonitorAllTrunkFeatures	This pair of operations is used to start and stop a class level monitor on the trunks for a specific MiVoice Business.																						
advMonitorACD2AgentFeatures advStopMonitorACD2AgentFeatures	This pair of operations is used to start and stop a class level monitor on the ACD2 agents for a specific MiVoice Business.																						
advMonitorACDExpress advStopMonitorACDExpressObjects	This pair of operations is used to start and stop a class level monitor on the ACD Express agents for a specific MiVoice Business.																						
advGetACDGroupStatus	<p>This operation retrieves the number of the agents in an ACD group, number of the logged in agents in an ACD group, the number of calls waiting on an ACD group, the number of the dnd agents in an ACD group, the number of the idle agents in an ACD group, the number of ACD agents that have logged in and are present.</p> <p>Note: When an Agent is in MakeBusy state and not available to receive calls, such status is not provided. A future release of Mitel OIG will fix this issue.</p>																						
advGetACD2GroupDescription	<p>These operations each gets a description of a specific object where the description includes:</p> <table> <tr> <td>boolean</td><td>result;</td></tr> <tr> <td>String</td><td>errorDescription;</td></tr> <tr> <td>String</td><td>group_number;</td></tr> <tr> <td>String</td><td>group_name;</td></tr> <tr> <td>int</td><td>number_of_members;</td></tr> <tr> <td>int</td><td>uses_skills;</td></tr> <tr> <td>boolean</td><td>isResilient ;</td></tr> <tr> <td>String</td><td>primaryIcpAddress;</td></tr> <tr> <td>long</td><td>primaryIcpId;</td></tr> <tr> <td>String</td><td>secondaryIcpAddress;</td></tr> <tr> <td>long</td><td>secondaryIcpId;</td></tr> </table>	boolean	result;	String	errorDescription;	String	group_number;	String	group_name;	int	number_of_members;	int	uses_skills;	boolean	isResilient ;	String	primaryIcpAddress;	long	primaryIcpId;	String	secondaryIcpAddress;	long	secondaryIcpId;
boolean	result;																						
String	errorDescription;																						
String	group_number;																						
String	group_name;																						
int	number_of_members;																						
int	uses_skills;																						
boolean	isResilient ;																						
String	primaryIcpAddress;																						
long	primaryIcpId;																						
String	secondaryIcpAddress;																						
long	secondaryIcpId;																						

advGetACD2PathDescription	<p>These operations each gets a description of a specific object where the description includes:</p> <ul style="list-style-type: none">boolean result;String errorDescription;String pathName;int numGroups ;String firstRad ;String secondRad ;String thirdRad ;String fourthRad ;String interflowDn ;boolean interflowEnabled ;String pathUnavailableDn ;boolean isResilient = false;String primaryIcpAddress;long primaryIcpId;String secondaryIcpAddress;long secondaryIcpId;
advGetACDAgentDescription	<p>These operations each gets a description of a specific object where the description includes:</p> <ul style="list-style-type: none">boolean result;String errorDescription;String agentId;String agentName;String agentDn;int agentSkill;boolean isResilient;String primaryIcpAddress;long primaryIcpId;String secondaryIcpAddress;long secondaryIcpId;AcdaAgentType agentType;
advGetACDXGroupDescription	<p>This operation is not supported. If used returns invalid device error.</p>
advGetHuntGroupDescription	<p>These operations each gets a description of a specific object where the description includes:</p> <ul style="list-style-type: none">boolean result;String errorDescription;String groupNumber;String groupName;String firstRad;String secondRad;String nightRad;int numberOfMembers;HuntGroupMode huntMode;HuntGroupType huntType;boolean isResilient;

String	primaryLcpAddress;
long	primaryLcpId;
String	secondaryLcpAddress;
long	secondaryLcpId;

advGetRingGroupDescription

These operations each gets a description of a specific object where the description includes:

boolean	result;
String	errorDescription;
String	groupNumber;
String	groupName;
String	overflowPoint;
int	numberOfMembers;
RingType	ringType;
boolean	isResilient = false;
String	primaryLcpAddress;
long	primaryLcpId;
String	secondaryLcpAddress;
long	secondaryLcpId;

advGetSipPeerDescription

These operations each gets a description of a specific object where the description includes:

boolean	result;
String	errorDescription;
PeerType	peerType;
PeerTrunkType	peerTrunkType;
boolean	isResilient;
String	primaryLcpAddress;
long	primaryLcpId;
String	secondaryLcpAddress;
long	secondaryLcpId;

AdvCancelTrunkMakeBusy

This operation allows an application to cancel make busy on a specific trunk number.

AdvGetACDAgentStatus

This operation allows an application to retrieve details about a specific ACD agent.

AdvGetSipPeerStatus

This operation allows an application to determine if a specific call through a SIP trunk is still connected.

AdvGetTdmTrunkStatus

This operation allows an application to determine the make busy state of a TDM trunk.

AdvStopMonitorConferences	This operation allows an application to stop a MiVoice Business level monitor on all conferences created on the MiVoice Business.
advGetSystemConferences	This operation is used to trigger the MiVoice Business to generate conference events for any active conferences in the MiVoice Business
advCoachCall	This operation is used by ACD Supervisors to coach an agent on a call. To use this operation the supervisor must first set a silent monitor on an agent.
advMonitorAllBridges	This operation is used by applications to monitor all bridges on a MiVB node for bridge feature events.
advStopMonitorAllBridges	This operation is used by applications to stop monitoring all bridges on a MiVoice Business for bridge feature events.

Call Control Service Operations – Standard Type

This section defines the Mitel OIG Call Control Service operations for the Standard type.

alternateCall

Definition

alternateCall (sessionId, localCallId, objectId)

Description

This operation 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 Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the active call that will be placed on consultation hold.
objectId	The monitor object ID for the object has an active call and a call on consultation hold.

Returns

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

Notes

- The application must **have** a monitor on an object that has an active call and a call on consultation hold.

*answerCall**Definition*

answerCall (sessionId, localCallId, objectId)

Description

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

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the active call to be answered.
objectId	The monitor object ID for the object being called.

Returns

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

Notes

- The application must have a monitor on the objectId.
- 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.
- The answerCall operation requires a valid localCallId (i.e., local call ID represents the active call to be answered). If an invalid local call id is used, the Mitel 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. The device used to place a campon will ring when the camped on device goes idle.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the call to be camped on.
objectId	The monitor object ID for the object trying to camp on.

Returns

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

Notes

- The application must have a monitor on the objectId being used to initiate a campon.
- camponCall fails with error "Feature not allowed" when invoked by application on a MiVoice Business SIP device. This operation is not supported on SIP devices.

cancelConsCall*Definition*

cancelConsCall (sessionId, localCallId, objectId)

Description

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

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the call to be cancelled.
objectId	The monitor object ID for the object doing the cancel.

Returns

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

Notes

- The application must have a monitor on the objectId used in the operation.
- An example on the use 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 / disconnects 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 object.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. localCallId is for the call to be cleared.
objectId	The monitor object ID for the object clearing the call.

Returns

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

Notes

- The application must have a monitor on the objectId used in the operation.
- If the monitored object is a phone prime line in hands-free or headset mode, then the line is left in IdleState. If the line is off-hook, the line is left in dialing state. The event received at the application is different when the call is cleared and line goes to idle compared to when the call is cleared and the line goes to dialing state.
- Calls on hard hold cannot be cleared.
- The cause provided in the call cleared event indicates which object cleared the call; callCleared the other connected object cleared, callClearedInvoked this end cleared the call.

clearCallMeBackMsg

Definition

clearCallMeBackMsg (sessionId, objIdOfDeviceToCall, deviceDn)

Description

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

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objIdOfDeviceToCall	The monitor object ID for the object to call back.
deviceDn	The phone number for object where you want to clear a message.

Returns

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

Notes

Steps to reproduce an error condition on a normal device:

1. Use Application and log into the Mitel 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 the call fails.
4. Invoke CC Operation SendCallMeBackMsgForCall() from Phone2.
5. Verify messagewaitingLamp = true by invoking GetDeviceFeature() on Phone1.
6. Invoke clearCallMeBackMsg on phone1:
 - Expected result on HCIRoute device:
clearCallMeBackMsg is successful
 - Expected result on regular phone:
clearCallMeBackMsg fails with error: UNSUPPORTED_BY_PBX

conferenceCall

Definition

conferenceCall (sessionId, localCallId, objectId)

Description

This operation merges two calls at an object (an active call and a call on consultation hold) into a conference call. This operation can be used to add more members to an existing conference call in the same manner used to setup the initial conference. This operation allows up to 7 conference members plus the initiator.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the call to be conferenced.
objectId	The monitor object ID for the object doing the conference.

Returns

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

Notes

1. The application must have a monitor on the objectId used in the operation.

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 phone number. The MiVoice Business only supports a two step process for doing a transfer or a conference. In either case the application first uses this operation to put an active call on consultation hold and make a call to another endpoint. Then the application has many options; create a conference, do a transfer, cancel the consultation call, alternate between calls, etc.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the call to be conferenced.
objectId	The monitor object ID for the object doing the conference.
Number	The phone number of the other endpoint to be called.

Returns

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

Notes

1. The application must have a monitor on the objectId used in the operation.
2. No other parties can already be held on consultation hold by the device.
3. The party placed on hold receives music if the MiVoice Business is so configured.
4. 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.

getLcpConfig

Definition

getLcpConfig(long sessionId, long icpId)

Description

This operation is used to retrieve the E.164 settings configured in the Mitel OIG. This operation can be used after the application has successfully logged in to the Mitel OIG using standard call control service.

Returns

- result - true or false
- errorDescription - if result is false
- maxExtLength - number of digits
- trunkPrefix - leading trunk digits configured in MiVB controller.
- longDistancePrefix - number of digits
- internationalPrefix - number of digits
- localAreaCodes - area codes separated by comma
- outsidePrefix - digits used to select outside line for making a call.

- localCountryCode - number of digits

Notes

None

getCallStatus

Definition

getCallStatus (sessionId, objectId)

Description

This operation allows the application to request the call state of a monitored object that can be in a call. Although the response to this operation is a generic call status message the application should only be looking for call state as other information is not always provided.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel 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 call state for the object
- UNKNOWN – if event type is not known

Notes

An application can be obtain call status on a line appearance, Personal Ring Group member, Ring Group, Hunt Group (Voice only), Multi Call Group and Keyline Group,.

This operation is not expected to return device information.

getDeviceConfigurationEx()

Definition

getDeviceConfigurationEx (sessionId, icpId, primeDn);

Description

This operation gets phone configuration information related to a phone prime line DN and a hotdesk User DN. For a phone that has line appearances, the response to this operation on the

phone prime line DN will provide how many lines are configured on the phone, the phone number for each line, the button number for each line and the objectId for each line.



Note: When using this operation, the application must indicate a specific MiVoice Business using an icpld. This is important when a MiVoice Business has resilient phones. For example, if the Mitel OIG cannot connect to a primary MiVoice Business where the phone is normally connected, the Mitel OIG does not automatically (on behave of the application) connect to the secondary MiVoice Business. In this case the application must provide the icpld for the associated secondary MiVoice Business when calling this operation in order to get device configuration information when the phone is on the secondary MiVoice Business.



Note: GetDeviceConfigurationEx() gives INVALID_DEVICE when the primeDn attribute provided is for an ACD2 Agent, a Traditional ACD2 agent or an ACD Express agent. This operation is not supported for ACD agents. For ACD agents an application must use advGetAcidAgentDescription.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
icpld	A MiVoice Business Object ID.
primeDn	The phone number of the object to get configuration.

Returns

- result – true or false
- errorDescription – if result false
- lineConfig – Includes phone number, objectId, and button number for each line
- primaryIcpAddress – primary MiVoice Business IP address
- secondaryIcpAddress- secondary MiVoice Business IP Address
- isResilient- true or false
- primaryIcpId- primary MiVoice Business Id
- secondaryIcpId – secondary MiVoice Business id

Notes

1. Only phone prime line DN is allowed in this operation (i.e., no line appearance DN).
2. Internal and External Hotdesk User DN is supported.
3. Internal and External Hotdesk User Agent DN is supported.
4. This operation replaces getDeviceConfiguration in Mitel OIG 1.0.
5. getDeviceConfiguration is now obsolete.

getDeviceFeatures()

Definition

getDeviceFeatures (sessionId, objectId);

Description

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

The following list describes the Mitel OIG response to this operation depending on what attributes are provided when calling the operation:

objectId provided is for External Hotdesk User DN – If External Hotdesk User is logged in, the Mitel OIG response returns external device phone number in registrationDn and the EH DU DN in hotDeskUserLoggedInDn. If the External Hotdesk User is not logged in, registrationDn and hotDeskUserLoggedInDn are blank.

objectId provided is for External Hotdesk Agent DN – If External Hotdesk Agent is logged in, the Mitel OIG response returns external device phone number in registrationDn and the EH DA DN in hotDeskUserLoggedInDn. If the External Hotdesk Agent is not logged in, registrationDn and hotDeskUserLoggedInDn are blank.

objectId provided is for Internal Hotdesk User DN – If Internal Hotdesk User is logged in, the Mitel OIG response returns phone number of phone where Internal Hotdesk User is logged. The phone number is returned in registrationDn and hotDeskUserLoggedInDn is blank. If the External Hotdesk User is not logged in, registrationDn and hotDeskUserLoggedInDn are blank.

objectId provided is for Internal Hotdesk Agent DN – If Internal Hotdesk Agent is logged in, the Mitel OIG response returns phone number of phone where Internal Hotdesk Agent is logged. The phone number is returned in registrationDn and hotDeskUserLoggedInDn is blank. If the External Hotdesk Agent is not logged in, registrationDn and hotDeskUserLoggedInDn are blank.

objectId provided is for Internal Hotdesk User enabled object – If hotdesk user is logged in, the Mitel OIG response returns an Internal Hotdesk User phone number (DN) for hotDeskUserLoggedInDn. If the Internal Hotdesk User DN is not logged in, hotDeskUserLoggedInDn is blank.

objectId provided is for Internal Hotdesk Agent enabled object – If hotdesk agent is logged in, the Mitel OIG response returns an Internal Hotdesk Agent phone number (DN) for hotDeskUserLoggedInDn. If the Internal Hotdesk Agent DN is not logged in, hotDeskUserLoggedInDn is blank.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the object or hot desk DN of interest.

Returns

- result – true or false
- errorDescription – if result false
- hotDeskUserLoggedInDn – hotdesk DN
- registrationDn – phone number of phone where hotdesk DN is logged in
- CFNAExt forwarding DN and fwdOn Boolean value
- CFNAInt forwarding DN and fwdOn Boolean value
- CFBusyExt forwarding DN and fwdOn Boolean value
- CFBusyInt forwarding DN and fwdOn Boolean value
- CFAlways forwarding DN and fwdOn Boolean value
- DND (Boolean value)
- autoAnswer (Boolean value)
- msgWaitingLamp (Boolean value)

Notes

When the response to this operation includes attributes that have no value, then the information is not provided by the MiVoice Business.

getDeviceMembers*Definition*

getDeviceMembers (sessionId, icplD, primeDn)

Description

The “getDeviceMembers” operation can be used to determine if there are line appearances of a Prime DN on other telephones in the system.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
icplD	A MiVoice Business Object ID.
primeDn	The phone number of the object to get line appearances.

Returns

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

Notes

Only phone prime line DN is allowed in this operation (i.e., no line appearance DN).

getEvent

Definition

getEvent (sessionId, timeout)

Description

This operation is used by the application to synchronously call for events held in the Mitel OIG. This operation is only used if the application polls for events. The registerEventHandler operation must be used when the application wants to receive events asynchronously from the Mitel OIG. This getEvent operation returns immediately if an event is pending in the Mitel OIG. The application must implement the getEvent operation in a way that allows the application to collect events as fast as possible. After retrieving an event from the Mitel OIG, the application should simply store the event for later processing and immediately ask for another event. The Mitel 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 Mitel OIG upon successful login.
timeout	The period of time the application wants to block while waiting for an event to be returned by the Mitel 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

1. 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).
2. Only one event is returned at a time.

getHotDeskUserLoginDevice()

Definition

getHotDeskUserLoginDevice (sessionId, hotDeskUserobjectId);

Description

This operation responds with hotDeskDeviceDn (i.e., the phone number where the hotdesk user is logged in; hotdesk enabled object) based on hotdesk user DN Object ID. If the hotdesk user DN is logged out, the hotDeskDeviceDn will be blank.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel 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 number of object where hotdesk user is logged in. If hotdesk user not logged in this value is blank.
- hotDeskUserobjectId – if true Monitor object ID of the hotdesk user DN

Notes

The Mitel OIG provides no hotDeskDeviceDn value if the hotdesk user is not logged in.

getHotDeskUserDn()

Definition

(sessionId, hotDeskUserobjectId);

Description

This operation gets a userDn (i.e., the phone number of a logged hotdesk user) based on an objectId for a hotdesk enabled object. If there is no hotdesk user logged in to the object identified by the hotDeskUserobjectId, the userDn in the response will be blank.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
hotDeskDeviceObjectId	Monitor object ID of the object DN that is hotdesk enabled.

Returns

- result – true or false
- errorDescription – if result false
- userDn – hotdesk DN
- hotDeskDeviceObjectId - Monitor object ID of the object DN that is hotdesk enabled..

Notes

None.

getIcpld

Definition

getIcpld (sessionId, IcplAddress)

Description

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

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

Steps to reproduce error condition with invalid IP address:

1. Invoke GetIcpld() with invalid MiVoice Business IP Address (say 10.112.60.25 on which there is no MiVoice Business configured)

GetIcpld() returns errorDescription error: "Failed to open connection to ICP at: 10.112.60.25"

An error condition also occurs when a MiVoice Business is in an unstable/usable state.

Steps to reproduce error condition when MiVoice Business is not ready but connected to IP network:

1. Invoke GetIcpld()

GetIcpld() 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 MiVoice Business IP Address used by application.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
IcplAddress	The IP address of the MiVoice Business that the application wants to communicate with.

Returns

- result – true or false
- errorDescription – if result false
- Icpld - if result true
- connectedState – indicates if Mitel OIG is actually connected to the MiVoice Business.

Notes

Icpld identifies a specific MiVoice Business instance.

getLineAppearancelId

Definition

getLineAppearancelId (sessionId, lcplId, primeDn, buttonNum)

Description

This operation provides the object ID for a MiVoice Business phone line appearance. The application uses this line appearance object ID in other operations to specifically identify a MiVoice Business phone line appearance to the Mitel 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 Mitel OIG upon successful login.
lcplId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getlcplId 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 Mitel OIG uses the button number as a group member number in requests to the MiVoice Business.

getPhoneNumberId

Definition

getPhoneNumberId (sessionId, lcplId, primeDn)

Description

This operation provides the object ID for the prime line of a MiVoice Business phone. The application uses this phone number object ID in other operations to specifically identify a

MiVoice Business phone prime line DN to the Mitel OIG. An application must call this operation before setting a monitor on the phone prime line.



Note: This operation can be applied to a MiVoice Business hot desk user DN but not applied to a MiVoice Business hot desk agent DN. Monitoring of hot desk agents is only supported using operations in Advanced Call Control Service.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
lcplId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getLcplId operation.
primeDn	The prime line number of a phone on a MiVoice Business.

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. When an application wants to monitor phone line appearances, an application needs to request an object ID for each specific line and then set a monitor on the line. See getLineAppearanceld operation description below.

holdCall

Definition

holdCall (sessionId, localCallId, objectId)

Description

This operation places an the active call at the specified object on hard hold.

An example of event flow goes as follows:

1. Establish a two-party call between Phone1 and Phone2
2. Invoke 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 Mitel OIG upon successful login.
localCallId	Local call ID was provided in the last call status event (from the Mitel OIG) to the monitored objectId. The local call ID represents the call to be held.
objectId	The monitor object ID for the object involved.

Returns

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

Notes

1. The application must have a monitor on the objectId used in the operation.

The held call can be subsequently retrieved using the retrieveCall operation.

loginExtHotDeskUser()

Definition

loginExtHotDeskUser (sessionId, pin, objectId);

Description

This operation logs in an External hotdesk user. The external device used by the external user is represented within the MiVoice Business as an external hotdesk user (EHDU) DN. When the EHDU DN is called using an internal MiVoice Business phone, the EHDU DN is first connected and then the call is directed out a trunk to the external device.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
Pin	The security PIN for the hotdesk login.
objectId	Monitor object ID of the hot desk user.

Returns

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

Notes

None

loginHotDeskUser()

Definition

loginHotDeskUser (sessionId, pin, hotDeskDn, objectId);

Description

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

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
Pin	The security PIN for the hotdesk login
hotDeskDn	The phone number of the hotdesk user.
objectId	Monitor object ID of the object to 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.

logoutExtHotDeskUser()

Definition

logoutExtHotDeskUser (sessionId, objectId);

Description

This operation logs out an External hotdesk user.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the hotdesk user.

Returns

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

Notes

None.

logoutHotDeskUser()

Definition

logoutHotDeskUser (sessionId, objectId);

Description

This operation logs out a hotdesk user and places the object that is hotdesk enabled into in-service. The Hotdesk user DN goes out-of-service.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the hotdesk user DN.

Returns

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

Notes

None

makeCall

Definition

makeCall (sessionId, objectId, number, accountCode)

Description

This operation makes a call from a monitored object ID to the phone number provided. An application can also set a forced account code (optional); see the notes below for details.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object involved.
number	The phone number of the endpoint to be called.
accountCode	A number terminated by # (e.g., 555#)

Returns

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

Notes

- The application must have a monitor on the object used in the operation.
- The monitored object must be either in IdleState (on hook) or OriginatedState (off hook, but not yet dialing a number).
- If in IdleState, the object is automatically placed in hands-free mode to originate the call.
- If the device is in OriginatedState, the number is called.
- The optional account code attribute is used for call charging purposes. The account code is used in the Call Detail Record (CDR/SMDR) output by the MiVoice Business at the end of the call.
- If the DN or the account code is invalid then an error is returned.
- An application can call a Traditional ACD agent using an agent ID. The delivered and received events both show the agent ID in the Dialed Digits field and show the DN of the phone where the agent is logged in, in the called party and calling party fields. After a call is answered or after a transfer, the agent ID is not reported. To correlate the agent ID and the events, the application needs to remember the relation between agent ID and agent DN in the delivered and received events.
- A forced non-verified account code may be specified with makeCall if required. The first makeCall operation specifying the phone number 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 #. The forced non-verified account code must be preceded with #. For example, makeCall with phone number 613-836-1234 in the number attribute, followed by makeCall with forced non-verified account code #456 in the number attribute.
- A forced verified account code may also be specified with makeCall, but the order of calls is reversed: the forced verified account code must come in the first makeCall and the phone number to be called must come in the second makeCall.
- Non-forced account codes with makeCall are not supported.

- When not using account codes, ensure accountCode attribute is null and not blank (i.e., passing account code field as ""). Providing a blank accountCode attribute in this operation will prevent the MiVoice Business from sending dialed digits in call status events.

monitorFeatures

Definition

monitorFeatures (sessionId, objectId)

Description

This operation creates a device feature monitor. Device features include Do-Not-Disturb, call auto answer, call forwarding, hotdesk user login and logout. The provided object ID identifies the object to be monitored. This operation instructs the MiVoice Business to report feature events when a device feature changes. The object ID also identifies the MiVoice Business to use to set the feature monitor. If this operation is successful the application receives feature events for the monitored object. See feature events section below.



Note: monitorFeatures is independent of monitor object. If an application creates a device monitor on a device followed by a device feature monitor on the same device, closing the device monitor does not automatically close the device feature monitor. If an application tries to create a monitor on a device when the monitor already exists, the application will receive Error: Device is already monitored by session.



Note: When an application creates a device feature monitor on a MiVoice Business resilient phone, two feature events will be generated for any feature change on the resilient phone (one from primary MiVoice Business and one from secondary MiVoice Business). For example, if DND is enabled on a resilient phone then the application will receive two feature events for DO_NOT_DISTURB for the same phone; this is only applicable to Mitel OIG MiVoice Business IP Phone Resiliency support.



Note: Note: In case of resilient Hot Desk user and resilient Hot Desk ACD agents, an application will only receive one device feature event (i.e., HOT_DESK_FEATURE with ENABLED state) when the resilient Hot Desk user or resilient Hot Desk ACD agent logs in or logs out. This is different behavior compared to physical phone features like DND and Auto answer that apply to the phone itself. Log in and log out action affect the single physical phone. Device features like DND and auto answer get copied to the secondary MiVoice Business and thus two device feature event updates.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called a specific get object ID operation.

Returns

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

Notes

The lcpld is the object ID for the MiVoice Business that is controlling the object.

monitorObject

Definition

monitorObject (sessionId, objectId)

Description

This operation creates a monitor using an object ID. The object ID identifies a MiVoice Business object (e.g., phone prime line, hotdesk user, agent, group, path,) to be monitored. An application calls this operation for each object to be monitored, one object at a time. This operation generates a monitored set event that confirms the monitor has been set successfully on a MiVoice Business. The object ID also identifies the MiVoice Business 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. The application should also expect events from the Mitel OIG related to this monitor even if no operations are called.



Note: If the monitored object is resilient, the Mitel OIG will attempt to set monitors on both the primary and secondary MiVoice Business instances. Thus the application can receive two monitor set events; one from Primary MiVoice Business and one from the Secondary MiVoice Business.



Note: If an application tries to create a monitor on a device when the monitor already exists, the application will receive Error: Device is already monitored by session.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called a specific get object ID operation.

Returns

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

Notes

1. The lcpld is the object ID for the MiVoice Business that is controlling the object.
2. When an application is setting a monitor, the application must wait for a monitor set event before sending any commands to the MiVoice Business related to that specific monitor. The monitor set event is confirmation to the application that the monitor has been created.

monitorPRGPresence*Definition*

monitorPRGPresence (sessionId, lcpld)

Description

This operation establishes a class level monitor on a MiVoice Business. This monitor reports all presence activities of all Personal Ring Groups on the MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
lcpld	The monitor object ID for the MiVoice Business controlling one or more PRGs.

Returns

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

Notes

The MiVoice Business class level monitor for Personal Ring Groups can receive many presence events for members in different PRGs.

newCall*Definition*

newCall (sessionId, localCallId, objectId)

Description

This operation ends a consultation call without retrieving the held call, making it possible for the application to do another consultation call (i.e., the monitored object is placed in dialing state). An Application must use makeCall to call another phone number or the application can use cancel consultation call to retrieve the call on consultation hold.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object involved.
localCallId	localCallId in this case is related to the consultation call

Returns

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

Notes

1. The application must have a monitor on the object used in the operation.

*outPulseDigits**Definition*

outPulseDigits (sessionId, dtmfDigits, objectId)

Description

This operation sends DTMF digits from the monitored object out a PSTN trunk while a call is in EstablishedState. Tones are not heard at the monitored object. An application can not use this operation to instruct one IP Phone to send DTMF digits to another IP Phone. The monitored object in this operation is an IP phone line.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object involved.
dtmfDigits	DTMF digits to be sent. The receiving endpoint is responsible for turning the delivered DTMF digits into tones.

Returns

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

Notes

1. The application must have a monitor on the objectId used in the operation.
2. The monitored IP Phone line is required to be in Established State.
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 an IP Phone line to send DTMF digits to another IP Phone.

*pickupCall**Definition*

`pickupCall (sessionId, rind, objectId)`

Description

This operation allows the monitored objectId to pick up a call which is ringing at another object (ringDn). This operation cannot be used to pickup a call which has landed at a Routing Device; see the notes below for details.

The pickupCall operation requires the MiVoice Business to have a FAC for directed pickup or the pickupCall operation will fail with Invalid_Device. The MiVoice Business configuration can be confirmed by using an actual physical phone to do a call pickup. If the physical phone is able to pickup a call, the Mitel OIG operation should also work.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object involved.
ringDN	The phone number of the object that is ringing.

Returns

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

Notes

1. Use pickupCall when the ringDN is in Received State for a call.
2. 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.

*redirectCall**Definition*

`redirectCall (sessionId, redirectDn, localCallId, objectId)`

Description

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

This operation in case of keyline and multicall depends upon MiVoice Business COS option "Display Caller ID on Multicall/Keylines" being enabled / on. Redirection throws an error "An Invalid Call Control operation was attempted (Call Control command failed)" when COS option is disabled (i.e. 'display caller id on multicall keys' is set to no (default)).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object 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

- Use redirectCall if the call is in Received State or Queued State.
- Use transferCall if the call is in Established State or HeldState.
- If the call's new destination is busy redirectCall returns FEATURE_NOT_ALLOWED.
- If calls are made to the same phone on different line appearances, this operation always redirects the first call, no matter what localCallId is used.
- Upon redirection of an ACD call, the MiVoice Business no longer regards (or reports) that call as an ACD call. This is an inherent characteristic of MiVoice Business ACD packages.
- Applications can: 1) redirect to a MiVoice Business remote directory number from a MiVoice Business ringing object 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 MiVoice Business remote directory number.
- If the call is being redirected from an ACD queue then a redirectDn is required. If the redirect is made on an ACD2 path and no directory number is present, and a path Interflow DN is programmed and enabled, then this destination will be used.
- If the call is being redirected from a ringing phone and a redirectDn has not been provided then a directory number from the phone's call forwarding table is used. When the phone has more than one call forwarding number programmed, the MiVoice Business chooses one of the numbers based on its call forwarding priorities.

registerEventHandler

Definition

registerEventHandler (sessionId, eventHandlerURL)

Description

This operation registers an application event callback with the Mitel OIG. The Mitel OIG uses the URL provided in the operation to send events asynchronously to the application. The application must process the https request from the Mitel 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 the 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. Refer to the Mitel OIG 2.1 Sample Application Package for sample code and documentation.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
eventHandlerURL	The URL used by the Mitel OIG to send events back to the application in an asynchronous manner.

Returns

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

Notes

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

remoteRetrieveCall

Definition

remoteRetrieveCall (sessionId, objectId, remoteDn)

Description

The remoteRetrieveCall operation is used to retrieve a call put on hold by one telephone to a different telephone.

Prerequisites to using this operation:

The COS option “Call Hold Remote Retrieve” must be set to “yes” and the feature access code “Call Hold – Remote Retrieve” must be set

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID of the object to which the call is to be retrieved.
remotetDN	The directory number of the device from which the call is to be retrieved.

Returns

result – true or false
errorDescription – if result false

*Notes**retrieveCall**Definition*

retrieveCall (sessionId, localCallId, objectId)

Description

This operation reconnects an existing hard held call at the specified holding object (i.e., monitored object). The call was previously held using the holdCall operation or the MiVoice Business call hold feature. This operation retrieves a call from hard hold. The application must provide a local Call ID zero (0) to retrieve the hard hold. The local Call ID is related to the device trying to retrieve the held call and not the local Call ID of the call on the device being held.

Example scenario is as follows:

MiVoice Business: MiVoice Business 1

- Devices: Device A, Device B, Device C and Device D on MiVoice Business 1 (all generic SIP Devices)

Steps:

1. Invoke CC operation Makecall() on Device A to B.
2. Press Speaker Key on Device A.
3. Press Speaker Key on Device B.
4. Invoke CC operation Consultationcall() on Device B to C.
5. Press Speaker Key on Device C.
6. Invoke CC operation Conferencecall() on Device B.
7. Invoke CC operation Consultationcall() on Device C to D.
8. Press Speaker Key on Device D.
9. Invoke CC operation Conferencecall() on Device D.
10. Invoke CC operation HoldCall() on Device D.
11. Invoke CC operation RetrieveCall() on Device D.

12. Invoke CC operation Clear call() on Device A.

13. Invoke CC operation Clear call() on Device B.

14. Invoke CC operation Clear call() on Device C.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object with a held call.
localCallId	The identity of the call to be retrieved.

Returns

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

Notes

None.

sendCallMeBackMsgForCall

Definition

sendCallMeBackMsgForCall (sessionId, localCallId, objectId)

Description

This operation allows a calling monitored object to leave a message at the called endpoint indicating who called. The called endpoint could have been busy at the time or unable to answer.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object leaving the message.
localCallId	The identity of the call to the called endpoint.

Returns

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

Notes

1. This operation is not supported over SIP or IP trunks on MiVoice Business instances.
2. This operation is not supported when leaving call back messages on SIP devices.

sendCallMeBackMsgNoCall

Definition

sendCallMeBackMsgNoCall (sessionId, objIdOfDeviceToCall, deviceDn)

Description

This operation allows a monitored object to set a call me back message at a device identified by deviceDn. SendCallMeBackMsgNoCall can only be invoked from a routing device (that is, only a routing device can be referred to by attribute objIdOfDeviceToCall).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objIdOfDeviceToCall	The monitor object ID for the object to call back.
deviceDn	The phone number for object where you want to leave a message.

Returns

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

Notes

1. This operation is not supported over SIP or IP trunks on MiVoice Business instances.
2. This operation is not supported when leaving call back messages on SIP devices.
3. Mitel OIG SendCallMeBackMsgNoCall fails with error UNSUPPORTED_BY_PBX when Device C is not a routing device.

Steps to reproduce an error condition:

- a. Make Call from Device A to Device B.
- b. Answer Call from Device B.
- c. Make Call from Device C to Device B (call should fail as the destination Device is busy).
- d. Set Call_Me_Back_No_Call on Device B from Device C.
- e. Clear Call from Device A.
- f. After setting Call_Me_Back_No_Call on Device B from Device C,

Mitel OIG gives an error UNSUPPORTED_BY_PBX.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 object. 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., MiVoice Business does not check the account code).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object with call.
accCode	The account code to be assigned to 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 object must be in EstablishedState when this operation is called. An account code is a string of 2 to 12 alphanumeric characters. setAccountCode will return INVALID_ATTRIBUTE_VALUE if this format is not correct.

setCallMeBack

Definition

setCallMeBack (sessionId, localCallId, objectId)

Description

This operation allows the caller to have the MiVoice Business 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 Mitel 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

1. This operation is not supported over SIP or IP trunks on MiVoice Business instances.
2. This operation is not supported when leaving call back messages on SIP devices.

*setCFNAExternal()**Definition*

setCFNAExternal (sessionId, objectId, dn, cfOn);

Description

This operation sets the call forwarding for an object on or off related to an external call. If the called object 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 Mitel OIG upon successful login.
objectId	Monitor object ID of the object to change forwarding.
Dn	The phone number to forward call to
cfOn	True is On or false is Off

Returns

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

Notes

None.

*setCFNAInternal()**Definition*

setCFNAInternal (sessionId, objectId, dn, cfOn);

Description

This operation sets the call forwarding for an object on or off related to an internal call. If the called object 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 Mitel OIG upon successful login.

objectId	Monitor object ID of the object to change forwarding.
dn	The DN to forward to
cfOn	True is On or false is Off

Returns

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

Notes

None.

setCFBusyExternal()

Definition

setCFBusyExternal (sessionId, objectId, dn, cfOn);

Description

This operation changes the call forwarding for an object on or off related to an external call. If the object 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 Mitel OIG upon successful login.
objectId	Monitor object ID of the object to change forwarding.
dn	The DN to forward to
cfOn	True is On or false is Off

Returns

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

Notes

None.

setCFBusyInternal()

Definition

setCFBusyInternal (sessionId, objectId, dn, cfOn);

Description

This operation changes the call forwarding for an object on or off related to an internal call. If the object 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 Mitel OIG upon successful login.
objectId	Monitor object ID of the object to change forwarding.
Dn	The DN to forward to
cfOn	True is On or false is Off

Returns

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

Notes

None.

setCFAlways()

Definition

setCFAlways (sessionId, objectId, dn, state);

Description

This operation changes the call forwarding always of the object on or off.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the object to change forwarding.
dn	The DN to forward to
cfOn	True is On or false is Off

Returns

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

Notes

None.

setDeviceDND()

Definition

setDeviceDND (sessionId, objectId,dndState);

Description

This operation sets or clears the Do-Not-Disturb state an object. When making a call to a phone in DND, the phone is considered busy.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the object to be controlled.
dndState	Indicates that DND is to be set or cleared

Returns

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

Notes

None.

setPrgPresence()

Definition

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

Description

This operation sets the presence of a PRG member to be presence in the group or not presence in the group. The presence status of a member in a PRG can be set by a supervisor (use invokerDn), another member(use invokerDn), and the member itself (no need for invokerDn).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the PRG member.
PrgDn	Personal Ring Group DN
presenceValue	Sets the member's presence to be in or out of PRG (true is in or false is out)
invokerDn	invokerDn allows a supervisor to change presence of a PRG member

Returns

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

Notes

To change presence with supervisor, supervisor needs to be programmed on same MiVoice Business as the PRG member.

*splitConferenceCall()**Definition*

splitConferenceCall (sessionId, objectId);

Description

This operation splits a conference call into two calls; one active call and one on consultation hold. The conference member placed on hold is usually the last member added to the conference but not always.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the conference member that is splitting the conference.

Returns

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

*stopFeatureMonitor**Definition*

stopFeatureMonitor (sessionId, objectId)

Description

This operation stops a specific device feature monitor.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called a specific get object ID operation.

Returns

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

Notes

Upon successful completion of this operation, the application will no longer receive device feature events for this monitor.

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 does not stop the device feature monitor.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called a specific get object ID operation.

Returns

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

Notes

Upon successful completion of this operation, the application can no longer use the monitor for sending commands and will no longer receive events related to the monitor.

stopPRGPresenceMonitor*Definition*

stopPRGPresenceMonitor (sessionId, lcpld)

Description

This operation stops a monitor on the MiVoice Business that reports all presence activities of all Personal Ring Groups.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
lcpld	The monitor object ID for the MiVoice Business controlling the PRGs.

Returns

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

Notes

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

*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 Mitel OIG upon successful login.
ObjectId	The monitor object ID for the object doing the trade operation.

Returns

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

Notes

This operation makes invoker talk with the queued party and the other party is placed consultation-hold rather than queued. Therefore, a second call to tradeCall will not work; 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 party ends its participation in the call.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	The monitor object ID for the object initiating the transfer.

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.
3. If the target party does not answer the unsupervised transfer, the requesting party will be recalled by the original party on consultation hold.

`verifyHotDeskUserPin()`*Definition*

```
verifyHotDeskUserPin (sessionId, pin, hotDeskDn, objectId);
```

Description

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

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
pin	The security PIN for the hotdesk login
hotDeskDn	The phone number of the hotdesk user.
objectId	Monitor object ID of the hotdesk user DN.

Returns

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

Notes

None.

Call Control Service Operations – Advanced Type

The sections below define the Mitel OIG Call Control Service operations for the Advanced type.

advACDSilentMonitor

Definition

advACDSilentMonitor (sessionId, objectId, localCallId, agentDn)

Description

This operation allows a supervisor to silent monitor an ACD agent call depending on the options enabled in the MiVB. There are three existing COS options that control the availability and use of the silent monitor feature.

- | | |
|-----------------------------------|----------|
| • ACD Silent Monitor Accept | Yes / No |
| • ACD Silent Monitor Allowed | Yes / No |
| • ACD Silent Monitor Notification | Yes / No |

The first option controls whether a user operating in the agent role will allow their calls to be silent monitored. The second option controls whether a user in the supervisor role is allowed to invoke the silent monitor feature. In order for a silent monitor to be allowed, the first option must be enabled in the agent's COS and the second option must be enabled in the supervisor's COS. The third option controls whether a user in an agent role is notified of a silent monitor. An agent with the 3rd COS option enabled receives a visual indication of the monitor on a MiNET display set during talk state. These are existing behaviors. For more information on the silent monitor feature, see the MiVoice Business system administrator help.

With Mitel OIG 2.1, a Mitel OIG application can use Mitel OIG 2.1 WSDL to communicate with both MiVoice Business 6.0 SP3 and MiVoice Business 7.0.

When communicating with MiVoice Business 6.0 SP3, this advACDSilentMonitor operation works as follows:

A CallConferencedEvent is generated to each of the three monitored parties (agent, caller, and supervisor) involved. Conference member list is presented to the application. The silently monitored ACD agent phone will be the last item in the conference member list. The supervisor must be on the same MiVoice Business as the ACD agent.

If more than two members and a supervisor are included in the conference, when someone other than the supervisor leaves the conference, a cause of `ConferenceMemberDropped` or `ACDSilentMonitorMemberDropped` is reported, irrespective of which party actually dropped. These causes are considered to be equivalent.

Note that there are restrictions on the support for Silent Monitor conference reporting:

- Silent Monitoring of a member in a two-party call is supported. Silent Monitoring of a member in a three or more party call is not supported. Events may occur, but their presence, content, and consistency with ordinary conference events cannot be guaranteed.
- When any member drops out of a Silent Monitored three-party conference, a `DeviceDroppedEvent` will be reported, with a cause of `ConferenceMemberDropped` or `ACDSilentMonitorMemberDropped`. These causes are equivalent and party list information must be used to determine whether a member or the supervisor dropped from the conference.
- Redundant (duplicate) events will occur when extra members join a Silent Monitored three-party conference.

When communicating with MiVoice Business 6.0 7.0, this `advACDSilentMonitor` operation works as follows:

A “BRIDGE_JOIN” event is generated when a phone or application sets a silent monitor on an ACD hot desk agent or telephone. This event will be returned to all advanced applications which either have set the “`advMonitorAllBridges`” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_LEAVE” event is generated when a phone or application clears a silent monitor on an ACD hot desk agent or telephone. This event will be returned to all advanced applications which either have set the “`advMonitorAllBridges`” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_CONNECTED” event is generated when the first application invokes the “Silent monitor” feature on an agent or telephone. This event will be returned to all advanced applications which either have set the “`advMonitorAllBridges`” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_DISCONNECTED” event is generated when all phones/supervisors clear their silent monitors on an agent or telephone. This event will be returned to all advanced applications which either have set the “`advMonitorAllBridges`” on the specified MiVoice Business or are monitoring the phone/agents device features.

The new Mitel OIG whisper coach feature availability is controlled by the first two MiVoice Business 7.0 COS options mentioned above. The first option must be enabled in the agent’s COS and the second option must be enabled in the supervisor’s COS.

Attributes

ATTRIBUTE	DESCRIPTION
<code>sessionId</code>	<code>sessionId</code> was provided by the Mitel OIG upon successful login.
<code>objectId</code>	Monitor object ID of the supervisor.
<code>localCallId</code>	This attribute is 0 unless the supervisor phone is in dialing state. When in dialing state this attribute relates to the local call ID for the supervisor

ATTRIBUTE	DESCRIPTION
	phone.
agentDn	Agent DN that will be silent monitored.

Returns

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

Notes

1. When using a monitor on a phone to invoke AdvACDSilentMonitor on an agent in a call, the localCallId required in the operation request can be two possible values: 1) If the supervisor phone is off hook, use the localCallId of the supervisor phone in dialing state, 2) provide a localCallId of 0. DO NOT use the localCallId of the Agent phone call. The localCallId of the agent phone call is only used when the Agent is requesting ACDHelp from a supervisor.
2. When the MiVoice Business COS option is enabled to allow a supervisor to silent monitor an agent call, the COS option change does not take effect until the ACD supervisor phone changes call state at least once (the physical phone goes off hook and then back on hook).
3. An ACD2 Hot Desk Agent phone must be monitored and generate at least one call status event before an application can invoke “advACDSilentMonitor” for the supervisor phone wanting to silent monitor. Otherwise the “advACDSilentMonitor” operation from the monitored supervisor phone will fail with the following error: PRIVILEGE_VIOLATION.

If a Mitel OIG application tries to create a silent monitor on an ACD hotdesk user for a 7th supervisor, the operation will be successful but the application will receive a call status event with an error code. The operation succeeds because the MiVoice Business then needs to process the command and determine how many supervisors are actually connected to the bridge created for silent monitoring.

For example: Silent Monitor operation triggered a call to the MiVoice Business which returns successful invoked call status event.

```
Received Event type: CALL_EVENT Call Event Type: CALL_ORIGINATED
--- localCallId: 16 ObjectID: 18
--- Call State: ORIGINATED Event Cause: NEW_CALL_INVOKED
--- Attribute Name: CURRENT_GLOBAL_CALL_ID Attribute Value: 7D32AE82AD2B00034E0C
--- Attribute Name: ICP_ID Attribute Value: 1
--- Attribute Name: TRUNK_CALL_ID Attribute Value: 7D32AE82AD2B00000010
--- Device Name: CALLING_DEVICE
----- Device Attribute Name: NUMBER Attribute Value: 2011
```

Then after the MiVoice Business tries to connect to the bridge with 6 members, the 7th will fail.

```
Received Event type: CALL_EVENT Call Event Type: CALL_FAILED
--- localCallId: 16 ObjectID: 18
--- Call State: FAILED Event Cause: ERROR_DETECTED
--- Attribute Name: CURRENT_GLOBAL_CALL_ID Attribute Value: 7D32AE82AD2B00034E0C
--- Attribute Name: ICP_ID Attribute Value: 1
--- Attribute Name: TRUNK_CALL_ID Attribute Value: 7D32AE82AD2B00000010
--- Device Name: CONTROLLER_DEVICE
```

----- Device Attribute Name: NUMBER Attribute Value: 2011

----- Device Attribute Name: DIALED_DIGITS Attribute Value: *112004#

advACDRequestHelp

Definition

advACDRequestHelp (sessionId, objectId, localCallId, supervisorDn)

Description

This operation asks an ACD supervisor to listen-in to an active ACD call with this ACD agent.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the agent.
localCallId	Call ID of the agent's active call.
supervisorDn	Phone number for the supervisor requested.

Returns

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

Notes

1. An application monitoring an ACD agent cannot cancel an agent request for supervisor to listen (agent must press cancel button on his phone).
2. When the MiVoice Business COS option is enabled to allow a supervisor to silent monitor an agent call, the COS option change does not take effect until the ACD supervisor phone changes call state at least once (the physical phone goes off hook and then back on hook).

advAssignCallerId

Definition

advAssignCallerId (sessionId, objectId, localCallId, newName)

Description

This operation is used to assign the identity of a caller (i.e.: name and number) to a desk phone. This operation is related to incoming or outgoing call from a trunk so that the new identity is displayed on a MiVoice Business phone. The trunk must be currently participating in the call, and the phone must be monitored by the application. After this operation the caller's identity will be associated with the specific trunk call ID and will be displayed on the MiVoice Business phone but not provided in the Mitel OIG call status events. The Trunk and MiVoice Business phone must be on the same MiVoice Business controller.

This operation 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 PSTN trunk. The IVR machine could for example ask the caller to enter their customer number through their DTMF keypad.

Not all trunk types will support the use of this function. See the Notes at the end of this section for details.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the object with active call from a trunk
localCallId	Local call id of the active call with the trunk
newName	Name or names to be applied to the phone display with active call from trunk

Returns

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

Notes

1. A monitor must have been created on the trunk and this operation applied to the trunk monitor.
2. The caller identity attributes describe the caller's phone number and name (last, middle, and first).
3. The operation can be performed for all trunks except SIP, IP and AUTOVON trunks. This operation can be applied to a monitor for a DPNSS trunk on the terminating MiVoice Business of a MSDN call (i.e. call terminates on a MiVoice Business where the device is also controlled). The operation is not allowed on a DPNSS trunk on an originating or pass through MiVoice Business.
4. IP consoles cannot display the assigned caller ID.
5. Trunks in the following call states are allowed: OriginatedState, DeliveredState, ReceivedState, EstablishedState, HeldState, and QueuedState. A MiVoice Business phone display is updated immediately if the trunk is in EstablishedState. For the other call states, a MiVoice Business phone display is updated the next time there is an event that causes a display update at the phone.
6. The newName can contain a maximum of 20 characters including the added space between the parts of the name (last, middle and first name).

advCancelACDResezeWorkTimer

Definition

advCancelACDResezeWorkTimer (sessionId, objectId)

Description

This operation allows an application to cancel an active External Hotdesk ACD agent reseize timer.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of the External Hotdesk ACD agent.

Returns

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

Notes

advCancelACDReseizeWorkTimer

advCancelACDWorkTimer

Definition

advCancelACDWorkTimer (sessionId, objectId)

Description

This operation allows an application to cancel an active ACD agent work timer.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of the Agent.

Returns

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

Notes

advCancelACDWorkTimer

advCancelTrunkMakeBusy

Definition

advCancelTrunkMakeBusy (sessionId, objectId)

Description

This operation is used to cancel a trunk busy operation, remove the make busy setting.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the trunk

Returns

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

Notes

advCancelTrunkMakeBusy

advCoachCall

Definition

advCoachCall (sessionId, objectId, localCallId, trueOn, coachType)

Description

This operation is used by ACD Supervisors to coach an agent on a call. To use this operation the supervisor must first set a silent monitor on an agent (be connected to a bridge in the MiVoice Business). This operation allows a supervisor to coach a specific ACD hot desk agent (coach on and off), and request coach all (on or off) so that all supervisors and the specific ACD hot desk agent can hear each other when in a phone call.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the object with active call from a trunk
localCallId	Local call id of the active call with the trunk
turnOn	Boolean to turn coaching on or off
coachType	Defines the type of coaching requested; coach all or coach

Returns

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

Notes

Before using this operation, a supervisor phone must be monitored and an application must have submitted the advSilentMonitor operation on an ACD2 Hot Desk Agent phone.

advGetACDAgentDescription

Definition

advGetACDAgentDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific ACD agent on a MiVoice Business.

When the Mitel OIG generates a response to a request for ACD Agent description, the results do not provide an Agent Id when the ACD Agent is a Hot Desk Agent. The Agent ID is only provided when the ACD Agent Description results are related to a MiVoice Business Traditional ACD Agent.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the ACD agent

Returns

- result – true or false
- errorDescription – if result false
- agentId – only for traditional ACD agent
- agentName
- agentDn – only for hotdesk agent (internal and external)
- agentType
- agentSkill
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business

Notes

1. Traditional ACD agent cannot be member of an ACD Express Group.
2. A MiVoice Business provides three types of Agents: 1) Traditional ACD2 Agent, 2) ACD2 Hotdesk Agent (internal and external), 3) ACD Express Agent.

3. When invoking `advGetACDAgentDescription`, the operation returns agent ID and agentName or agentDN and agentName, but not all three at once. AgentID only applies to traditional ACD2 agents. AgentID in this operation is not to be confused with the objectID returned for an Agent DN.

For example:

- To get an object ID for a Traditional ACD2 Agent, provide `getObjId` operation with the AgentID.
- To get an object ID for an ACD2 Hotdesk Agent (internal and external), provide `getObjId` operation with the Hotdesk Agent DN or EHDA DN.
- To get an object ID for an ACD Express Agent, provide `getObjId` operation with the Express Agent DN.

Now when invoking `advGetACDAgentDescription` with object ID from `getObjId` we get:

- For traditional ACD2 Agent, we get the Traditional ACD2 Agent ID (not object ID) and Agent Name (Agent DN is null).
- For ACD Express Agent, we get the Agent DN and Agent Name (AgentID is null).
- For Hotdesk ACD2 Agent, we get the Agent DN and Agent Name (AgentID is null).

`advGetACDAgentId`

Definition

`advGetACDAgentId` (sessionId, icpld, agentDn)

Description

This operation provides the object ID for an ACD agent on a MiVoice Business. The application uses this ACD agent object ID in other operations to specifically identify a MiVoice Business ACD agent to the Mitel OIG. An application must call this operation before setting a monitor on the ACD agent.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the <code>getIcpld</code> operation.
agentDn	agent DN programmed in the MiVoice Business.

Returns

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

Notes

1. Traditional ACD agent cannot be member of an ACD Express Group.

2. advGetACDAgentId

advGetACDAgentStatus

Definition

advGetACDAgentStatus (sessionId, objectId)

Description

This operation allows an application to query the status of an ACD agent.

For Hotdesk ACD agents, use the advGetACDAgentStatus operation on the Agent DN. For example:

- getIcpId on the MiVoice Business
- getObjectId on the hotdesk ACD Agent DN
- monitorObject - monitor the hotdesk ACD Agent DN
- advGetACDAgentStatus - on the hotdesk ACD Agent DN

For traditional ACD agents, use the advGetACDAgentStatus operation on the phone where the agent is logged in. For example:

- getIcpId on the MiVoice Business
- getObjectId on the ACD phone
- monitorObject - monitor the ACD phone
- advGetACDAgentStatus - on the ACD phone

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID for the Hotdesk ACD Agent or Object ID of phone where traditional ACD agent is logged in.

Returns

- result – true or false
- errorDescription – if result false
- agentDn (for Traditional ACD agent this is the ACD Agent ID)
- registrationDn (this is the DN of the phone where ACD agent is logged in)
- CFNAExt
- CFNAInt
- CFBusyExt
- CFBusyInt
- CFAlways

- acdAgentGroupPresence
- DND
- autoAnswer
- msgWaitingLamp

Notes

advGetACDAgentStatus

advGetACDGroupStatus

Definition

advGetACDGroupStatus (sessionId, objectId)

Description

This operation allows an application to request status information for an ACD2 group.

The advGetACDGroupStatus operation does not report the status of agent phones that are in MAKE BUSY state. Any phone in MAKE BUSY is reported as logged in, available, idle even though the agent phone is not able to take an ACD call.

When an agent is in MAKE BUSY (agent puts their phone in this state) it is not reflected in the advGetACDGroupStatus information. For example, one agent is logged in and has their phone in MAKE BUSY; the following is returned from advGetACDGroupStatus:

Calls Waiting: 0 members busy: 0 members idle: 1 members in DND: 0 members logged in: 1 members logged in Present: 1 number of members: 3

The above "members busy" field is a count of agents busy in a call not in make busy state. If the application is monitoring agents, the application will get an event indicating when the agent phone in MAKE BUSY.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the agent.

Returns

- result – true or false
- errorDescription – if result false
- numOfMembers
- membersLoggedIn
- membersBusy
- callsWaiting

- membersInDND
- membersIdle
- membersLoggedInPresent

Notes

1. This operation is only supported for ACD2 groups; no support for ACD Express.
2. advACDGroupStatus

advGetACD2GroupDescription

Definition

advGetACD2GroupDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific ACD2 group on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Monitor object ID for ACD2 Group

Returns

- result – true or false
- errorDescription – if result false
- groupNumber – if result true
- groupName – if result true
- numberOfMembers – if result true
- usesSkills – if result true
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business

Notes

advGetACD2GroupDescription

advGetACD2GroupId

Definition

advGetACD2GroupId (sessionId, icpld, acd2GroupDn)

Description

This operation provides the object ID for an ACD2 group on a MiVoice Business. The application uses this ACD2 group object ID in other operations to specifically identify a MiVoice Business ACD2 group to the Mitel OIG. An application must call this operation before setting a monitor on the ACD2 group.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.
acd2GroupDn	agent DN programmed in the MiVoice Business.

Returns

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

Notes

advGetACD2GroupId

advGetACD2PathDescription

Definition

advGetACD2PathDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific ACD2 path on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Monitor object ID for ACD2 path

Returns

- result – true or false
- errorDescription – if result false
- pathname – if result true
- firstRad – if result true
- secondRad – if result true
- thirdRad – if result true
- fourthRad – if result true
- interflowDn – if result true
- pathUnavailableDn – if result true
- numGroups – if result true
- interflowEnabled – if result true
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business
- Notes
- advGetACD2PathDescription

*advGetACD2PathId**Definition*

advGetACD2PathId (sessionId, icpId, pathDn)

Description

This operation provides the object ID for an ACD2 path on a MiVoice Business. The application uses this ACD2 path object ID in other operations to specifically identify a MiVoice Business ACD2 path to the Mitel OIG. An application must call this operation before setting a monitor on the ACD2 path.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpId operation.
pathDn	ACD2 path DN programmed in the MiVoice Business.

Returns

- result – true or false

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

Notes

advGetACD2PathId

advGetACDXGroupId

Definition

advGetACDXGroupId (sessionId, icpld, acdXGroupDn)

Description

This operation provides the object ID for an ACD Express group on a MiVoice Business. The application uses this ACD Express group object ID in other operations to specifically identify a MiVoice Business ACD Express group to the Mitel OIG. An application must call this operation before setting a monitor on the ACD Express group.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.
acdXGroupDn	ACD Express Group DN programmed in the MiVoice Business.

Returns

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

Notes

advGetACD2GroupId

advGetCallStatus

Definition

advGetCallStatus (sessionId, objectId)

Description

The “advGetCallStatus” operation is used by advanced call control service applications to determine the call state of an object. If the object is in a call, then the operation provides details about the call such as localCallId, globalCallId, and trunkCallId if the call is using a SIP trunk.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectID	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

If the DN passed in is not a prime DN an error will be returned INVALID_DN. If the prime DN has no members the members array will be null.

*advGetEvent**Definition*

advGetEvent (sessionId, timeout)

Description

This operation allows an application to poll events from the Mitel OIG. The timeout indicates how long the poll waits for an event to be provided from the Mitel OIG.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
timeout	Time to wait for event.

Returns

- result – true or false
- errorDescription – if result false
- advEventType with (UNKNOWN, CALL_EVENT, FEATURE_EVENT, ACD_EVENT, SYSTEM_EVENT)
- advCallEventMsg with (advCallEventType, eventCauseType, callState, advCallEventAttributeValue, featuresAllowed, advDevice, advPartyMember, accountCode, accountCodeType, localCallId, callEventTime, objectId)
- advFeatureEventMsg with (featureEventType, featureEventAttribute, advFeatureEventAttributeValue)
- advSystemEventMsg with (systemEventType, connectionState, icpld, time)

Notes

1. See the Mitel OIG 1.1 WSDL from what is returned by advGetEvent. Not all the information above is always returned; it depends on the type of event.
2. advGetEvent

advGetHuntGroupDescription*Definition*

advGetHuntGroupDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific Hunt group on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the Hunt group

Returns

- result – true or false
- errorDescription – if result false
- groupNumber - if result true
- groupName - if result true
- firstRad
- secondRad
- nightRad
- huntMode
- huntType
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business

Notes

advGetHuntGroupDescription

advGetHuntGroupId

Definition

advGetHuntGroupId (sessionId, icpId, huntGroupDn)

Description

This operation provides the object ID for a Hunt group on a MiVoice Business. The application uses this Hunt group object ID in other operations to specifically identify a MiVoice Business Hunt group to the Mitel OIG. An application must call this operation before setting a monitor on the Hunt group.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpId operation.
huntGroupDn	Hunt Group DN programmed in the MiVoice Business.

Returns

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

Notes

advGetHuntGroupId

advGetRingGroupDescription

Definition

advGetRingGroupDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific Ring group on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Monitor object ID of the Ring group

Returns

- result – true or false
- errorDescription – if result false
- groupNumber - if result true
- groupName - if result true
- overflowPoint - if result true
- ringType - if result true
- numberOfMembers- if result true
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business

Notes

advGetRingGroupDescription

advGetRingGroupId

Definition

advGetRingGroupId (sessionId, icpId, ringGroupDn)

Description

This operation provides the object ID for a Ring Group on a MiVoice Business. The application uses this Ring Group object ID in other operations to specifically identify a MiVoice Business Ring Group to the Mitel OIG. An application must call this operation before setting a monitor on the Ring group.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpId	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpId operation.
ringGroupDn	Ring Group DN programmed in the MiVoice Business.

Returns

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

Notes

advGetRingGroupId

advGetSipPeerDescription

Definition

advGetSipPeerDescription (sessionId, objectId)

Description

This operation allows the application to retrieve information about a specific SIP Peer (SIP Trunk) on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Monitor object ID for SIP Peer

Returns

- result – true or false
- errorDescription – if result false
- peerType – if result true
- peerTrunkType – if result true
- primaryIcpAddress – IP Address of the Primary MiVoice Business
- secondaryIcpAddress – IP Address of the Secondary MiVoice Business
- isResilient – Boolean on the Resiliency status of the Object
- primaryIcpId – ICP ID of the Primary MiVoice Business
- secondaryIcpId – ICP ID of the Secondary MiVoice Business

Notes

advGetSipPeerDescription

advGetSipPeerId

Definition

advGetSipPeerId (sessionId, icpId, sipPeerName)

Description

This operation provides the object ID for a SIP Peer (SIP trunk) on a MiVoice Business. The application uses this SIP Peer object ID in other operations to specifically identify a MiVoice

Business SIP Peer to the Mitel OIG. An application must call this operation before setting a monitor on the SIP Peer.

The Mitel OIG supports monitoring of MiVoice Business SIP trunks (referred to as SIP Peer monitoring). A MiVoice Business Network Element name associated with a SIP trunk is needed to get an objectId before creating a monitor. All SIP trunk calls for the Network Element will be reported as call status events to the Peer monitor.

The following new error codes may be returned when creating a SIP Peer monitor:

ERROR	DESCRIPTION
INVALID_PEER	The peer name specified does not exist in Network Element form within the MiVoice Business.
UNSUPPORTED_PEER	The peer name specified exists in Network Element form but is not associated with a SIP trunk.

A get SIP Peer status operation is also provided to get the call state of a single call associated with a SIP Peer monitor. The status operation can only be used after a call status event is returned to the Peer monitor for a call; the get status operation requires both a local call ID and a global call ID. The possible call states include: DeliveredState, EstablishedState, FailedState, HeldState, IdleState, OriginatedState, OutOfServiceState, QueuedState, ReceivedState and UnavailableState.

The call status events generated for a SIP Peer monitor are the same as a TDM trunk monitor. A Peer monitor will also generate service state events. A Peer monitor event can be generated due to a change in the state of the underlying trunk (call state change) or for the peer itself (service state change) when it goes in or out of service.

A MonitorSetEvent is sent when a monitor is initially set on a SIP Peer.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.
sipPeerName	Network Element Name programmed in the MiVoice Business for the SIP Peer (trunk) to be monitored.

Returns

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

Notes

1. SIP trunks are not resilient.
2. advGetSipPeerId

*advGetSipPeerStatus**Definition*

advGetSipPeerStatus (sessionId, objectId, localCallId, globalCallId)

Description

This operation allows an application to determine the call state of a specific call within SIP Peer on a MiVoice Business using the local and global call ids of the call. The application needs to be monitoring the SIP Peer (trunk). A call status event must have been provided to the application containing the local and global call ids of the call; the application is checking if the call is still in connected state.

This operation is provided to get the call state of a single call associated with a SIP Peer monitor. The status operation can only be used after a call status event is returned to the Peer monitor for a call; the get status operation requires both a local call ID and a global call ID. The possible call states include:

- DeliveredState
- EstablishedState
- FailedState
- HeldState
- IdleState
- OriginatedState
- OutOfServiceState
- QueuedState
- ReceivedState
- UnavailableState.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID for the Sip Peer (trunk)
localCallId	Local Call ID of the call to query.
globalCallId	Global Call ID of the call to query.

Returns

- result – true or false

- errorDescription – if result false
- globalCallId
- callState
- localCallId

Notes

advGetSipPeerStatus

advGetSystemConferences

Definition

advGetSystemConferences (sessionId, objectId)

Description

This operation is used to trigger the MiVoice Business to generate conference events for any active conferences in the MiVoice Business.

The prerequisite for using this operation is that the application must first invoke advMonitorConferences. Also, if multiple applications are monitoring the same MiVoice Business for conferences it will result in duplicate events to each application when this operation is invoked.

Example:

- Application A starts a system conference monitor on MiVoice Business 1
- Phones 2000, 2001 and 2002 start a conference
- Application A receives a conference event
- Application B starts a system conference monitor on MiVoice Business 1
- Application B invokes the “advGetSystemConferences” operation for MiVoice Business 1
- Applications A and B will receive a conference event (for application A this will be a duplicate event)

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the geticpld operation.

Returns

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

Notes

The prerequisite for using this operation is that the application must first invoke `advMonitorConferences`. Also, if multiple applications are monitoring the same MiVoice Business for conferences it will result in duplicate events to each application when this operation is invoked.

`advGetTdmTrunkId`

Definition

`advGetTdmTrunkId (sessionId, icpld, trunkNum)`

Description

This operation provides the object ID for a TDM trunk on a MiVoice Business. The application uses this TDM trunk object ID in other operations to specifically identify a MiVoice Business trunk to the Mitel OIG. An application must call this operation before setting a monitor on the TDM trunk.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the <code>getIcpld</code> operation.
trunkNum	Number of the TDM trunk to be monitored.

Returns

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

Notes

`advGetTdmTrunkId`

`advGetTdmTrunkStatus`

Definition

`advGetTdmTrunkStatus (sessionId, objectId)`

Description

This operation allows an application to check the in-service and out-of-service state of a specific MiVoice Business TDM trunk.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID for the TDM Trunk.

Returns

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

Notes

advGetTdmTrunkStatus

advIntrudeCall

Definition

advIntrudeCall (sessionId, objectId, callId)

Description

This operation allows the caller to conference himself into an existing conversation after an attempt to call results in failure due to the destination being busy. The switch must be programmed to allow the caller this privilege. Note that this function is invoked when the calling device is in the Failed state.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the object intruding
callId	Call ID of the call affected.

Returns

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

Notes

advIntrudeCall

advLoginACDAgent

Definition

advLoginACDAgent (sessionId, objectId, agentDn, pin)

Description

This operation allows an application to login a Traditional ACD agent and an internal hotdesk ACD2 agent. When logging in a traditional Agent, ensure the pin attribute is Null as it is not used by the Mitel OIG. When logging in a hotdesk ACD2 agent a pin is needed for login but is not needed for logout.

Logging in a hotdesk ACD2 agent (User DN) causes the agent to take control of the MiVoice Business phone (prime line, line keys, softkeys, etc.) identified by objectId so that the phone now has a new prime line DN, which is the hotdesk ACD2 agent DN. The MiVoice Business phone prime line (registration DN) will go unavailable as long as the hotdesk ACD2 agent is logged in. Logging out causes the Registration DN (with line keys, softkeys, etc.) to be restored, and the User DN to become unavailable.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of a MiVoice Business phone.
agentDn	Agent DN logging in.
Pin	Pin to support login.

Returns

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

Notes

When logging in a traditional Agent, ensure the pin attribute is Null; the pin is not used by the Mitel OIG for a traditional agent log in.

advLoginExternalACDAgent

Definition

advLoginExternalACDAgent (sessionId, objectId, pin)

Description

This operation allows an application to login an external hotdesk ACD2 agent. When logging in an external hotdesk ACD2 agent a pin is needed for login, but is not needed for logout.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of the External Hot Desk Agent (EHDA).
Pin	Pin to support login.

Returns

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

Notes

advLoginExternalACDAgent

advLogoutACDAgent

Definition

advLogoutACDAgent (sessionId, objectId)

Description

This operation allows an application to log out an ACD agent; traditional ACD agent and hotdesk ACD2 agent.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of the Agent.

Returns

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

Notes

advLogoutACDAgent

advLogoutExternalACDAgent

Definition

advLogoutExternalACDAgent (sessionId, objectId)

Description

This operation allows an application to log out an external ACD2 hotdesk agent.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	sessionId was provided by the Mitel OIG upon successful login.
objectId	Object ID of the External Hot Desk Agent (EHDA).

Returns

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

Notes

advLogoutExternalACDAgent

advMonitorACDExpress

Definition

advMonitorACDExpress (sessionId, icpld)

Description

This operation establishes a class level device feature monitor on the specified class of objects. The application is notified of changes in feature programming for the class by means of feature events. This specific monitor will receive ACD Express Agent feature events for ACD Express Agents on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

advMonitorACDExpress

advMonitorACD2AgentFeatures

Definition

advMonitorACD2AgentFeatures (sessionId, icpld)

Description

This operation establishes a class level feature monitor on the specified class of objects. Thereafter, the application is notified of changes in feature programming for the class by means of feature events. This specific monitor will receive ACD2 agent feature events for ACD2 agents on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the geticpld operation.

Returns

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

Notes

1. The monitoring concept used in the Mitel OIG is that an application first gets an object ID on the object the application is interested in. This confirms the MiVoice Business has such an object, its type and that it can be monitored. Then an application can create a monitor on the object using the object ID. All events generated related to that object should include the object ID.
2. In the case of class level monitors (like this operation), there is no specific MiVoice Business object involved. A class level monitor is a Mitel OIG concept used to reduce the number of monitors needed to get status information from a large group/type of objects. In the case of ACD2 agent features, there is no one object. The application is able to create a special monitor that provides status events for all ACD2 agent features in a MiVoice Business. The Mitel OIG cannot provide a specific object ID as there can be many ACD2 agents at one time. The Mitel OIG does not provide a class level object ID because the ID is not unique in a MiVoice Business system cluster.
3. The object ID attribute is present in the provided event and the object ID is set to zero to indicate the object cannot be used as an object ID as 0 is not valid (i.e., all object IDs must be 1 or greater). Internal to the OIG, the object ID in this operation is used but then set to zero.

advMonitorAllBridges

Definition

advMonitorAllBridges (sessionId, icpld)

Description

The advMonitorAllBridges operation is used by applications to monitor all bridges on a MiVoice Business node for bridge feature events. The MiVB node bridge is related to a supervisor silent monitoring an ACD hot desk agent using the advACDSilentMonitor operation defined in this document.



Note: This operation creates a MiVoice Business level monitor and is not resilient.

Attributes

ATTRIBUTE	DESCRIPTION
-----------	-------------

sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the agent.

Returns

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

Notes

1. This operation must be called before any feature events for Bridges are received.
2. This operation is only to be called once by an application for each supervisor.

advMonitorAllTrunkFeatures

Definition

advMonitorAllTrunkFeatures (sessionId, icpld)

Description

This operation establishes a class level feature monitor on the specified class of objects. Thereafter, the application is notified of changes in feature programming for the class by means of events. This specific monitor will receive in-service and out-of-service events for trunks on a MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

1. This operation must be called before any feature events for trunks are received.
2. This operation is only to be called once by an application for each class of objects.
3. This operation has no effect on the status of the trunks or any calls or feature settings at the trunks.

advMonitorConferences*Definition*

advMonitorConferences (sessionId, icpld)

Description

This operation establishes a class level feature monitor on the specified class of objects. Thereafter, the application is notified of changes in feature programming for the class by means of feature events. This specific monitor will receive conference feature events for all conferences on a MiVoice Business. This monitor reports all conference activity on the MiVoice Business. This activity will be reported in ConferenceFeatureEvents, which have various causes indicating whether a conference was initiated, added to, dropped from, held, split, etc. (See ConferenceFeatureEvent).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

- objectId – if result true
- icpId – if result true

Notes

1. The monitoring concept used in Mitel OIG is that an application first gets an object ID on the object the application is interested in. This confirms the MiVoice Business has such an object, its type and that it can be monitored. Then an application can create a monitor on the object using the object ID. All events generated related to that object should include the object ID.
2. In the case of class level monitors (in this case conferences), there is no specific MiVoice Business object involved. A class level monitor is a Mitel OIG concept used to reduce the number of monitors needed to get status information from a large group/type of objects. In the case of conferences there is no one object or group of objects. The application is able to create a special monitor that provides status events for all conferences in a MiVoice Business. The Mitel OIG cannot provide a specific object ID as there can be many conferences at one time. The Mitel OIG does not provide a class level object ID because the ID is not unique in a MiVoice Business system cluster.
3. The object ID attribute is present in the event and we set it to zero to indicate the object cannot be used as an object ID as 0 is not valid (i.e. all object IDs must be 1 or greater). Internal to the OIG, the object ID is used but then set to zero.

advSetACDAgentMakeBusy

Definition

advSetACDAgentMakeBusy (sessionId, objectId, busyState)

Description

This operation allows an application to set an ACD agent into busy state.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the agent.
busyState	Boolean with true indicating make busy.

Returns

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

Notes

advSetACDAgentMakeBusy

advSetACDAgentMakeBusyWithCode

Definition

advSetACDAgentMakeBusyWithCode (sessionId, objectId, busyState, busyCode)

Description

This operation allows an application to set an ACD agent into busy state. The busyCode used to specify a Reason Code as indicated by an integer. A Reason Code of zero is equivalent to specifying no Reason Code.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the agent.
busyState	Boolean with true indicating make busy.
busyCode	Reason Code is indicated by an integer. A Reason Code of zero is equivalent to specifying no Reason Code.

Returns

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

Notes

advSetACDAgentMakeBusyWithCode

advSetACDAgentPresence

Definition

advSetACDAgentPresence (sessionId, objectId, groupDn, presenceValue, invokerDn)

Description

This operation allows an application to change the presence of an ACD 2 agent in an ACD2 group on a MiVoice Business. This operation can be invoked on behave of the ACD2 agent or for a supervisor of the ACD2 agent (i.e., see invokerDn).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the ACD Agent
groupDn	Group DN where the ACD agent is configured

presenceValue	Boolean with true indicating present
invokerDn	Phone number of device changing the group presence (i.e., supervisor)

Returns

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

Notes

advSetACDAgentPresence

advSetACD2AgentPresenceAllGroups

Definition

advSetACD2AgentPresenceAllGroups (sessionId, objectId, groupPresenceValue, invokerDn)

Description

This operation allows an application to have an ACD agent become presence in all ACD groups where the ACD agent is configured in a MiVoice Business (join all ACD groups or leave all ACD groups).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the ACD2 agent.
groupPresenceValue	Boolean with true indicating present.
invokerDn	Phone number of object setting the presence (optional)

Returns

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

Notes

advSetACD2AgentPresenceAllGroups

advSetAutoAnswer

Definition

advSetAutoAnswer (sessionId, objectId, autoAnswerMode)

Description

This operation allows an application to change the auto answer status of a phone.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the phone to set to auto answer.
autoAnswerMode	Boolean with true indicating auto answer.

Returns

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

Notes

advSetAutoAnswer

advSetGroupPresence

Definition

advSetGroupPresence (sessionId, objectId, groupDn, presenceValue, invokerDn)

Description

This operation allows an application to change the presence of a member in a ring group or hunt on a MiVoice Business. This operation can be invoked by the member or on behalf of the member (i.e., see invokerDn).

The invokerDn attribute is the phone number of a device which changes the group presence for a member. If the operation is invoked without specifying an “invoker DN” the operation is subject to the “Group Presence Control” Class Of Service (COS) option setting of the “member DN” parameter. If the operation is invoked with the “invoker DN” parameter, the operation is subject to the “Group Presence Third Party Control” COS option setting of the “invoker DN”.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the member
groupDn	Group DN where member is configured
presenceValue	Boolean with true indicating present
invokerDn	Phone number of device changing the group presence (i.e., supervisor)

Returns

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

Notes

advSetGroupPresence

advSetMsgWaitingInd

Definition

advSetMsgWaitingInd (sessionId, objectId, state)

Description

This operation is used to turn on and turn off a message waiting lamp/indicator on a MiVoice Business phone.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the object with message waiting indicator
State	Boolean with true indicating on and false indicating off.

Returns

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

Notes

advSetMsgWaitingInd

advSetTrunkMakeBusy

Definition

advSetTrunkMakeBusy (sessionId, objectId, forced)

Description

This operation is used to set a trunk busy.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the trunk
forced	Boolean with true indicating make busy.

Returns

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

Notes

advSetTrunkMakeBusy

advStopMonitorACD2AgentFeatures

Definition

advStopMonitorACD2AgentFeatures (sessionId, icpld)

Description

This operation stops the advMonitorACD2AgentFeatures monitor on a specific MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

advStopMonitorACD2AgentFeatures

advStopMonitorACDExpressObjects

Definition

advStopMonitorACDExpressObjects (sessionId, icpld)

Description

This operation stops the advMonitorACDExpressFeatures monitor on a specific MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

advStopMonitorACDExpressObjects

advStopMonitorAllBridges

Definition

advStopMonitorAllBridges (sessionId, icpld)

Description

This operation is used by applications to stop monitoring all bridges on a MiVoice Business (i.e., stop MiVoice Business node class level monitor for bridge feature events).

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

None

advStopMonitorAllTrunkFeatures

Definition

advStopMonitorAllTrunkFeatures (sessionId, icpld)

Description

This operation stops the advMonitorAllTrunkFeatures monitor on a specific MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

advStopMonitorAllTrunkFeatures

advStopMonitorConferences

Definition

advStopMonitorConferences (sessionId, icpld)

Description

This operation stops the advMonitorConferenceFeatures monitor on a specific MiVoice Business.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
icpld	A specific MiVoice Business object ID provided by the Mitel OIG when the application called the getIcpld operation.

Returns

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

Notes

advStopMonitorConferences

advTapCall

Definition

advTapCall (sessionId, objectId, localCallId, dn)

Description

TapCall connects a call that is waiting on an ACD path to another device, without removing the call from the ACD queue. This function can be used to enable the caller to listen to messages and/or interact with an IVR while waiting for his turn in the queue. If the caller reaches the head of the queue during this activity, the “tap” call is aborted and the caller is connected to the available agent. Tap call destinations can be softphone ports on a MiVoice Business.

An application can set a monitor on an ACD2 Path (not a Group), and if all agents in the group(s) belonging to that ACD2 Path are busy, an incoming call is queued to the Path (ACD2PathEvent, ReceivedState, ACDCallQueued). An application can then “tap” the waiting call.

An application uses TapCall with the the local call ID and the phone DN designating the device which will tap into the call. No event will be reported to the Path monitor or to the calling party, but the tapping device monitor will report a ReceivedEvent with a cause of TapCall.

When the tapping device answers, the tapping device monitor reports another CallReceivedEvent but with state EstablishedState and cause AnswerInvoked. Again, no other monitor gets a report at this time, but the tapping device now has an audio connection with the call.

When an agent becomes available, its monitor immediately reports a CallReceived event with a cause of ACDCallReceived, and the Path monitor reports ACDCallDelivered and goes into Idle state. No event is reported to the tapping device monitor or the calling device monitor.

When the agent answers the call, the agent monitor gets the usual Established event, and the monitor on the caller gets Established event while the tapping device monitor reports a Call Cleared event.

If the caller hangs up in the middle of the tap call interaction, then the Path monitor immediately reports ACDCallAbandoned, and the tapping device monitor reports Call Cleared. Tap calls are supported on destinations residing in the local MiVoice Business only.

Attributes

ATTRIBUTE	DESCRIPTION
sessionId	Provided by the Mitel OIG upon successful login.
objectId	Object ID of the ACD path.
localCallId	Local Call ID of the call to be tapped.
Dn	Phone number of path.

Returns

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

Notes

1. Only calls that are queued to an ACD2 Path may be "tapped". Thus, the objectId attribute listed above is for a monitor that has been set on the ACD2 path.
2. To obtain the required local call ID, use the ACD2PathEvent {cause: CDCallQueued}
3. advTapCall

Call Control Service Events

There are two main categories of Call Control Service events, Standard and Advanced.

Call Control Service Events – Standard Type

This section describes the Mitel OIG events provided to an application. The application can receive events synchronously or asynchronously. Synchronously means the application polls the Mitel OIG for events. Asynchronously means the application provides a way for the Mitel OIG to send events to the application when the events occur. This section describes the Call Control Service Events for the Standard type only.

Standard Event Classes

- Mitel OIG events are generated as a result of monitoring objects or features on the MiVoice Business. 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 Class (for Call Control Server Standard)

Events in Standard Call Event Class

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

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

AccountCodeSetEvent

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 or the other device connected in the call disconnects. An event cause of Cleared means the other device disconnected. An event cause of ClearedInvoked means this device disconnected.

When the monitored device is operating in hands-free mode and the other device invokes disconnect, this device receives a CallOriginatedEvent because the phone is off hook.

An Event cause of Cleared is reported to a monitor on an ACD2 agent when the caller abandons the call while the agent phone is ringing. In this case, the monitored phone goes to Idle State after the CallClearedEvent.



Note: No device information is provided in a Call Clear event when clearing a three-party conference. On clearing a three-party conference, the party invoking Clearcall gets the event Call_Clear, Clear Invoked. No device information comes in this event. No device information is provided in the other CALL_CLEAR events either.

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).

DATA	DESCRIPTION
	ACDAgentTimeout (this device is an ACD2 agent who did not answer in time. Such calls will normally be re-queued to the path).
Event Attribute (Group Member Answered)	Event data used to determine whether a Personal Ring Group member has answered a call.
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 (Mitel OIG Time Stamp)	The time when the Event was received at the Mitel 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. Conference data is also provided in ConferenceHeldEvent, DeviceDroppedEvent, and CallRetrievedEvent.

The event flow received by monitors within a conference on one MiVoice Business is different compared to monitors received when the monitors are on different MiVoice Business instances.

When a Silent Monitor (advanced operation) 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.

Event Data

DATA	DESCRIPTION
Event Type	CallConferenceEvent
Event State	EstablishedState.

DATA	DESCRIPTION
Event Cause	ACDSilentMonitor ACDSilentMonitorEstablished Conferenced ConferenceInvoked Intrusion IntrusionInvoked.
Event Local Call ID	Identifies the conference call.
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). This list is not provided across a DPNSS trunk connecting MiVoice Business instances. This means that a remote phone from the MiVoice Business with a conference does not receive a list.
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 and 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 (Mitel OIG Time Stamp)	The time when the Event was received at the Mitel 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 ring back 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	<code>CallDeliveredEvent</code>
Event State	<code>DeliveredState</code>
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	<code>ACDSilentMonitorInvoked</code> <code>Delivered (destination alerting)</code> <code>DivertedAlwaysTo</code> <code>DivertedNoAnswerTo</code> <code>DivertedOnBusyTo</code> <code>HandoffPush</code> <code>RedirectedOnError</code> <code>RedirectedTo</code> <code>UnsupervisedTransfer.</code>
Event Device Data (Calling Device)	
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 MiVoice Business.
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 MiVoice Business.
Event Attribute (Line Appearance Number)	The line appearance (keyline or multicall group number) of the other device, if such is involved.

DATA	DESCRIPTION
Event Attribute (Line Appearance Name)	Returns the name associated with the line appearance (keyline or multical 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 MiVoice Business.
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 (Mitel OIG Time Stamp)	The time when the Event was received at the Mitel 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.
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 Cause	Indicates how the call entered this state. Possible values include: Answered AnswerInvoked ConferenceSplit ConferenceSplitInvoked Pickup PickupInvoked SupervisedTransfer.
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 MiVoice Business.
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 MiVoice Business.
Event Attribute (Line Appearance Number)	The line appearance (keyline or multical group number) of the connected device, if such is involved..
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 (Splitting Device)	Identity of the Device that split a conference to cause call to this device.
Event Device Data (Transferring 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.

DATA	DESCRIPTION
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 MiVoice Business 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 a 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

DATA	DESCRIPTION
	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() using Phone1
4. Verify Events:

On Phone2:

1. CallHeldEvent HeldState ConsHold
3. CallHeldEvent HeldState HardHold

On Phone1:

1. 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 MiVoice Business 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

DATA	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)

DATA	DESCRIPTION
	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

DATA	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. CallsWaiting – 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 Transferor 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.

DATA	DESCRIPTION
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 MiVoice Business.

Event on an 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	<code>CallReceivedEvent</code>
Event State	<code>ReceivedState</code>
Event Cause	<code>ACDHelpCall</code> <code>ACDCallReceived</code>

DATA	DESCRIPTION
	CallbackMatured DivertedAlwaysFrom DivertedNoAnswerFrom DivertedOnBusyFrom NewCall (the normal case) ONSCallOriginated RecallHeldParty RedirectedFrom (for other reasons) TapCall TransferRecallNoAnswer TransferRecallOnBusy TransferredCall WaitingCallRinging
Event Device Data (Called Device)	This information describes the called device. When the event cause is ACDCallReceived, this information provides the identity of an ACD group that distributed call to this monitored device (i.e., ACD agent). If this event cause is ACDCallReceived and the ACD Agent Group Number is NULL, then the agent is part of a Remote Agent Group and the call came from a remote ACD Path. In this case, Dialed Digits described below 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. When the Call Received event has a cause of DivertedNoAnswerFrom and is from a Personal Ring Group (PRG) with members (eg, EHDU and internal phone), then the called device will provide information about the PRG. Note: No Group Device information will be provided.
Event Attribute (Calling Device Local Call ID)	The Local Call ID for this call reported to the Calling Device (Other Device).
Event Device Data (Calling Device)	Identity of the Calling Device (Other Device) Exception: Cannot 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 transfer). 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 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 MiVoice Business.

DATA	DESCRIPTION
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 MiVoice Business.
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 PUBLIC_TRUNK Event Attribute PRIVATE_TRUNK Event Attribute VOICE_SET Event Attribute DATA_EXTENSION or -1 for error.
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.
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)	<p>Provides category of calling device on incoming trunk call (indicates whether call is from within a cluster or from PSTN). Values are:</p> <p>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.</p> <p>Note: If a device in PSTN conferences in a device from the private network, then the CLC can be one of the following:</p> <p>CLC_DEC CLC_DASS2 CLC_PSTN CLC_MF5.</p> <p>But never:</p> <p>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.</p> <p>A value of “-1” is provided when not a trunk call.</p>

DATA	DESCRIPTION
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).
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	CallTransferredEvent
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.
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.

DATA	DESCRIPTION
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)

DATA	DESCRIPTION
	<p>QueuedPartyDropped (ACD and call waiting)</p> <p>ThisDeviceDropped (ONS phone only; state = unavailable).</p> <p>ACDSilentMonitorDropped (only from a Silent-Monitored two-device call)</p> <p>ACDSilentMonitorMemberDropped (any member of a conference formed with a Silent Monitor)</p>
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 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.

DATA	DESCRIPTION
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 MiVoice Business to the next group member. A GroupEvent with CallAnswered cause is generated when one of those members answers.

Event Data

DATA	DESCRIPTION
Event Type	GroupEvent
Event State	idle or received
Event Cause	<p>CallDistributed – call has been distributed to group member(s).</p> <p>CallAnswered – call has been answered by a group member.</p> <p>CallDisconnected – call has been distributed to the members of a group and the caller has disconnected the call.</p> <p>CallOverflowed – call has been distributed to the members of a group and a ringing timer has expired. The call is transferred to an overflow point.</p> <p>CallWaiting – call is queued to a group when all group members are busy.</p> <p>CallWaitingPartyDisconnected – caller disconnects while queued on a busy group.</p>
Event Device Data (Group Member Device)	Identifies member device. Valid for CallDistributed and CallAnswered event causes.

DATA	DESCRIPTION
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.

InformationUpdateEvent

Description

This event is provided only in the case where a MiVoice Business PSTN trunk does not provide the caller name in a CallReceivedEvent. This event is the same as a CallReceived Event with the addition of the external caller name.

This event provides a name update when PSTN trunk signaling does not provide calling name until after a call received event is generated. This event will have the same information as a call received event. The InformationUpdateEvent will have the same state as a call received event and the cause will be calling name updated. When a phone is monitored and the phone receives an incoming trunk call that does not initially provide calling name, use the InformationUpdateEvent to get the calling name. When a trunk is monitored and the trunk gets a call from an external party (incoming trunk call), use the InformationUpdateEvent to get the calling name.

See CallReceivedEvent for event definition and description.

InServiceEvent

Description

This event is sent when this monitored device is put into service by the MiVoice Business. A device is taken out of service when a phone user exceeds a MiVoice Business allowable time period while attempting to place a call (e.g., user lifted phone receiver and did 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 taken out of service and put back into service by MiVoice Business maintenance activities (e.g., power up, reset of MiVoice Business, reset device).

Event Data

DATA	DESCRIPTION
Event Type	InServiceEvent
Event State	IdleState
Event Cause	DeviceInService ControllerCommsRestored: when communication with MiVoice Business restored for non-Resilient monitor.
Event Attribute (Local Call ID)	Null (0)

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

DATA	DESCRIPTION
Event Type	MonitorFailedEvent
Event State	Nil
Event Cause	Unknown

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

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

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 MiVoice Business allowable time period while attempting to place a call (e.g., user lifted phone receiver and did 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 taken out of service and put back into service by MiVoice Business maintenance activities (e.g., power up, reset of MiVoice Business, 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 MiVoice Business fails.
Event Attribute (Local Call ID)	Null (0).

PrgMonitorFailedEvent

Description

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

Event Data

DATA	DESCRIPTION
Event Type	PrgMonitorFailedEvent
Event State	Nil
Event Cause	Unknown

PrgMonitorSetEvent

Description

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

Event Data

DATA	DESCRIPTION
Event Type	PrgMonitorSetEvent

Event State	Nil
Event Cause	Unknown

ResilientDeviceEvent

Description

This event is sent when a MiVoice Business phone is changed from resilient to non-resilient and non-resilient to resilient (MiVoice Business administrator changes the phone configuration using the MiVoice Business configuration management web browser UI). This event also indicates when a monitored phone falls back to the primary MiVoice Business from the secondary MiVoice Business.

When a phone is made resilient, the application can also receive device feature events for call forwarding from the secondary MiVoice Business (if application has created device feature monitor on the phone).

Event Data

DATA	DESCRIPTION
Event Type	ResilientDeviceEvent
Event State	Nil
Event Cause	REDIRECTED_HANDOFF, RESILIENT_INFO_UPDATE

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

DATA	DESCRIPTION
	RoutingDeviceRadStarted
Event Attribute (Local Call ID)	Local Call ID for the monitored routing device.
Event Device Data (Routing Device)	Identity of the routing device.
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 (Transferee Device)	Identity of the transferring device for RoutingDeviceTransCallQueued cause.
Event Device Data (Reroute 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,	yes	caller	forwarder

CALL CAUSE	ROUTING DEVICE	OTHER PARTY	THIRD PARTY
CFALWAYS) CallQueue,			
RoutingDeviceRadStarted	yes	no	no

Events in Standard 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
- Auto answer

ActivateFeatureEvent

Description

This event identifies a binary feature that has changed.

Event Data

DATA	DESCRIPTION
Event Type	ActivateFeatureEvent.
Event State	FeatureEnabled or FeatureDisabled.
Event Attribute (Feature)	<p>The feature which has changed. This may be one of:</p> <p>AutoAnswerFeature</p> <p>DoNotDisturbFeature (only theDND feature is allowed for ACD2 paths and groups. Now works for non-ACD sets as well)</p> <p>InServiceFeature</p> <p>MakeBusyFeature</p> <p>MakeBusyStateFeature</p> <p>DoNotDisturbAll (Disabled state only. Note: this message may be received even when DoNotDisturb is already disabled).</p> <p>MsgWaitingIndicator</p> <p>MsgWaitingIndAll (Disabled state only. Note: this message may be received even when MsgWaitingIndicator is already disabled).</p> <p>ACDAgentBlockedFeature (see Note below).</p>

DATA	DESCRIPTION
Event Attribute (Make Busy Code)	Reason Code for the MakeBusyStateFeature.



Note: The ACDAgentBlockedFeature may be Activated/Deactivated if the MiVoice Business supports Networked ACD. “FeatureEnabled” means that an agent has logged into an ACD2 group used as a remote subgroup, but is blocked from receiving calls until setup is complete. “FeatureDisabled” means that the agent will now receive calls. In addition, the ACDAgentBlockedFeature may be Activated (enabled) when a call is distributed from the remote ACD Path (about to be delivered to the agent), and then Deactivated (disabled) after the Call Received Event. This will be a short interval, designed to prevent calls arriving simultaneously from being lost.

AutoAnswerFeatureEvent

Description

This event is sent when a phone auto answer mode is enabled or disabled.

Event Data

DATA	DESCRIPTION
Event Type	AutoAnswerFeatureEvent
Event State	FeatureEnabled OR Feature Disabled
Event Attribute (phone number)	Identity of the phone.

ForwardFeatureEvent

Description

This feature event is presented when one of: CF_ALWAYS, CF_BUSY_EXTERNAL, CF_BUSY_INTERNAL, CF_NO_ANSWER_INTERNAL, CF_NO_ANSWER_EXTERNAL, CF_ALL_MODES, is changed on a monitored phone.

Event Data

DATA	DESCRIPTION
Event Type	ForwardFeatureEvent.
Event State	FeatureEnabled or FeatureDisabled.
Event Attribute (Feature)	CFAlways CFBusyExternal CFBusyInternal

CFNoAnswerExternal

CFNoAnswerInternal

CFAAllModes.

Note: CFAAllModes 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.

HotDeskUserFeatureEvent

See UserIDFeatureEvent below.

JoinGroupFeatureEvent

Description

The JoinGroupFeatureEvent is sent when a personal 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 personal 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.

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 (Ext HotDesk User Dn)	Returns External Hot Desk User DN.
Event Attribute (UserType)	Returns the user type: Internal or External

Events in Standard System Event Class

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

- ICP Communication
- OIG Gateway Shutdown

ICPCommunicationEvent

Description

This event indicates the connection status to a specific MiVoice Business.

Event Data

DATA	DESCRIPTION
Event Type	ICPCommunicationEvent
icpId	MiVoice Business object Id
Time	OIG time
connectedState	Indicates the MiVoice Business connection status to the OIG

OIG GatewayShutdownEvent

Description

This event indicates the Mitel OIG has shutdown.

Event Data

DATA	DESCRIPTION
Event Type	OIG GatewayShutdownEvent
Time	OIG time

Call Control Service Events – Advanced Type

This section describes the Mitel OIG events provided to an application. The application can receive events synchronously or asynchronously. This means that the application can poll the Mitel OIG for events or ask the Mitel OIG to send events when they occur. An application using Mitel OIG for the Advanced service will also need to process some or all of the events defined in the Call Control Service Events – Standard Type section of this document.

Advanced Event Classes

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

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

Events in Advanced Call Event Class

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

- **ACD Express Group**
- **Trunk Digits Outpulsed**
- **Work Timer** Expired

ACDExpressGroupEvent

Description

ACDExpressGroupEvent only occurs when the monitored device is an ACD Express Group or (for the ACDRadStarted cause only) an IP RAD device.

Event Data

DATA	DESCRIPTION
Event Type	ACDExpressGroupEvent
Event State	ReceivedState, UnavailableState, IdleState.
Event Cause	Indicates why the call was held. Possible values include: ACDRequest ACDRequestDisconnected ACDRequestRerouted ACDRequeueRequest ACDRequeueRequestDisconnected ACDRequeueRequestRerouted ACDInterflowedDisconnected ACDInterflowedRerouted ACDCallRedirected ACDCallQueued ACDCallDelivered ACDCallAbandoned
Event Device Data (ACD Agent)	Identity of agent for ACDCallDelivered cause.
Event Device Data (ACD Caller)	Identity of caller (if known) for all ACDRequest and all ACDRequeue causes.
Event Device Data (ACD Forwarding Device)	Identity of device which forwarded call to the Express Group DN, if required.
Event Device Data (Interflow Device)	Identity of device to which a call is interflowed out of the current path with an interflow point, if required.
Event Device Data Path Unavailable Answer Point)	Identity of device for ACDRequestRerouted and ACDRequeueRequestRerouted causes, if required.

DATA	DESCRIPTION
Event Device Data (ACD Path Exit Device)	Identity of device to which a call is rerouted out of the path. For the causes ACDRequestRerouted and ACDRequeueRequestRerouted, this is the Unavailable Answer Point DN; for ACDInterflowedRerouted, it is the Interflow DN; for ACDCallRedirected, it is the number to which the call is redirected (will be the Interflow DN if no device was specified in the redirect command).
Event Attribute (External Features Allowed)	Indicates whether certain more-recently-added Call Manipulation routines are now allowed.
Event Attribute (Event Class)	ACDExpress EventClass.
Event Attribute (Current Global Call ID)	Each call has a current 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.

TrunkDigitsOutpulsedEvent

Description

TrunkDigitsOutpulsedEvent reports on monitored outgoing trunks. For a system with a trunking gateway configuration, all trunk calls are routed through the trunking gateway. When an EHDU makes a call, the call will involve two trunks on the trunking gateway; one for the EHDU external device and one for the outgoing call.

Normally, digits outpulsed from a trunk are identical to digits dialed by users with ARS leading digits removed. However, with MiVoice Business advanced ARS programming (e.g. Digit Modification) the digits outpulsed can be different from the digits dialed.

Event Data

DATA	DESCRIPTION
Event Type	TrunkDigitsOutpulsedEvent
Event State	Any State: The current state of the device.
Event Cause	TrunkDigitsOutpulsed
Event Attribute (Outpulsed Digits)	Outpulsed Digits
Event Attribute External HotDesk User DN)	External Hot Desk User DN
Event Attribute (Current Global Call ID)	Each call has a current 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 the Secondary Global Call ID is used.

WorkTimerExpiredEvent

Description

WorkTimerExpiredEvent is sent when an ACD agent work timer expires (either normally or when canceled). This event is generated with the same Cause for a work timer expiring normally and a work timer canceled.

Event Data

DATA	DESCRIPTION
Event Type	WorkTimerExpiredEvent
Event State	Any State: The current state of the device.
Event Cause	WorkTimerExpired.
Event Attribute (Local Call ID)	Null (0)

Events in Advanced Feature Event Class

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

- ACD_AGENT
- ACD_AGENT_BLOCKED
- ACD2_LOGIN_NO_LICENSE
- BRIDGE_CONNECTED
- BRIDGE_DISABLE_COACHING
- BRIDGE_DISCONNECTED
- BRIDGE_ENABLE_ALL_COACHING
- BRIDGE_ENABLE_COACHING
- BRIDGE_JOIN
- BRIDGE_LEAVE
- JOIN_ALL_ACD_GROUPS
- LEAVE_ALL_ACD_GROUPS

ACD_AGENT

DescriptionAn ACDAgentFeatureEvent is generated when an ACD agent logs in or out of a hot desk enabled phone. When setting a device monitor on the ACD Agent Hot Desk DN and a Class Level ACD Agent feature monitor on the OIG, two events are generated independently; one for the class level ACD Agent feature monitor advMonitorACD2AgentFeatures and one for the device monitor.

The same "HOT_DESK_USER" with "FEATURE_ENABLED" event is provided for both Hot Desk ACD Agent log in and Hot Desk User log in. The application can map a feature event to a

specific type of login (hot desk user or hot desk ACD Agent) using the Agent DN in the feature event.

"FEATURE_ENABLED" is a specific reference to the fact that the event HOTDESK_USER with FEATURE_ENABLED indicates a Hot Desk log in was successful (Hot desk user log in or Hot desk ACD Agent log in); it does not specifically indicate an ACD Agent hot desk log in only (FEATURE_DISABLED means hot desk log out). When the Mitel OIG generates a class feature event there is no object ID (i.e., object ID is zero) as the event does not apply to a specific device monitor.

Doing a Traditional ACD Agent log in using Agent *101 on a phone with Extension number 101 provides the following Traditional ACD log in event (some information in the event is removed to make review easier):

```
<featureEvent xmlns="http://mitel.com/oig/cc/advanced" objectId="0" time="1426876347">
  <eventType>ACD_AGENT</eventType>
  <attributeName>FEATURE_EVENT_STATE</attributeName><attributeValue>FEATURE_ENA
    BLED</attributeValue>
  <attributeName>AGENT_DN</attributeName><attributeValue>101</attributeValue>
  <attributeName>AGENT_ID</attributeName><attributeValue>*101</attributeValue>
  <attributeName>ICP_ID</attributeName><attributeValue>1</attributeValue>
```

Doing a Hot Desk ACD Agent log in using hot desk ACD Agent DN 136 on a hot desk enabled phone with extension number 137 provides the following Hot Desk ACD Agent log in event (a Hot Desk User log in provides the same event as when doing a Hot Desk ACD Agent log in. Both events indicate hot desk user type). The request to log in a hot desk ACD Agent is different from the request to log in a hot desk user and is different from request to log in Traditional ACD Agent. Some information in the event is removed to make review easier.

```
<featureEvent xmlns="http://mitel.com/oig/cc/advanced" objectId="0" time="1426876359">
  <eventType>HOT_DESK_USER</eventType>
  <featureEventAttribute><attributeName>FEATURE_EVENT_STATE</attributeName><attribute
    Value>FEATURE_ENABLED</attributeValue>
  <attributeName>USER_DN</attributeName><attributeValue>137</attributeValue>
  <attributeName>REGISTRATION_DN</attributeName><attributeValue>136</attributeValue>
  <attributeName>USER_TYPE</attributeName><attributeValue>Internal</attributeValue>
  <attributeName>ICP_ID</attributeName><attributeValue>1</attributeValue>
```

Event Data

RETURN TYPE	DATA	DESCRIPTION
Digits	Event Attribute (ACD Agent	Identity of the phone where logging in/out and the

	ID / DN)	agent ID/DN logging in/out. For Traditional ACD agent the USER_DN, REGISTRATION_DN). Agent ID is 1-7 digits. For Hot Desk Agent, Internal an agent DN is provided. For Hot Desk ACD Agent External an EHDA DN is provided..
Int	Event Attribute (Feature)	Indicates the feature (login or logout status) related to this event. Enabled means log in and Disabled means log out. A hot desk log in may return a different event called "CD2LoginNoLicensesFeatureEvent". if Mitel OIG or MiVoice Business controller does not have the proper licensing.

ACD_AGENT_BLOCKED

Description

Event Data

RETURN TYPE	DATA	DESCRIPTION
Digits	Event Attribute (ACD Agent ID / DN)	Identity of the agent logging in or out. For ACD2 an agent ID of 1-7 digits can be specified. For ACD2 Hotdesk Agent Internal an agent DN is provided. For Hotdesk Agent External an EHDA DN is provided.
Int	Event Attribute (Feature)	Indicates the feature (login or logout) related to this event. May return: ACD2LoginNoLicensesFeature.

ACD2_LOGIN_NO_LICENSE

Description

Event Data

RETURN TYPE	DATA	DESCRIPTION
Digits	Event Attribute (ACD Agent ID / DN)	Identity of the agent logging in or out. For ACD2 an agent ID of 1-7 digits can be specified. For ACD2 Hotdesk Agent Internal an agent DN is provided. For Hotdesk Agent External an EHDA DN is provided.
Int	Event Attribute (Feature)	Indicates the feature (login or logout) related to this event. May return: ACD2LoginNoLicensesFeature.

BRIDGE_CONNECTED

Description

This event is generated when a phone or application invokes the "Silent monitor" feature on an agent or telephone. This event will be returned to all advanced call control service applications which either have set the "advMonitorAllBridges" on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features..

The following is an example:

Received Event Type: FEATURE_EVENT Feature Event Type: BRIDGE_CONNECTED

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2007

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 1

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2007

The above event was generated when 2007 started a silent monitor on 2004.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has created a silent monitor; indicates connected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that that created the silent monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUESTORS	Indicates how many supervisors are silent monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on /off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

BRIDGE_DISABLE_COACHING

Description

This event is generated when a phone or application clears the “Coach Call” feature on an agent or telephone. When a supervisor removes a request to coach, the supervisor will still remain connected to the bridge for whisper coach. This event will be returned to all advanced call

control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features..

For example:

Received Event Type: FEATURE_EVENT Feature Event Type:
BRIDGE_DISABLE_COACHING

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2006

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 2

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2007

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2006

The above event was generated when 2006 ended the whisper coach call on DN 2004 while a second supervisor 2007 was also silent monitoring.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has created a silent monitor; indicates connected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that that disabled coaching related to the silent monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUE STORS	Indicates how many supervisors are silent monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on/off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

BRIDGE_DISCONNECTED

Description

This event is generated when all supervisors/phones have cleared their silent monitors on the agent or telephone. This event will be returned to all advanced call control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features.

For example:

Received Event Type: FEATURE_EVENT Feature Event Type: BRIDGE_DISCONNECTED

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2004

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when all supervisors have disconnected a silent monitor on the bridge party DN; indicates disconnected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates the bridge party DN that is disconnected from all silent monitoring.

BRIDGE_ENABLE_ALL_COACHING

Description

This event is generated when an application invokes the “Coach Call” feature with the COACH_ALL type specified on an agent or telephone. When a supervisor requests the coach all feature, all coaching supervisors and the ACD hot desk agent can hear each other. This event will be returned to all advanced call control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features.

For example:

Received Event Type: FEATURE_EVENT Feature Event Type: BRIDGE_ENABLE_ALL_COACHING

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2007

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 2

Attribute Name: BRIDGE_COACH_MODE Value: COACH_ALL

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2007

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2006

The above event was generated when 2007 invoked the Coach Call feature with the COACH_ALL type specified.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has created a silient monitor; indicates connected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that enabled coach all related to the silient monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUESTORS	Indicates how many supervisors are silient monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on /off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

BRIDGE_ENABLE_COACHING

Description

This event is generated when a phone or application invokes the “Coach Call” feature on an agent or telephone. The Coach Call feature allows a supervisor to speak with a specific ACD hot desk agent. This event will be returned to all advanced call control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features.

For example:

Received Event Type: FEATURE_EVENT Feature Event Type: BRIDGE_ENABLE_COACHING

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2006

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 1

Attribute Name: BRIDGE_COACH_MODE Value: WHISPER_COACH

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2006

The above event was generated when 2006 invoked “Coach Call”.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has created a silent monitor; indicates connected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that enabled coach related to the silent monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUESTORS	Indicates how many supervisors are silent monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on /off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

BRIDGE_JOIN

Description

This event is generated when a phone or application sets a silent monitor on an agent or phone. This event will be returned to all advanced call control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features.

For example:

FEATURE_EVENT Feature Event Type: BRIDGE_JOIN

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2006

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 2

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2007

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2006

The above event was generated when 2006 started a silent monitor on 2004.



Note: There can be up to 6 requestors (supervisors). The “BRIDGE_REQUESTOR_DN” and “BRIDGE_COACH_MODE” use the same list index in the event. When 2007 is the first “BRIDGE_REQUESTOR_DN” then look at the first BRIDGE_COACH_MODE value to determine coach mode for 2007.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has created a silent monitor; indicates connected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that enabled coach related to the silent monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUESTORS	Indicates how many supervisors are silent monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on /off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

BRIDGE_LEAVE

Description

This event is generated when a supervisor/phone clears their silent monitor on an agent or telephone. This event will be returned to all advanced call control service applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring supervisor phone, phone or ACD hot desk agent using monitor device features..

For example:

Received Event Type: FEATURE_EVENT Feature Event Type: BRIDGE_LEAVE

Feature Event Attributes:

Attribute Name: ICP_ID Value: 1

Attribute Name: BRIDGE_PARTY_DN Value: 2004

Attribute Name: BRIDGE_STATUS Value: Connected

Attribute Name: INVOKER_DN Value: 2007

Attribute Name: BRIDGE_NUMBER_REQUESTORS Value: 1

Attribute Name: BRIDGE_COACH_MODE Value: COACH_OFF

Attribute Name: BRIDGE_REQUESTOR_DN Value: 2007

The above event indicates that 2007 is no longer monitoring 2004.

Event Data

RETURN TYPE	DATA	DESCRIPTION
See WSDL	Attribute Name: ICP_ID	Identifies the MiVB where bridge monitor event was generated.
See WSDL	Attribute Name: BRIDGE_PARTY_DN	Phone or ACD hot desk agent being monitoring
See WSDL	Attribute Name: BRIDGE_STATUS	Indicates when supervisor has removed a silent monitor; indicates disconnected status.
See WSDL	Attribute Name: INVOKER_DN	Indicates DN that created the silent monitor on bridge party DN.
See WSDL	Attribute Name: BRIDGE_NUMBER_REQUESTORS	Indicates how many supervisors are silent monitoring.
See WSDL	Attribute Name: BRIDGE_COACH_MODE	Indicates the coach on /off status of a supervisor. When there is more than 1 supervisors (ie, number of requestors greater than 1 in attribute above) then coach mode (this attribute) and requestor DN (the attribute below) are paired.
See WSDL	Attribute Name: BRIDGE_REQUESTOR_DN	Indicates the DN of a supervisor. When there is more than 1 supervisor (ie, number of requestors greater than 1 in attribute above) then coach mode (above attribute) and requestor DN (this attribute) are paired.

JOIN_ALL_ACD_GROUPS

Description

The feature event is generated when an ACD agent is marked as presence in all ACD2 groups in a MiVoice Business. When a Feature Monitor is set on a group member this event is generated.

Event Data

DATA	DESCRIPTION
Event Type	JoinAllACDGroupsFeatureEvent.
Event Attribute (Feature)	JoinAllACDGroupsFeature
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.

LEAVE_ALL_ACD_Groups

Description

The feature event is generated when an ACD agent is marked as absent in all ACD2 groups in a MiVoice Business. When a Feature Monitor is set on a group member, this event is generated.

Event Data

DATA	DESCRIPTION
Event Type	LeaveAllACDGroupsFeatureEvent.
Event Attribute (Feature)	LeaveAllACDGroupsFeature
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.

Events in Advanced System Event Class

None

Events in Advanced ACD Event Class

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

- ACD2 Path
- ACD2 Group

ACD2PathEvent

Description

This event only occurs when the monitored device is an ACD2 path or (for the ACDRadStarted cause only) an IP RAD device (see MiVoice Business RAD notes below). ACD2PathEvent is the only event that is reported when the monitored device is an ACD2 path.

Event Data

DATA	DESCRIPTION
Event Type	ACD2PathStarted
Event State	ReceivedState, UnavailableState, IdleState.
Event Cause	Possible values include: ACDRequest ACDRequestDisconnected ACDRequestRerouted ACDRequeueRequest ACDRequeueRequestDisconnected ACDRequeueRequestRerouted ACDInterflowedDisconnected ACDInterflowedRerouted ACDCallRedirected ACDCallQueued ACDCallQueuedOverflowed ACDCallQueuedFailed ACDCallDelivered ACDCallAbandoned ACDRadStarted - An ACD RAD has been started (ReceivedState), or an IP RAD has been started (EstablishedState; see Note below on IP RAD Support).
Event Device Data (ACD Agent)	Identity of agent for ACDCallDelivered cause.
Event Device Data (ACD Caller)	Identity of caller (if known) for all ACDRequest and all ACDRequeue causes.
Event Device Data (ACD2 RAD)	Identity of a RAD device (ACD2 path) for ACDRadStarted cause
Event Attribute (ACD2 Queue Position)	Provided only when the cause is ACDRadStarted, and that RAD is an IP RAD. In this case, this attribute provides the number of callers now listening to the IP RAD. This number should not be taken as the actual position of the call in the queue (See the note below on IP RAD Support).
Event Device Data (ACD Agent Group)	Identity of agent group involved. For causeACDCallDelivered, this is the group to which the agent belongs; for ACDCallQueued, it is the group to which the call is queued; for ACDCallQueuedOverflowed, it is the group from which the call (predictive) overflowed; for

DATA	DESCRIPTION
	ACDCallQueuedFailed, it is the group which was unavailable, and hence caused the failure.
Event Device Data (ACD Forwarding Device)	Identity of device which forwarded call to the ACD2 path DN, if required.
Event Device Data (Interflow Device)	Identity of device to which a call is interflowed out of the current path with an interflow point, if required.
Event Device Data (Path Unavailable Answer Point)	Identity of device for ACDRequestRerouted an ACDRequeueRequestRerouted cause, if required.
Event Device Data (ACD Path Exit Device)	Identity of device to which a call is rerouted out of the path. For the causes ACDRequestRerouted and ACDRequeueRequestRerouted, this is the Path Unavailable Answer Point DN; for ACDInterflowedRerouted, it is the Interflow DN; for ACDCallRedirected, it is the number to which the call is redirected (will be the Interflow DN if no device was specified in the redirect command).
Event Attribute (External Features Allowed)	Indicates whether certain more-recently-added Call Manipulation routines are now allowed. Currently the only Feature Type is TapCall.
Event Attribute (Event Class)	ACD2PathEventClass.
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.

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

DATA CAUSE	ACD CALLER DEV	AGENT GROUP DEV	ACD AGENT DEV	ACD FWDING DEV	INTER- FLOW DEV	PATH UNAV ANS-PT	ACD PATH EXIT DEV	ACD2 RAD DEV	ACD2 QUE- POS'N
ACDRequest	YES	NO	NO	YES	NO	NO	NO	NO	NO
ACDRequestDisconnected	YES	NO	NO	YES	NO	NO	NO	NO	NO
ACDRequestRerouted	YES	NO	NO	YES	NO	YES	YES	NO	NO
ACDRequeueRequest	YES	NO	NO	NO	NO	NO	NO	NO	NO
ACDRequ'Req'stDisc.	YES	NO	NO	NO	NO	NO	NO	NO	NO
ACDRequ'Req'stRer'td	YES	NO	NO	NO	NO	YES	YES	NO	NO
ACDInterflowedDisc.	NO	NO	NO	NO	NO	NO	NO	NO	NO
ACDInterflowedRer'td	NO	NO	NO	NO	YES	NO	YES	NO	NO
ACDCallRedirected	NO	NO	NO	NO	NO	NO	YES	NO	NO
ACDCallQueued	NO	YES	NO	NO	NO	NO	NO	NO	NO
ACDCallQueuedOverfl.	NO	YES	NO	NO	NO	NO	NO	NO	NO
ACDCallQueuedFailed	NO	YES	NO	NO	NO	NO	NO	NO	NO
ACDCallDelivered	NO	YES	YES	NO	NO	NO	NO	NO	NO
ACDCallAbandoned	NO	NO	NO	NO	NO	NO	NO	NO	NO
ACDRadStarted	NO	NO	NO	NO	NO	NO	NO	YES	YES

Notes

1. When a call is transferred to a monitored ACD path, the following event is generated: **CallReceivedEvent / ReceivedState / TransferredCall**. The Call Received event occurs immediately following a **ACDCallQueued** event on the path.
2. IP RAD Support: An **ACD2PathEvent** with a Cause of **ACDRadStarted** in the **EstablishedState** will be returned only by a monitor on an IP RAD device. It is provided as an **ACD2PathEvent** because when an IP RAD is started (unlike the usual ACD RAD device), no event occurs for a monitor on the ACD2 Path.

Details

The IP RAD device is supported when communicating with a MiVoice Business that provides IP RAD functionality. In its intended operation, the application will set both a monitor and a feature monitor on this device (just as on a regular phone), and the following event sequences may occur: (EventType – EventState – EventCause – FeaturesAllowed).

The event data can include the identity of a RAD device as well as the number of devices currently listening to the RAD recording (up to a maximum of 50). The number of devices should not be taken as the actual queue position since other factors such as multiple use, differing priorities, etc., will make it inaccurate for any such purpose.

In order to generate the ACD2 Path Event with a cause of **ACDRadStarted**, you must not only monitor the Path DN but also either the RAD hunt group pilot number where the RAD is programmed or the RAD DN itself.

Call events reported by a monitor on an IP RAD do not provide Call ID or other party information.

- **CallReceivedEvent – ReceivedState – NewCall – Feature allowed (AnswerCall)**
In a normal scenario, the application uses AnswerCall operation to cause a RAD to answer the call. The monitor on the RAD reports
- **ACD2PathEvent – EstablishedState – ACDRadStarted – feature allowed (ClearCall)**
This is the only ACD2PathEvent generated by a monitor set on any device other than an ACD2 Path. It is provided because a monitor on the associated ACD2 Path will generate no event when the RAD starts.
- **CallClearedEvent – IdleState – Cleared or ClearInvoked – None**
If the application fails to answer the call (before the RAD timer expires), then the IP RAD will go to IdleState, with DoNotDisturb also being set for the device. A Feature Monitor on the device is used to signal this, so that the application can put it back into service (cancel the DND). Otherwise, the DND is not set, and the IP RAD is ready for the next caller. This will be the case if the caller(s) all hang up before or after the RAD answers, or the RAD plays to completion and is disconnected, or the application tells the IP RAD to end (Clear) the call.

ACD2GroupEvent

Description

This event only occurs when the monitored device is an ACD2 Group, and that ACD2 Group is used by a remote path for handling some of its calls. Monitors set on ACD2 groups otherwise report no events, and are used only to obtain information about the group DN, name, number of members, and skills.

ACD2GroupEvent is the only event that may be reported when the monitored device is an ACD2 group.

Event Data

DATA	DESCRIPTION
Event Type	ACD2GroupEvent
Event State	ReceivedState, UnavailableState, IdleState.
Event Cause	<u>Possible values include:</u> ACDRemoteCallQueued ACDRemoteCallDequeued ACDRemoteCallDistributed
Event Device Data (ACD Remote Path)	Identity of remote ACD2 Path for which this ACD2 Group is acting as a Remote Subgroup. (ACDRemoteCallQueued and ACDRemoteCallDequeued only. See note below for ACDRemoteCallDistributed cause).
Event Device Data (ACD Caller)	Identity of caller (if known).
Event Device Data (ACD2 RAD)	Identity of a RAD device for ACDRemoteCallQueued cause
Event Attribute (ACD2 Queue Position)	Indicates position in queue of last caller, when cause is ACDRemoteCallQueued.
Event Device Data (ACD Agent)	Identity of agent to whom the call is distributed (ACDRemoteCallDistributed only)
EventClass	ACD2PathEventClass.
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.

Events in Advanced Conference Event Class

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

- Conference Feature

ConferenceFeatureEvent

Description

This event is provided by a Conference Monitor when a conference is started, or a Conference Monitor is set when a conference is already in progress, or any change takes place affecting the composition of the conference. Note that for the first two causes, which indicate “initial” events, all of the conference party members are reported, but for the other events, only the member(s) involved in a change to the conference are reported. For example, when a new member is

added to an existing three-party conference to form a four-party conference, only the new member is reported in the associated conference event.

A conference member can be one of the following three device types:

- phone
- attendant
- trunk

In Conference Feature monitor events, phones and attendants are represented by directory number (DN), and trunks are represented by trunk number.

Conference Feature monitor event reporting does not include the Call Reference IDs of the conference members.

- If a conference member is a phone and has a name programmed, the phone's name is not reported.
- If a conference member is a trunk and has a trunk label programmed, the trunk label is not reported.
- If a conference member is an MSDN or CO trunk, it can have additional information included in feature events. If an MSDN trunk has a remote party, then the remote party's DN is also reported. If a CO trunk has CLID, then its CLID is also reported.

A conference can be composed of one or more remote conferences. A remote conference is a conference that is set up in a remote switch. When a remote conference is involved, only one member of the remote conference is reported by conference feature monitor events from local switch. To receive all members of a remote conference, your application needs to enable a conference feature monitor on the remote switch.

Existing device monitors generate conference call events. These call events will provide information similar to conference feature monitor events. We recommend that no suppression be done for call events reporting when conference feature event reporting is enabled. This allows device-monitoring applications and feature-monitoring applications to co-exist.

Event Data

DATA	DESCRIPTION
Event Type	ConferenceFeatureEvent
Event State	EstablishedState.
Event Cause	The cause will be one of: ConferenceMonitorSet (Party List contains info for all party members) ConferenceEstablished (Party List contains info for all party members) ConferenceMemberAdded (Only the added party member info provided) ConferenceMemberDropped (Only the dropped party info provided) ConferenceSplit (Only the Conference Controller info and the remaining party connected info are provided) ConferenceConsHeld (Only the holding party info provided) ConferenceConsRetrieve (Only the retrieving party info provided) ConferenceCallHeld (Only the holding party info provided)

DATA	DESCRIPTION
	ConferenceCallRetrieved (Only the retrieving party info provided) ConferenceTransferred (Only the transferring party info provided) ConferenceRemoteUpdated (Only the updated party info provided).
Event Attribute (Conference Call ID)	Conference Call ID for this monitor in the conference.
Event Attribute (Total Number of Members)	Returns the number of members. Except for the initial events, this number is not the same as the number of parties involved in the conference; To obtain the total number of members in a conference, the application must store the number (Total Number of Members) from the initial event (ConferenceMonitorSet or ConferenceEstablished), then increment or decrement when parties are added or dropped. When the total number of members in the conference is reduced to two, then that conference ceases to exist, and no further conference events will be reported.
Event Attribute (Device Type for a member)	The type of a conference member (for example, VOICE_SET, PRIVATE_TRUNK).
Event Attribute (Device Number for a member)	The DN of a conference member.
Event Attribute (Conference Call ID for a member)	The Conference Call ID that may be reported for this call by a monitor set on another member. This allows applications to correlate calls seen by different monitors. A returned value of zero indicates that no call reference number has been reported by the MiVoice Business.
Event Attribute (Remote member Extension)	Directory number of a remote conference member.

Notes

1. The ConferenceRemoteUpdate cause may be reported when a remote conference member transfers to a different extension.
2. Conferences created as a result of Executive Override feature or Silent Monitor feature do not generate conference feature events.

Call Control Service States (for both Std. and Adv.)

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 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	<p>A problem has occurred while trying to process user's request. Explanations for this failure include (among others):</p> <ul style="list-style-type: none">The user has dialed a busy extensionThe user has been off-hook too long without dialingThe user has dialed a phone that has activated do-not-disturbThe user has dialed a non-existent number <p>The target has violated class of service, class of restriction, or interconnect privileges in the call attempt.</p> <p>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 MiVoice Business resources.</p>
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).
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 MiVoice Business 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 MiVoice Business maintenance.

Call Control Service 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. Some of the event causes below only apply to Advanced Call Control Service (i.e., event causes beginning with ACD).

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.
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.
ACD Call Overflowed	The ACD call to the path overflowed to another ACD group.
ACDCallDelivered	<p>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.</p> <p>This cause is reported from the perspective of the ACD structure being monitored. It occurs only for ACD structure monitors or ACD2 path monitors.</p>
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.

EVENT CAUSE	DESCRIPTION
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. This cause is reported only for ACD2 path monitors.
ACDRemoteCallQueued	This cause is reported in an event delivered to a remote agent group monitor when no agent is available.
ACDRemoteCallDequeued	This cause is reported in an event delivered to a remote agent group monitor when an agent becomes available.
ACDRemoteCallDistributed	This cause is reported in an event delivered to a remote agent group monitor when an agent is ringing.
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.
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.

EVENT CAUSE	DESCRIPTION
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	<p>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.</p> <p>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.</p>
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	<p>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.</p> <p>See also: ACDSilentMonitorEstablished, ACDSilentMonitor and Conferenced call causes.</p>
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.
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

EVENT CAUSE	DESCRIPTION
	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	<p>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.</p> <p>When this device is alerted an event containing CallbackMatured cause is received by this device.</p>
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
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

EVENT CAUSE	DESCRIPTION
	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: ConferenceMonitor() has just been invoked for a MiVoice Business, 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	This cause is reported in an event delivered to a device that initiates the consultation hold.
ControllerCommsFailed	Communication with the device's MiVoice Business has been broken.
ControllerCommsRestored	This cause will be reported to when communication with a MiVoice Business 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.

EVENT CAUSE	DESCRIPTION
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.
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

EVENT CAUSE	DESCRIPTION
	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	This cause is reported in an event delivered to a device that retrieves a call on hard hold.
Intrusion	<p>This party and another party are engaged in a call, and a third party has intruded into that call. The MiVoice Business feature which allowed this is called executive over-ride (also barge). The result is a conference call between all three parties.</p> <p>Use of the executive override feature requires that the intruder have special privileges assigned to him or her on the MiVoice Business.</p>
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	This event cause is reported to a monitored group.
MonitorSet	A device monitor was set against this device. This cause is returned in a MonitorSetEvent to the invoker of the monitor operation.
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.
NewCallInvoked	This device is initiating a new call.
NoAnswer	This device was attempting to make an outgoing call.

EVENT CAUSE	DESCRIPTION
	The call failed because the destination device did not answer within the MiVoice Business 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 MiVoice Business call pickup feature.
PickupInvoked	This device answered a call which was ringing at another device by using the MiVoice Business 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 (clearCall).
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	A MiVoice Business acting as a Secondary MiVoice Business for a Resilient device has determined that communication with the primary MiVoice Business has been re-established, and registration for the device has been passed back to that MiVoice Business.
RedirectedOnError	The MiVoice Business 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 MiVoice Business 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.
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 MiVoice Business.
RoutingDeviceCallQueued	This happens when a new call arrives and queues on a routing device.
RoutingDeviceCallRerouted	If a call is rerouted by MiVoice Business system rerouting options (due to always_reroute or reroute-on-DND) from a routing device to a new destination programmed in the MiVoice Business.
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	This event cause is reported to a monitored HCI routing device indicating status of call queued to the routing device.
RoutingDeviceCFNOANSCallQueued	This event cause is reported to a monitored HCI routing device indicating status of call queued to the routing device.
RoutingDeviceCFALWAYSCallQueued	This event cause is reported to a monitored HCI routing device indicating status of call queued to the routing device.
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 MiVoice Business controller or its secondary MiVoice Business

EVENT CAUSE	DESCRIPTION
	controller(s).
SecondaryMonitorSet	<p>ControllerCommsFailed – Communication with the device's primary or "home" MiVoice Business has been broken.</p> <p>SecondaryMonitorSet – Communication with the device's secondary MiVoice Business is up, and the monitor on the device is now established there.</p> <p>SecondaryControllerCommsFailed – There is no communication with either the primary MiVoice Business controller or secondary MiVoice Business controller(s).</p> <p>RedirectedHandoff – A MiVoice Business acting as a Secondary MiVoice Business for the device has determined that communication with the primary MiVoice Business has been re-established, and registration for the device has been passed back to that MiVoice Business.</p>
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 tapCall() 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 outputting 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	This event cause is reported to a monitored phone.

EVENT CAUSE	DESCRIPTION
CallAnswered	This event cause is reported to a monitored phone.
CallDisconnected	This event cause is reported to a monitored phone.
CallOverflowed	This event cause is reported to a monitored phone.
CallWaitingPartyDisconnected	This event cause is reported to a monitored phone.
PeerOutOfService	This event cause is reported to a monitored SIP Trunk.
PeerInService	This event cause is reported to a monitored SIP Trunk.
EhduAccess	This event cause is reported to a monitored EHDU in a MiVoice Business.
EhduCallback	This event cause is reported to a monitored EHDU in a MiVoice Business.

Call Control Service Event Data

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.
- **Class-Specific Data** – available for all events of a specific event class. For detailed information, see Class-Specific Data.
- **Event-Specific Data** - available and valid only for certain events types. For detailed information, see Event-Specific Data.

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 (Name and/or number may be suppressed by MiVoice Business Class of Service settings). TRUE (1) indicates that Privacy is "on" for that device.

DATA	DESCRIPTION
Conference Member Name Privacy	Privacy status of a member of a Conference call. (Note: Name and/or number may be suppressed by MiVoice Business 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	<p>Identifies the call that was affected. In the case of a ConferenceFeatureEvent, this is a number assigned by the MiVoice Business to track a particular conference.</p> <p>Note 1: Call events reported by a monitor on an IP RAD do not provide Call ID information.</p> <p>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 MiVoice Business is a MiVoice Business), however, and applications which depend on Call ID will need to handle them:</p> <p><i>Hard Hold:</i> 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, where device 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.</p> <p><i>ACD Re-queuing:</i> 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 gets re-queued 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, where device points to the other device. If the Call ID for the other device is unchanged, then it is the same call.</p> <p>Hard Hold and ACD re-queuing 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.</p>

DATA	DESCRIPTION
GlobalCallId	A system-wide unique call ID that can be used to track a call through the MiVoice Business 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 MiVoice Business, 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.
EventState	Indicates the current state of feature, call, or device. The possible states are dependent 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: CancelACDWorkTimer AlternateCall AnswerCall AssignCallerId CamponCall CancelConsultationCall ClearCall NewCall ConferenceCall ConsultationCall HoldCall IntrudeCall ListenCall OutputPulseDigits 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

- EventDeviceNumber

EventFeature

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 (adv)	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 (adv)	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.
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	Member has joined a group.
LeaveGroup	Member has left a group.
JoinAllAcdGroups (Adv)	ACD agent has joined all ACD groups on a MiVoice Business where the agent is configured.
LeaveAllAcdGroups (Adv)	ACD agent has left all ACD groups on a MiVoice Business where the agent is configured.
AcdAgentBlocked (Adv)	Agent login is blocked.

Acd2LoginNoLicenses (Adv)	MiVoice Business has no more ACD licenses.
Activate	Feature has been enabled.
Forward	Call has been forwarded.
UserId	Login ID for a user.
HotDeskUser	DN of hot desk user.
Conference	Conference has been created.

EventState and EventDeviceNumber

DATA	DESCRIPTION
EventState	The current state of 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, this data is provided in the CallReceivedEvent. The Mitel OIG WSDL defines what data is provided in events.

Design and Error Handling Guidelines

This section provides guidelines on how to develop applications for the Mitel OIG. The next few paragraphs provide some high level guidance while the Topics table below refers to specific guidelines related to specific topics.

A Mitel OIG Application should do the following:

1. Login to the Mitel OIG and store the sessionId for the communication session created.
2. Ensure the communication session is kept open using the Mitel OIG keep alive operation.
3. Get the ICP id for all MiVoice Business instances in the MiVoice Business system cluster associated with the Mitel OIG.
4. Get the object id for all devices on a specific MiVoice Business that the application is interested in.
5. Repeat for each MiVoice Business in the MiVoice Business system cluster.
6. Start a monitor on each device using the associated object id retrieved from the Mitel OIG.
7. Start a background thread which will loop calling "getEvent" for a specific MiVoice Business.

This event polling thread will log the MiVoice Business events retrieved and store device information for each monitored device.

8. The application can then start calling more operations on the monitored devices (i.e., make call on a phone).
9. Finally, create shutdown logic which does the following:
 - Stops all monitors on all MiVoice Business instances
 - Terminates all event polling threads for all MiVoice Business instances
 - Logout from the Mitel OIG

The Mitel OIG application must:

1. Wait for operation success and any associated status events before sending more operations to the Mitel OIG.
2. Check for errors and take appropriate recovery action (i.e., resend the operation).
3. Do not stop monitors after each operation. Once the monitor is stopped the application will no longer get any status events for that device.
4. When a call is made or answered, the call needs to be cleared (disconnected) at some point, otherwise the phone will not return to idle and will not be ready to make or receive another call.



Note: Connected device information in Call Cleared event - When executing PSTN trunk calls (MiVoice Business call control is dependent on PSTN trunk signals) and outside caller clears the call, the call cleared event for a local device with cause cleared will have missing connected device. There are several other call scenarios where a monitored device is on one MiVoice Business and the device clearing the call is on another device, and the call cleared event is missing the connected device information.

For Call Cleared event to parties in a conference involving an EHDU the developer must consider:

1. In the case where a local member and EHDU are in a conference, the call clear event to local member does not have device information on the party are cleared the call.
2. Sometimes even in the case where a local member clears down a conference, the call clear events to other local members will not have device information on the local member that cleared the call.

For a Call Cleared event to remote member of Ring Group the developer must consider:

1. In the case where a local member answers a call to a Ring Group, the call clear event to remote DN (member of Ring Group) does not have device information on the local member that answered.
2. Sometimes even in the case where a local member answers a call to a Ring Group, the call clear events to other local DNs (members of Ring Group) will not have device information on the local member that answered.

TOPICS

Call Control Service Error Definition

Call Forwarding incorrect event data

Session Communication Security

ANI and DNIS Information in Events

Call Flow Exceptions for Call Park and Paging

Monitor All ACD Agents in One MiVoice Business

DTMF Generation

DTS Programmed on a Phone/Key Button (Advanced)

Trunk Monitor Events

Device Data

Identifier for Phone Non-prime Line

Monitoring IP Trunks

Monitoring Limits

Redirecting Call on Line Appearance

Missing Called Number in SIP Peer Monitor Events

IP RADs (Recorded Announcement Devices)

Call Received Event to RAD

Silent Monitoring an Agent

Calling Multicall Groups

Identity of Physical Device Answering a Call

MiVoice Business DPNSS and Feature Indications

General Notes

Call Control Service Error Definition

Mitel OIG operations return success or an error on failure. The Mitel OIG errors are organized 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 MiVoice Business. Some communication errors are recoverable and some are not. For recoverable communication errors, the application should retry the operation. **Operation-specific** errors are those relating to the failure of an individual operation. These errors include attempts to monitor or manipulate phones unknown to a MiVoice Business, and attempts to perform operations that are not valid. Typically, these errors indicate an application programming error.

The Mitel OIG pre-appends error messages with an ICP object ID. For example, an error message in Mitel OIG 1.2 looks like : **'MCD: 16 Answer call failed with error: FEATURE_NOT_ALLOWED'** where 16 is the ICP object ID of the MiVoice Business that generated the error message.

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 MiVoice Business.
FAIL	A generic error was returned by the MiVoice Business.
UNSPECIFIED	An unspecified error has occurred. The Mitel OIG did not receive error information from the MiVoice Business.

Communication Errors

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

ERROR	DESCRIPTION
CANNOT_CONNECT_SERVICE	The Mitel OIG was unable to connect to a MiVoice Business. This error indicates that the MiVoice Business was not found or the network is not configured correctly.
SERVER_UNAVAILABLE	When opening a connection to a MiVoice Business, the Mitel OIG found that the MiVoice Business was not started.
SERVER_NOT_READY	In opening the connection to the MiVoice Business, the MiVoice Business indicated call processing was not yet available.
SERVICE_FULL	The application was unable to connect to a Mitel OIG. This error indicates that the Mitel 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 objects, and attempts to perform operations which are not valid.

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 MiVoice Business.
Failed to open connection to ICP at:	The application was unable to connect to the MiVoice Business because the MiVoice Business 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 a Mitel 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 or is 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.

ERROR	DESCRIPTION
FEATURE_NOT_ALLOWED	Attempted to manipulate a held call which does not exist.
MiVoice Business_TIMEOUT_ERROR	The MiVoice Business did not respond to a Mitel OIG operation request in the allowed time.
PRIVILEGE_VIOLATION	Invoker does not have sufficient privileges.

Call Forwarding incorrect event data

Incorrect event data can occur in case of CALL_DELIVERED AND CALL_ESTABLISHED event while call forwarding over SIP Trunk.

When forwarding a call (CF always, CF busy external, CF No Answer External, CALL_Reroute scenario) over an MiVoice Business SIP trunk, incorrect event data can be received in Call_Established event on Device A and Trunk 1 and Call_Delivered event on Device A. If the MiVoice Business SIP trunk is properly configured using public SIP trunk config, then the required event data is provided in the call scenario identified. The Assert P header and allow display update must be enabled in SIP peer signaling form. See the MiVoice Business ESM web browser UI on-line help for the MiVoice Business related to CPN for calling party number and Call Billing for SIP Gateway. Please see brief summary below. For complete configuration details use MiVoice Business on-line help.

Calling Party Number (CPN) Substitution

Calling Party Number (CPN) substitution is typically used to show the customer's corporate name and number for all outgoing calls to the public network. The MiVoice Business provides two types of CPN substitution: DID-based and DN-based. DID-based substitution is used for individual DID numbers or ranges of DID numbers and applies to calls over embedded PRI/DPNSS trunks, embedded BRI trunks ("T" interface only), and SIP trunks. DN-based substitution is done at the directory number level and applies to calls over embedded PRI/DPNSS trunks and SIP trunks. You can program both types of substitution on the same system, or within a cluster or network, with priority given to DN-based substitution. When using DN-based CPN substitution in a network with multiple public network gateways, the gateways must all connect to Central Offices that allow the programmed calling party numbers. The CPN programmed for DN-based substitution should allow uncompleted calls to return to the call originator or to some other number such as the attendant.

The following rules apply to CPN delivery to the public network for forwarded/rerouted calls:

- When an internal call from an extension (Party A) which is provisioned with DN-based CPN is call forwarded to the public network via another Class Of Restriction (COR) extension (Party B), the CPN Substitution programmed for Party A will be passed to the public network not the CPN Substitution programmed for Party B. If Party A has both DID-based and DN-based CPN substitution programmed, the DID-based substitution will be followed.
- When Party A transfers an external call— either from a public or private PRI trunk— to the public network, the DN-based CPN Substitution for Party A will be used if "Display Held Call ID on Transfer" is disabled in its Class of Service. If the option is enabled, then CPN number received from the external trunk will be passed to the public network.

You have the option to configure a unique Default_CPN and Default_Billing_Number for each zone in a cluster. For all SIP trunk calls, if there is no public number (CPN Substitution digits) configured in the Associated Directory Numbers form, the MIVOICE BUSINESS will send these default numbers to a gateway to identify the calling or diverting party and to provide the correct billing number. For more information, see Call Billing for SIP Gateway.

Session Communication Security

Applications use HTTPS to communicate with the Mitel OIG. The Mitel OIG uses a Secure Socket Layer (SSL) to communicate with a MiVoice Business.

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. Mitel 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 phone A on MiVoice Business A to phone B on MiVoice Business B. Phone B transfers incoming call to phone C on MiVoice Business B. Events are for phone C on MiVoice Business B.

Events:

1. CallEstablishedEvent / NewCall / Nil
2. CallEstablishedEvent / NewCall / TransferredCall
3. CallRetrievedEvent / AnswerInvoked / TwoPartyCall

ANI is reported in events 2 and 3 above.

Example 2: Remote Transfer (Supervised)

Scenario: Call from phone A on MiVoice Business A to phone B on MiVoice Business B. Phone B transfers incoming call to phone C on MiVoice Business B. Events are for phone C on MiVoice Business B.

Events:

1. CallEstablishedEvent / NewCall / Nil
2. CallRetrievedEvent / AnswerInvoked / TwoPartyCall

NO ANI is reported in this scenario.

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

Scenario: Call is made to phone A in MiVoice Business X. Phone A transfers the call to phone B on MiVoice Business Y. Phone B then transfers the call to phone C on MiVoice Business X. Events are for phone C on MiVoice Business X.

Events:

1. CallEstablishedEvent / NewCall / Nil
2. CallRetrievedEvent / AnswerInvoked / TwoPartyCall
3. CallRetrievedEvent / Transfer / TransferredCall

ANI is reported for event 3 as main DN.

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

Scenario: Call is made to phone A in MiVoice Business X. Phone A transfers the call to phone B on MiVoice Business Y. Phone B then transfers the call to phone C on MiVoice Business X. Events are for phone C on MiVoice Business X.

Events:

1. CallEstablishedEvent / NewCall / Nil
2. CallEstablishedEvent / NewCall / TransferredCall
3. CallRetrievedEvent / AnswerInvoked / TwoPartyCall
4. CallRetrievedEvent / Transfer / TransferredCall

ANI is reported for events 2 and 3 as sub DN.

ANI is reported for event 4 as main DN.

Call Flow Exceptions for Call Park and Paging

Call Park functionality, like Paging, is accessed by dialing MiVoice Business Feature Access Codes. The Mitel OIG currently has no special support for these functions. Like other dialable strings, the Feature Access Codes can be invoked using `makeCall()`. Whether these functions are invoked by `makeCall()`, or by dialing from a phone, the event sequences (call flows) that result are different from the usual event call sequences.

Applications that must function in an environment where Call Park and Paging functionalities are in use should be aware of the differences in call flows. See Appendix D (Call Flow Exceptions for Call Park and Paging).

Monitor All ACD Agents in One MiVoice Business

The Mitel OIG allows an application to create a class level monitor (`advMonitorACDAgent` Features) on a specified class of objects on a specific MiVoice Business. Thereafter, the application is notified of changes in feature programming for this class by means of feature events. Two device classes can be monitored:

- report feature events for all trunks on one MiVoice Business

- report feature events for all ACD agents on one MiVoice Business

Monitoring feature events using `advMonitorACDAgent` is more efficient than a Feature Monitor on a specific agent.

DTMF Generation

An application CANNOT use `outPulseDigits` to send DTMF digits to IP phones. The MiVoice Business performs the outpulsing, but only to trunks and DNI connected phones.

DTS Programmed on a Phone Key/Button (Advanced)

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

Trunk Monitor Events

Events reported from trunk monitors may be best understood by considering the trunk in the MiVoice Business as a local representation of the remote device (usually a phone) to which the call is connected. Consequently, for a local phone originating an external call via a trunk, the trunk is considered to be a terminator as though it were the final destination. A monitor set on the trunk will generally first report a `CallReceived` event. This event will have calling device information that used to determine the originating local extension. If the remote phone answers there will be a `CallEstablished` event. This event will have connected device information that will provide the originating local extension.



Note: In some cases, a monitor set on a trunk may not report a `CallReceived` event. Analog trunks, for example, do not provide sufficient information to decide when the far end is connected.



Note: Trunk monitors do not support trunk resiliency.

Using the trunk model as a local representation of the remote phone, an incoming call makes the trunk the originator of a local call. Thus, a monitor set on the trunk will report a `CallOriginated` event (unless it is a non-dial-in trunk), and then a `CallDelivered` event (in which the called device is the local phone), and then (if the local phone answers) a `CallEstablished` event.



Note: Not all events may be reported for some scenarios, depending upon the type of trunk:

A successful outgoing call via an analog trunk will miss the `CallReceived` event and only begin reporting with the `CallEstablished` event.

Outgoing unsuccessful calls show the most differences, since what is reported depends on what information the trunk is able to obtain and how that information is handled. For example, an outgoing external call to a remote phone that is busy will almost immediately yield a `CallCleared` event for a DPNSS or a DASS2 trunk. A `CallReceived` event for other digital trunks (including

ISDN) makes available the trunk type as soon as a monitor is established on a trunk. This information enables the application to better interpret the different events in these scenarios.

Incoming calls may also generate different events depending on trunk type in certain scenarios. For example, an incoming call to a busy phone will normally cause the trunk monitor to report a `CallFailed` event, but an LS/GS trunk will camp on to the busy DN and a `CallQueued` event will be reported instead of the `CallFailed`.

The clearing down of the far end of an external call normally generates a `DeviceCleared` event. For ISDN trunks in this scenario, the trunk is not released automatically but waits for the local phone to clear. Local clearing will always generate the `DeviceDropped` event, since the `DeviceCleared` event must wait for the remote switch or CO to release the trunk.

Knowledge of the operation of different trunk types will enable you to handle most scenarios; experimentation is recommended for any uncertain situations that you want to monitor.

Device Data

The Mitel OIG events contain device data related to a phone. 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 MiVoice Business. 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 MiVoice Business. If available, DNIS is provided in both `CallReceivedEvent` and `CallEstablishedEvent`.

Device information includes:

- Device Name
- Device Number
- Device TrunkNumber
- Device TrunkName
- Device ANI
- Device DNIS
- Device Domain
- Device NetworkExt

The Mitel OIG events also include Device domain information:

- Internal DeviceType
- External DeviceType
- Events also provide:
 - First Name
 - Last Name
 - Middle Name
 - Main Dn

- Sub Dn

Identifier for Phone Non-prime Line

When a network call is answered by a monitored phone on a local MiVoice Business, the event information reported is dependent on the MiVoice Business 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 cannot be monitored. 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 MiVoice Business 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 MiVoice Business COS option violation.

Missing Called Number in SIP Peer Monitor Events

Scenario: Called Number missing in Call_Delivered or Call Established events over SIP trunk on MiVoice Business 6.0 SP1.

Have phone 1000 on MiVoice Business A call phone 2000 on MiVoice Business B over SIP trunk between MiVoice Business instances. When the call is received on 2000 at ICP B, the number of calling party 1000 may not be provided to SIP trunk monitor on MiVoice Business B. Ensure MiVoice Business instances are configured with:

- SIP Peer Profile: "Use P-Asserted Identity Header" = YES
- SIP Peer Profile: "Use P-Asserted Identity for Billing" = No



Note: Billing assert defaults to TRUE. This assert was added in MCD 6.0 SP1. If you do not have a default programmed (billing number), the MCD/MiVoice Business will NOT send the called number identity.

Device Attributes in Call Clear Event

Call cleared events on devices only provide connected device. Calling device or called device does not make sense when a call is being cleared. The purpose of the event is to indicate that a call is being cleared and to indicate the device that is being cleared on the other end of the call.

IP RADs (Recorded Announcement Devices)

IP RAD devices provide Recorded Announcement Device (RAD) functionality. If, for example, an IP RAD device is programmed on the MiVoice Business as the RAD for an ACD Path, then the IP RAD will receive an inbound call when all agents are busy and none become available within the time specified by MiVoice Business programming. Alternatively, it can be called directly, have calls forwarded to it, or have calls redirected to it by an application.

Operation

A call reaches the IP RAD device, and the monitor on it reports a `CallReceivedEvent` (`ReceivedState`, cause `NewCall`). Note that the calling device is not reported. The call is still queued or ringing at the original destination device (for example, an ACD Path).

If additional calls reach the IP RAD while it is still in the `ReceivedState`, the monitor does not report any additional `CallReceivedEvents`. However, as soon as an application commands the IP RAD to answer, the call goes into the `EstablishedState` with all waiting parties listening to the message. An application can use Mitel OIG to determine the number of callers listening.

The event reported when the call goes into `EstablishedState` is `ACD2PathEvent` (since this is the most common use of the RAD), and the cause is `ACDRadStarted`.

If the IP RAD is never commanded by an application to answer the call received by the RAD, then eventually a timer in the MiVoice Business will expire, and the IP RAD will go to `IdleState`, with `DoNotDisturb` also being set for the device. An application should have a feature monitor as well as a device monitor set on the IP RAD, so that it can detect this occurrence and put the IP RAD back into service, if desired (cancel DND).

There are several other ways that the IP RAD may be returned to the `Idle` state, but none of these put the RAD into DND. These other ways are:

- the caller(s) hangs up before or after the RAD answers, or
- the RAD plays to completion and is disconnected, or
- an application tells the IP RAD to end (Clear) the call.

A Call Received Event is generated when a RAD in a RAD Hunt Group receives a call. The RAD Hunt Group DN is provided in the Call Received Event to allow an application controlling a RAD soft phone to decide what message to play to the queued caller.

Call Received Event to RAD

A Call Received Event is generated when a RAD in a RAD Hunt Group receives a call. The RAD Hunt Group DN is provided in the Call Received Event to allow an application controlling a RAD soft phone to decide what message to play to the queued caller. An application can extract the RAD Hunt Group DN from a Call Received Event.

Silent Monitoring an Agent

The Silent Monitoring of an agent call by a supervisor may be initiated either by the agent pressing a softkey requesting help, or by the supervisor triggering silent monitoring (MiVoice Business 6.0 SP3 provides different silent monitoring behavior compared to MiVoice Business 7.0), depending on the options enabled in the MiVoice Business.

In MiVoice Business 7.0, bridge feature events are generated when silent monitoring has been turned on.

A “BRIDGE_JOIN” event is generated when a phone or application sets a silent monitor on an ACD hot desk agent or telephone. This event will be returned to all advanced applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_LEAVE” event is generated when a phone or application clears a silent monitor on an ACD hot desk agent or telephone. This event will be returned to all advanced applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_CONNECTED” event is generated when the first application invokes the “Silent monitor” feature on an agent or telephone. This event will be returned to all advanced applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring the phone/agents device features.

The “BRIDGE_DISCONNECTED” event is generated when all phones/supervisors clear their silent monitors on an agent or telephone. This event will be returned to all advanced applications which either have set the “advMonitorAllBridges” on the specified MiVoice Business or are monitoring the phone/agents device features.

In MCD 6.0 SP3, a CallConferencedEvent is generated to each of the three monitored parties (agent, caller, and supervisor) involved. A conference party list is also provided to applications monitoring the parties (agent, caller, and supervisor). The silently monitored device (agent phone) will be the last item in the party list.

If more than two members and a supervisor are included in the conference, when someone other than the supervisor leaves the conference, a cause of ConferenceMemberDropped or ACDSilentMonitorMemberDropped is reported, irrespective of which party actually dropped. These causes are considered to be equivalent.

It should be noted that there are restrictions on the support for Silent Monitor conference reporting.

- Silent Monitoring of a member in a two-party call is supported. Silent Monitoring of a member in a three or more party call is not supported. Events may occur, but their presence, content, and consistency with ordinary conference events cannot be guaranteed.
- When any member drops out of a Silent Monitored three-party conference, a DeviceDroppedEvent will be reported, with a cause of ConferenceMemberDropped or ACDSilentMonitorMemberDropped. These causes are equivalent and party list information must be used to determine whether a member or the supervisor dropped from the conference.
- Redundant (duplicate) events will occur when extra members join a Silent Monitored three-party conference.

Calling Multicall Groups

When a call is directed to a DN that specifies a multicall group, all of the idle members of the group report a CallReceivedEvent. If one member answers the call before another call comes, then a monitor on that member will report a CallEstablishedEvent, while monitors on the others will report CallClearedEvents. If one or more additional calls arrive before any member answers that first call, when one member answers the first call, the other members all receive a CallClearedEvent followed immediately by a CallReceivedEvent, which carries information about the identity of the next caller.

Identity of Physical Device Answering a Call

When a monitored device calls a Multicall Group or a Keyline, only one member can answer. The answering party (ConnectedDevice) reported to the calling device monitor will be the Prime DN of the set which actually answers. The application knows which member actually answered the call. Subsequent events will also carry the Prime DN information.

MiVoice Business DPNSS and Feature Indications

An MiVoice Business trunk with dpnss sends a transfer feature indication between MiVoice Business nodes. Using this transfer feature indication at the remote MiVoice Business node, events are generated in addition to the RemotePartyUpdated and GlobaCallIDUpdated events (new CallEstablishedEvent EstablishedState SupervisedTransfer and new CallEstablishedEvent EstablishedState UnsupervisedTransfer). The example below shows the events when the feature invoker is a remote phone 4005 (supervised transfer).

Scenario:

- Phone 4006 and 4007 reside on MiVoice Business IP address 10.112.95.41
- Phone 4005 resides on MiVoice Business IP address 10.112.95.39
- Phone 4006 calls 4005
- Phone 4005 answers the call and puts 4006 on consultation hold.
- Phone 4005 calls 4007.
- Phone 4005 invokes transfer (supervised transfer)

- Phones 4006 and 4007 get connected in a call.

Below are the new events generated, in addition to existing globalcallidupdated and remotepartyupdated events.

EVENT: 4006 RemotePartyUpdateEvent EstablishedState GlobalCallIDUpdated
Connected:4007 SuitePilotNumber=(null)((null)) LineNumber=(null)
PersonalRingGroup=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:E50E2C00000000011F4D
SecGlobalCallId:null

EVENT: 4006 CallEstablishedEvent EstablishedState SupervisedTransfer
Connected:4007 SPN=(null)((null)) LN=(null) PRG=(null)((null)) EHDU=((null));
Ctrl: 4005 SPN=(null)((null)) LN=(null) PRG=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:E50E2C00000000011F4D
SecGlobalCallId:null

EVENT: 4006 RemotePartyUpdateEvent EstablishedState RemotePartyUpdated
Connected: 4007 SuitePilotNumber=(null)((null)) LineNumber=(null)
PersonalRingGroup=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:null SecGlobalCallId:null

EVENT: 4007 RemotePartyUpdateEvent EstablishedState GlobalCallIDUpdated
Connected: 4005 SuitePilotNumber=(null)((null)) LineNumber=(null)
PersonalRingGroup=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:D2A73400000000011D80
SecGlobalCallId:null

EVENT: 4007 CallEstablishedEvent EstablishedState SupervisedTransfer
Connected:4006 SPN=(null)((null)) LN=(null) PRG=(null)((null)) EHDU=((null));
Ctrl: 4005 SPN=(null)((null)) LN=(null) PRG=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:D2A73400000000011D80
SecGlobalCallId:null

EVENT: 4007 RemotePartyUpdateEvent EstablishedState RemotePartyUpdated
Connected:4006 SuitePilotNumber=(null)((null)) LineNumber=(null)
PersonalRingGroup=(null)((null)) EHDU=((null))
GlobalCallId:E50E2C00000000012005 PriGlobalCallId:null SecGlobalCallId:null

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 MiVoice Business). 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 MiVoice Business. 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 MiVoice Business.
 - **ACD Re-queuing:** 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 re-queued 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. 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.
6. When a call is auto recall to an ACD Agent the Call Received Event cause is RecallHeldParty.
7. Consider monitoring Hotdesk User 7500 on primary MiVoice Business 1 (with secondary MiVoice Business 2), physical phone 6001 on MiVoice Business 2 (with secondary MiVoice Business 1) and physical phone 6500 on MiVoice Business 1 (with secondary MiVoice Business 2); Application creates a monitor on 7500 on MiVoice Business 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 MiVoice Business instances), 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 logs 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.
8. To distinguish the difference between a "PSTN to phone call" (external trunk on the public network) from a "Remote phone on another MiVoice Business", 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.
9. To allow an application to retrieve group presence on a DN, ensure MiVoice Business COS settings "Group Presence Control" and "Group Presence Third Party Control" are enabled.

10. 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. See the Global Call ID Developer Guide on the MSA web portal for more information.
11. The Mitel OIG presents different event flow when monitoring devices on one MiVoice Business compared to monitoring devices on different MiVoice Business instances in the same system. Example, the events presented at the held party for a local recall scenario (i.e., holder gets recalled when both holder and held party on same MiVoice Business).
- 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 held party for a remote recall scenario (i.e., holder gets recalled when holder and held party on different MiVoice Business instances).

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 held party for a local retrieval scenario (i.e., holder retrieves held party when holder and held party on same MiVoice Business).

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 held party for a remote retrieval scenario (i.e., holder retrieves held party when holder and held party on different MiVoice Business).

Remote Retrieve

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

Mitel OIG Operation Considerations

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

1. **Session** – An application must login to the Mitel 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 MiVoice Business instances connected, overall OIG version, version of each Mitel OIG service).
2. **Event Time Stamps (Future)** – The Mitel OIG CCService events sent to applications are time stamped with two times; 1) Mitel OIG time, and 2) MiVoice Business time.
3. **Opening a MiVoice Business Connection** – An application requests the Mitel OIG to open a connection to a MiVoice Business. If two applications request the same connection to the same MiVoice Business, only one connection will be opened from the OIG to the MiVoice Business.
4. **Closing a MiVoice Business Connection** – An application does not request a OIG to disconnect a MiVoice Business. All successful MiVoice Business connections remain open until the OIG is shut down.
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 *Mitel OIG Installation and Maintenance Guide* for detailed licensing information.
7. **Engineering Limits** – The OIG CCService allows a maximum of 5000 monitors on one MiVoice Business. The OIG CCService allows a total of 50000 monitors for all MiVoice Business instances. Refer to the *Mitel OIG Engineering Guidelines* for more information.
8. **Monitor Multiplexing** – When two or more OIG s request phone monitors on the same phone on one MiVoice Business, only one monitor is created in the MiVoice Business. When two or more applications ask for phone monitors on same phone, the Mitel OIG CCService only requests one monitor from the MiVoice Business. (Setting several phone monitors on the same phone from the same application is not supported).
9. **Device Dropped event** is replaced with Call Cleared Event when a two-party call is cleared over a SIP Trunk. This change was introduced in MCD 6.0 SP1.
Steps to reproduce:
 - a. Device A of ICP A and Device B of ICP B and start monitors.
 - b. SIP Peer A of ICP A and SIP Peer B of ICP B and start monitor.
 - c. Make call from Device A to Device B using ARS.
 - d. Answer call from Device B.
 - e. Clear call from Device A or B.
 - f. Device dropped event no longer received on SIP Peer A monitor when call is cleared.

Hunt Groups, Ring Groups, ACD Groups and Group Presence

Group Presence

Group Presence allows members of call groups (ACD, Hunt, Ring and Personal Ring) to make themselves absent or present in a group. Only members who are present in a group are offered calls to that group. Users can also change the presence status of other users, allowing ACD supervisors, for example, to control which agents are available to take calls from a particular ACD group. Users belonging to multiple groups can control their presence in each group separately through the use of feature keys (one for each group), or by dialing a feature access code (FAC) followed by the group pilot number. ACD agents can join or leave all the ACD groups they belong to by dialing a single FAC. With Personal Ring Groups, users can choose which of their personal answer points are 'Present' or 'Absent.'

Group Definitions

Hunt Groups

Hunt groups let you dial a pilot number and have the call completed to the first idle phone in a group of phones. Any phone within a Hunt Group may be accessed directly by dialing the phone number.

Two types of hunting are provided by the MiVoice Business: circular and terminal. In circular hunting, the search for an idle phone begins at the phone after the last one to receive a call. In terminal hunting, the search for an idle phone begins at the first phone programmed in the Hunt Group. In both cases, the search progresses through all phones in the Hunt Group in a programmed sequence.

If there is a free member in the hunt group, that phone is rung by a caller who accesses the group. If the hunt group is busy, one of the following operations take place:

- An internal caller receives busy tone and has the option of going on-hook, setting a callback, or camping on.
- A Direct In Line caller receives ringback tone and is placed in a first-in, first-out call queue within the MiVoice Business system.
- A DID trunk caller receives busy tone.
- A transferred trunk call receives music or silence.
- The master phone of the group will receive the camp-on tone, and then the first member of the hunt group to become free is connected to the next waiting caller.

Ring Groups

When Ring Groups are programmed, calls ring multiple telephones, stopping when any one of them picks up. You can program a ring group to ring all extensions simultaneously (Ring All) or one at a time (Cascade Ring), starting with the first member of the group and continuing to other members in the order in which they are programmed to the group.

Other features of Ring Groups:

- Calls queue if there is no free member to ring.
- Calls unanswered within a specified period are sent to the group's designated overflow destination.
- Member phones with large displays show both the group and caller name and number when the group is rung. Small-display phones show the ring group only.

Personal Ring Groups

Personal Ring Groups (PRG) are an association of two or more devices for one user under a common Directory Number (DN). The devices ring simultaneously (Ring All) when called. The typical scenario is a person's desktop phone and cell phone twinned together, where the desk phone is considered the prime extension, which is referred to as the “pilot number” or “prime member”. Other group members are referred to as non-prime members.

PRGs provide functionality that is similar to Ring Groups and Hunt Groups, but for a single user. In contrast to Ring Groups, PRGs support Ring All, but not Cascade Ringing and support a maximum of 8 as opposed to 32 members. In contrast to PRGs, Hunt Groups can include Trunks.

PRG programming includes a One Busy/All Busy option that can be turned on or off as required.

The “Handoff” feature for Personal Ring Groups (PRGs) allows calls to be “pushed” or “pulled” between group member devices. A push passes the call to the group, where it can be answered by any other present group device. A pull takes an in-progress call away from another member. To push a call to, or to pull a call from, another member, users configure a Handoff key in the Multiline Set Key Assignment form. Users can program any button, including Programmable Key Module (PKM) buttons, as a Handoff key.

The same key is used to push a call and to pull a call. If the Handoff key is pressed while on a call, the call is pushed. If the PRG member device is idle or listening to dial tone, and there is an in-progress call on another member in the PRG, the call is pulled by pressing the handoff Feature Access Key (FAK).

The voice path to a non-member device remains connected to the member device that initiated the handoff until that device releases the call or another PRG member answers. If the pushing device stays connected until the new device answers, the caller hears a short break in audio when the call is pushed to the new device. If the pushing device releases the call before the new device answers it, the caller hears ringback until another member device answers the call.

When a call is pulled, it connects to the pulling member's set and disconnects from the pulled-from member's set. The caller hears a short break in audio when the call is connected to the new device. The caller does not hear Music-On-Hold. The call is not disconnected from the originating party until it is successfully connected to the party that pulled the call.

If a push goes unanswered, when the line that initiated the handoff becomes available, it rings simultaneously with all other available members. If the call remains unanswered and the pushing member does not release the call, the other PRG members stop ringing when the ring timer expires. If the pushing member does release the call prior to the Call Forward No Answer timer expiration, it is redirected to the Call Forward No Answer destination for the PRG. See the “Group Presence” topic in the Mitel Call Director System Administration Tool Help for more information about presence checking.

If more than one PRG member is in an in-progress call, the programmed order of the members in the group determines which call is pulled. If an in-progress call is not pull-able, the next call is checked to determine if it can be pulled. For example, if member two is on one call, and member three is on another call, the member-two call is pulled first. However, if the member-two call cannot be pulled (for example, if the call is on Hold), then the call on member three is pulled.

A PRG pilot (prime member) can be a member of a Ring Group. When a call comes into the Ring Group and the PRG pilot is rung, all available members of the PRG ring. If one of the members in the PRG picks up the Ring Group call, all other ringing members stop ringing, including members in the Ring Group. When the PRG is busy (no available members to ring), the Ring Group call rings all available Ring Group members for Ring All mode, or for Cascade Ring mode the Ring Group call rings the next available member in the Ring Group.

If a Ring Group call rings the PRG pilot but the call is not answered and the caller's ring timer expires, timeout treatment is returned, or the call forwards to the overflow destination of the Ring Group if programmed.

If Call Forwarding or Call Rerouting is active on a PRG and a call comes into the Ring Group and rings the PRG members, forwarding or rerouting is overridden.

If the PRG is busy, the call will:

- for Cascade Ring mode, ring the next available Ring Group member.
- for Ring All mode, ring all available Ring Group members, but not the PRG.
- If the call is not answered by a PRG member, the call will:
 - for Cascade Ring mode, ring the next member in the Ring Group upon expiry of the Cascade Ring timer.
 - for Ring All mode, continue to ring available Ring Group members, including the PRG, until the caller's ring timer expires, at which point timeout treatment is provided.

Groups and Group Presence Monitoring

Ring Group and Hunt Group

Applications can use monitors to receive call events. The group distribution type is also reported. For Ring Groups, the types include new ring group modes: SX_DIST_TERM_CASCADE (Terminal cascade mode), SX_DIST_TERMINAL (Terminal mode), SX_DIST_CIRC_CASCADE (Circular cascade mode), SX_DIST_CIRCULAR (Circular mode), SX_DIST_RING_ALL (:Ring all mode). For Hunt Groups, the distribution type reported is circular or linear.

A GroupEvent contains data that applications can use to determine whether a call is distributed to a member, disconnected from the Ring/Hunt Group, answered by a Ring/Hunt Group member, overflowed, queued, or abandoned while queued.

Personal Ring Groups

Personal Ring Groups (PRG) can be monitored to differentiate whether a caller has abandoned a call or a PRG member has answered the call.

Group Presence

Group Presence monitoring is supported through the `monitorFeatures` operation. Group presence changes are reported through the join group feature and leave group feature events which contain the following information: group DN, member DN, invoker DN, and group presence status (either “join” or “leave”). The `getACDGroupStatus` operation allows applications to query group presence information for ACD group members.

Use `setGroupPresence` to set a group member absent or present in a particular group (ring group or hunt group)

Use `setACDAgent Presence` and `setACDAgentPresenceAllGroups` to set an ACD agent absent or present in all configured Agent Skill Groups.

Personal Ring Groups Presence

The Mitel OIG supports Personal Ring Group (PRG) Presence monitoring on a MiVoice Business with an operation called `monitorPRGPresence`. Applications that need to monitor all PRG presence changes can set up one PRG Presence monitor to receive changes on all PRGs. Invoking this function will report all presence activities of Personal Ring Groups through the join and leave group presence feature events.

External Hot Desk User (EHDU)

External Hot Desking extends hotdesking capabilities to an external device, which makes the external device appear as an extension on a MiVoice Business. When the external hot desk user (EHDU) is logged in to the MiVoice Business, if a caller dials the extension number assigned to the EHDU, the MiVoice Business automatically rings the user's external device.

As a MiVoice Business extension, the external device user has access to extension dialing along with select MiVoice Business features and enterprise Calling Line ID (CLID) on all calls originated through the MiVoice Business system. When a call is extended from the MiVoice Business to an EHDU, the internal state of the user is updated to reflect the busy condition. Since the state of the user is implied, an EHDU can be monitored by DSS/BLF keys on other devices. Call diversion features and group features, such as hunt groups, follow the normal treatments for busy, out of service, etc. For example, if an EHDU is a member of hunt group, and is active on a call, the hunting algorithm will skip the busy EHDU and try the next member.

External Hot Desking provides support for the following additional capabilities:

- the ability to remotely change the external device number as well as the login PIN used for authenticating the user and granting access to MiVoice Business resources.
- access to mid-call features—Hold/Retrieve, Transfer, Conference, Cancel Call.
- the option (via Class of Service) to remain permanently logged in.
- push-to-answer, which requires the EHDU to press a DTMF key to accept an incoming call. (This a Class of Service option used when the trunk terminating on the EHDU device fails to provide answer supervision.)
- group membership, including hunt groups, ring groups, personal ring groups, hospitality suites.
- MiVoice Business IP resiliency for devices.



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

ACD Express Groups

Mitel Automatic Call Distribution (ACD) Express Groups are call answer groups similar to Hunt Groups, and provide the same functionality as an ACD Path, except ACD Express Groups support only one group. ACD Express Groups have limited support for Recorded Announcement Devices, embedded media for music on hold, Interflow, and resiliency.

ACD Express Groups support the following features:

- up to 150 members if the Extended Agent Skill Group option is set to No, and up to 500 members if the option is set to Yes.
- automatic call queue when members are busy.
- call distribution to the longest idle member.
- two recorded announcements using RADs or RAD groups.
- one embedded media source (that is, music on hold) or one external alternate recording device.
- one programmable Interflow point and associated Interflow timeout.
- one programmable Group Unavailable Answer Point.
- ACD Express Groups have the following limitations:
 - a maximum of 64 combined ACD Express Groups and ACD Agent Skill Groups if the Extended Agent Skill Group option is set to No.
 - a maximum of 128 combined ACD Express Groups and ACD Agent Skill Groups if the Extended Agent Skill Group option is set to Yes.
 - a maximum of 999 combined ACD Express Groups and ACD 2000 paths.
 - resiliency is not formally supported, but some MiVoice Business IP resiliency programming is possible. See the MiVoice Business IP Resiliency Guidelines for the steps.



Note: ACD Express Groups contribute to consumption of available MiVoice Business maximums for ACD2 path, ACD2 groups and ACD2 agents.

Hot Desk ACD Express Agents

A Hot Desk ACD Agent becomes a Hot Desk ACD Express Agent on a MiVoice Business when the first group in which it is programmed is an ACD Express Group. If a Hot Desk ACD Agent is not a member of an Express Group, it is considered an ACD2 Agent. ACD2 Agents cannot be members of ACD Express Groups.

Monitoring Support for ACD Express

Hot Desk ACD Express Agents have the same behaviors as Hot Desk ACD Agents, except that Hot Desk ACD Express Agents:

- never consume an ACD license;
- can be members of up to eight different ACD Express Groups;
- cannot be members of ACD2 Agent Skill Groups;
- cannot log in if 100 other ACD Express agents are already logged in. The maximum number of ACD Express Agents that can be logged in at any given time is 100.



Note: The maximum number of agents that can be programmed on a MiVoice Business, whether traditional, Hot Desk, or Express, is 1181.

Interflow

Interflow is a time-based or load-based feature that takes an ACD call out of the path and routes it to the interflow answer point, (if programmed). All calls that interflow lose their priority in the queue.

ACD Express Group Interflow differs from ACD2 Path Interflow as follows:

- A caller's original call destination is not altered by the Interflow.
- A caller's camp-on destination is not altered by the Interflow.

See the “ACD Express Groups – Conditions” topic in the MiVoice Business System Administration Tool Help for more information about Interflow.

Resiliency

The MiVoice Business provides resilient and non-resilient devices. For example, Hot Desk ACD Express Agents can be resilient if the Hot Desk ACD Agent that they are based on is resilient. ACD Express Groups cannot be resilient, but a certain level of resiliency can be obtained. To program resiliency, see the MiVoice Business IP Resiliency Guidelines.

Even though the MiVoice Business 6.0 SP1 provides IP resiliency for some devices, it does not mean that the Mitel OIG provides Resilient monitoring and control for those same devices. The Mitel OIG must also support resilient monitoring and control. For Mitel OIG Release 1.1, resilient monitoring and control is not supported.

External Hot Desk Agent (EHDA)

External Hot Desk Agent (EHDA) support is very similar to what is offered for External Hot Desk Users (EHDU). This feature allows applications to monitor ACD Hot Desk extensions (DNs) that have been associated to external dialable numbers. This allows ACD agents to receive MiVoice Business ACD calls on external devices such as analog and SIP phones, and third party PBX endpoints. Through the phone, an Agent is allowed to login to an external device and use dialed feature access codes to trigger call control features. This capability can also be offered via an application. By monitoring the ACD Hot Desk DN, an application can control calls to that DN, i.e., transfer, conference, etc., and set ACD functions: Make Busy, Group Presence, etc., but has no control over the physical external endpoint itself. The external device is used to manually take the phone off hook to answer and speak and go on hook to end a call.

EHDA External Login

The external device, which typically would be a cell phone, dials an access number and then enters the EHDA DN and PIN to perform a login. The access number need only be provisioned at the PSTN gateway nodes. The access number may be the same on all PSTN gateways and this is recommended. The EHDA user does not need to know a different access number.

The access number can only be dialed by a PSTN call, any attempt to call the access number from internal extensions will be blocked. If the PSTN gateway is running older software and the call is made to the access number across a TDM DPNSS link, the normal login rules will apply.

Should the validation of either EHDA or PIN fail as described below, the caller is given reorder tone and the trunk lockout timer is started, and the current login remains logged in.

The EHDA DN and PIN are to be input as two parameters after the access number provides the initial dial tone. The maximum number of digits that may be entered when performing a login is 15. The first 7 digits would represent the EHDA DN and the remaining 8 would be the PIN.

Upon EHDA DN translate success, dial tone is applied and the collection of the PIN is performed. If the DN is deemed to be invalid then reorder tone is provided and the user must hang up and call the access number again to attempt another login. If successful dial tone is returned the collection of the PIN begins. Upon receipt of the first PIN digit, dial tone is removed and silence applied.

The PIN can only contain digits 0-9, the # DTMF digit will be used to terminate PIN input should the PIN be shorter than 8 digits. Otherwise termination of a PIN that is less than 8 digits will require the inter-digit timer to expire. If the PIN collection is terminated before any digits are collected either by the user entering # or the inter-digit timer expiring then reorder is provided and the user must hang up.

If the validation of the EHDA DN and PIN is successful and the EHDA user is not currently logged in at a MINET location, or busy at either a MINET or external location, then the external location is logged in and confirmation dial tone is returned. At this point the user may make a new call or hang up and remain logged in.

Please note that the external device login can only pre-empt the internal device login if and only if the internal device is in the idle state; otherwise, reorder tone is returned.

When logged in at the external device, all calls directed to the EHDA will be sent to the location defined by the external digit string for the EHDA.

EHDA Behavior Summary

The section summarizes the behavior of monitoring support for EHDA.

External Hot Desk Agent monitoring is the same as External Hot Desk User monitoring with some exceptions.

An External Hot Desk User can be programmed as an ACD Agent. All available ACD agent features shall be provided to an External Hot Desk Agent (EHDA), including:

- EHDA can be programmed in an ACD Express Group or Agent Skills Group, but not both.
- EHDA can be programmed in multiple agent groups of the same type (ACDX or ACD2).

- EHDA can be assigned a skill and follows skills-based routing algorithms.
- EHDA shall be offered an ACD call based on the longest idle, most skilled agent algorithm.
- An EHDA can be programmed as a resilient device.

Each EHDA user logs in/logs out via the following facilities. A user can login to an external device (device not controlled by MiVoice Business) or to an internal device (device controlled by MiVoice Business). An internal device offers some different functions compared to an external device when used for EHDA:

- Mitel OIG login/logout API;
- Calling the Hot Desk Access Number and entering a feature access code;
- When using a hot desk enabled display phone (a type of internal device), existing Superkey and soft key interfaces can be used to login/logout.

When logged in to an external device, the following EHDA features are available via feature access codes:

- Make busy support
- Group presence support
- Do not disturb support
- Logout

When logged into a hot desk enabled display phone (a type of internal device), an EHDA has the same behavior as a logged in Internal Hot Desk Agent, including the following features:

Superkey and soft key interfaces to all ACD and telephony features (make busy, logout, request help, etc.).

- Feature key support (make busy, DND, auto answer, headset, etc.).
- An EHDA can be silent monitored.
- An EHDA can request help via Mitel OIG.
- An EHDA can be auto logged out on no answer.
- An EHDA login will log out another logged in EHDA or EHDU.
- An EHDA can perform all single digit mid-call features, except handoff. An EHDA cannot handoff since it cannot be programmed in a Personal Ring Group.
- To login an EHDA user, an application can use the EHDA number and PIN. The EHDA user can also call the MiVoice Business Hot Desking Access Number and be authenticated via the Call Recognition Service feature.

EHDA actions can be initiated, however, the answering or clearing of a call must be performed manually.

During a call, the EHDA can invoke features by dialing DTMF-based mid-call feature access codes. The EHDA can also call MiVoice Business Hot Desking Access Number, get authenticated, and invoke features via dialing feature access codes.

Once logged in and available, the EHDA can begin to receive ACD calls. Once an ACD call is completed, the EHDA can receive a work timer. Once the work timer expires, the EHDA shall be available to receive another ACD call.

When on a personal call that does not involve the MiVoice Business (i.e. a personal call with the public network), the MiVoice Business may offer an EHDA an ACD call. If the attempt to seize the EHDA fails, then the EHDA will be blocked from receiving another ACD call until the “EHDA reseize timer” expires or is cancelled.

Permanent login COS is ignored for an EHDA

The “Hot Desk External User: Permanent login” COS setting is **ignored** for EHDA, resulting in the following behaviors:

After a system reset, the “Hot Desk External User: Permanent login” COS setting is ignored for an EHDA. Thus, if the EHDA was logged out prior to a system reset, the EHDA will remain logged out.

If an EHDA user logs out the EHDA shall be logged out regardless of the “Hot Desk External User: Permanent login” COS setting.

If an EHDA user enters the Agent Logout FAC the EHDA shall be logged out regardless of the “Hot Desk External User: Permanent login” COS setting.

If the “Auto-logout agent on no answer timer” expires for an EHDA with Permanent Login enabled the EHDA shall be logged out as per existing rules for logging out a ringing agent. The Permanent Login setting is ignored. The last ringing EHDA in a group will be logged out only if “ACD - Auto Logout Last Agent On No Answer” is set to YES.

Programmable Reseize Timer for ACD group calls to an EHDA

When a call queues to an ACD group and fails to ring an EHDA member, a retry timer is needed so that the queued caller can be reoffered to the EHDA member. The reseize timer is **not** started for direct calls to an EHDA.

If an EHDA is offered an ACD call and the system fails to ring the EHDA, then a reseize timer will be started. The ACD caller will be offered to other available agents. No other group call will be offered to the EHDA until the reseize timer expires. When the EHDA’s reseize timer expires, and there are queued ACD callers, then the system will try to ring the EHDA again on behalf of the highest priority, longest waiting ACD caller.

An active reseize timer does not affect direct calls to an EHDA. A direct call will still attempt to ring an EHDA with an active reseize timer.

It is possible that an EHDA with an active reseize timer could answer a direct call while having other group calls waiting.

If an ACD call is made while a reseize timer is active for an EHDA member, then the EHDA member shall be treated as if the member was in make busy.

When a new call fails to ring any member of a group, there are existing rules for treating the “best reason” why the group’s members could not be rung:

- Busy (caller can queue to the group).

- Absent (unless allowed to queue with all members absent, the caller should be rerouted or provided reorder tone).
- DND (group is unavailable and caller should be rerouted or provided reorder tone). This handling will not be provided unless the group DN is put into DND.
- Interconnect restriction (group is unavailable and caller should be rerouted or provided reorder tone).
- Terminator out of service (group is unavailable and caller should be rerouted or provided reorder tone).
- An EHDA can stop a reseize timer by logging out and logging in.
- An EHDA can stop a reseize timer by going make busy and canceling make busy.

It is now possible for ACD callers to interflow out of a path with idle agents that have reseize timers running. Suppose that the last ACD caller to a path interflows out of the path while a reseize timer is running for an EHDA. The EHDA's reseize timer will continue to run even when there are no calls queued to its group. When the EHDA's reseize timer expires, the EHDA will not be offered a call since there are none available.

Suppose a reseize timer is started after unsuccessfully trying to ring an EHDA and while the reseize timer is running, the EHDA performs one of the following actions:

- The EHDA activates Do Not Disturb via Mitel OIG or calls into the switch, logs in, and dials the DND FAC.
- The EHDA becomes absent via the Group Presence API in Mitel OIG or by calling into the switch and dialing a Group Presence Leave FAC.
- Then when the reseize timer expires, no attempt shall be made to ring the EHDA since the EHDA is no longer available. Any callers queued to the group shall remain queued waiting for the next available member.

The reseize timer is a non-resilient entity. Thus, if an EHDA fails over while the timer is running, the timer shall not be restarted after the failover on the secondary controller. Similarly, if an EHDA fails back while the timer is running, the timer shall not be restarted on the primary controller after the failback.

EHDA must manually answer and hang-up calls

An EHDA external device must always answer or clear a call manually. The EHDA cannot answer or clear a call via Mitel OIG. If Mitel OIG is used (answerCall function) to answer a call ringing an EHDA, an INVALID_CALL_ID is returned and the request is rejected.

If Mitel OIG clearCall function is used to clear a call involving an EHDA, the call terminates however the EHDA physical phone is still off hook and dial tone is heard on the phone. All subsequent attempts to ring the EHDA will fail until the EHDA is manually disconnected.

Answer Confirmation and Auto Logout Interaction

If an EHDA has answer confirmation enabled in its COS, then an ACD call shall not be answered until the EHDA confirms by pressing a DTMF key.

Suppose an EHDA is not the last agent in a group, and the EHDA does not confirm the answering of a call before the auto logout timer expires. In this scenario, the agent shall be logged out and the ACD call shall be re-queued.

Suppose an EHDA is the last agent in a group, and the EHDA does not confirm the answering of a call before the auto logout timer expires. In this scenario, the agent will be logged out only if “ACD – Auto Logout Last Agent On No Answer” is set to “YES”.

Attendant Call Information Display

If an EHDA is local to an attendant, a call from an EHDA to the attendant will display the name and number from the EHDA. The device type will be the trunk the user is active on. If the EHDA is remote, then a call from the EHDA will display the device type as the trunk type between the EHDA and the attendant. The trunk information will consist of the trunk type and number.

Call Forwarding

ACD calls do not get forwarded, but direct calls to an EHDA can be forwarded.

Call Park

The EHDA will be able to park calls by using the Call Park FAC. If paging is configured, the EHDA will be connected to the paging equipment or direct page DN. If paging is not configured, the EHDA will be returned dial tone and able to make another call.

Call Park Retrieve

The EHDA can retrieve a call on park by dialing the FAC for Call Park - Retrieve. If the EHDA is parked, any device can retrieve the EHDA as is done today.

Call Recognition Services (CRS) Integration

When an EHDA dials into the MiVoice Business, the calling number shall be used to search the CRS database to see if it is a programmed EHDA. If an EHDA calls the Hot Desking Access Number on a trusted CRS trunk from its designated CRS DN, then the EHDA will not have to enter any PIN to receive service from the MiVoice Business.

Campon

Camping on to a busy EHDA is supported in the same manner as camping on to an EHDU. An EHDA can campon to other devices. When a call is made from the EHDA to any busy destination, the EHDA caller will be provided the same treatment as any other internal station calling a busy destination. Specifically, the EHDA caller will be provided busy tone. The EHDA must press the one-digit Campon FAC to campon to the busy destination. Once camped on, the EHDA caller will hear music on hold or silence (which is dependent upon system programming of music on hold resources). Note that an EHDA call to a busy destination will not auto campon. Moreover, if the EHDA caller does no action while listening to busy, eventually the Busy Tone Timer will expire and then the EHDA caller will hear reorder tone.

Class of Restriction

The EHDA will take on the COR of the HDU and not the trunk.

Class of Service

The EHDA will take on the COS of the HDU and not the trunk.

Direct Page

An EHDA can perform a direct page by using the FAC for direct page.

Emergency Call

Although it is possible to perform an emergency call using an EHDA, it is not recommended to do so because there is no guarantee that the appropriate location information is sent with the emergency call. Mitel assumes no legal, financial or personal responsibility for users or persons performing such actions.

Hold

If an EHDA is put on hold (by the other party), the EHDA can only invoke the cancel call mid-call feature to release from being held and initiate a new call. Any attempt to invoke other mid-call features will be ignored. As a result, the EHDA does not support hold on hold. If an EHDA puts a party on hold using the mid-call feature, the other party can still use the hold on hold feature.

Interflow Timer Expiry While Ringing An EHDA

If the interflow timer expires while ringing an EHDA, the EHDA will continue to ring. The call does not interflow from the path unless the “auto logout agent on no answer” timer expires.

Message Waiting

When an EHDA dials into the system when there is a message waiting for the EHDA, it will hear a broken dial tone to indicate that there is a message waiting for the EHDA. The Message waiting inquiry of an EHDA will be supported.

Music on Hold on Transfer

An EHDA will behave like a local device. If an EHDA is talking with a local trunk (and the EHDA's COS Option “Music on Hold on Transfer” is set to Yes), and transfers the call to a local device, then the caller will maintain MOH during the transfer. If the EHDA transfers the call to a remote device or back out to the PSTN (and the Trunks COS Option “Music on Hold on Transfer” is set to Yes), the caller will maintain MOH during the transfer.

Personal Speedcalls

A personal speedcall can be dialed by an EHDA if the speedcall FAC is used.

PIN Lockout Support

This feature shall be supported for an EHDA.

Devices supported for supervisor (no EHDU supervisor support)

A monitoring Supervisor must be using a local ONS or SS4 device. A supervisor cannot be an EHDU.

Supervisor and agent must be hosted by the same MiVoice Business

The monitoring supervisor and the EHDA must be active and hosted by the same MiVoice Business.

Existing Monitoring Support for EHDA

Almost all existing Mitel OIG operations that support a Hot Desk Agent shall be provided for an EHDA including the following:



Note: The Mitel OIG operations not supported include Answer call and Clear call.

Make busy support	Group presence support	Do not disturb support
Hot desk login and logout	Cancel work timer	Cancel Resize Timer
Call forward programming	Message waiting indication	Make call
Listen call	Cancel call	Consultation call
Alternate call (answer camped on call)	Hold	Transfer
Split	Conference	Retrieve
Redirect	Campon (not on SIP devices or over SIP trunks)	Trade
Intrude	Set account code	Pickup call
Outpulse digits	Query ACD status	

External ACD Agent Login

There are different types of EHDA logins and logouts (e.g., using a PIN or not using a PIN). To receive feature events an application must set one of the feature monitors (monitorFeatures or advMonitorAcd2AgentMonitorFeatures) on the EHDA number. To use monitorFeatures an application must also use monitorObject on an EHDA.

Login User as EHDA with no PIN using External Phone

To login an EHDA number from an external device without specifying a PIN, set the pin attribute to Null.

Login User as EHDA with PIN using External Phone

To login an EHDA number with a PIN, set the pin attribute to the required value.

Auto Answer

The auto-answer feature is not supported for EHDA.

Cancel ACD Work Timer

When an EHDA answers and clears an ACD call through an ACD path and ACD group (both ACD2 and ACD Express) an ACD Work timer is started. An application can cancel an ACD work timer for an EHDA.

Call Forward Programming

The following call forwarding features are supported for EHDA:

FEATURE ACTION
Set CFNA External
Cancel CFNA External
Set CFNA Internal
Cancel CFNA Internal
Set CFB External
Cancel CFB External
Set CFB Internal
Cancel CFB Internal
Set CFA
Cancel CFA
Set CFNA and CFB
Cancel CFNA and CFB

Cancel Reseize Timer

An application can cancel the reseize timer of an EHDA. Only the Cancel Feature can be used. The reseize timer only applies to EHDA number logged in externally. Also reseize timer only applies after an ACD path call to EHDA is disconnected.

Feature Events for EHDA

ActivateFeatureEvent

This event is generated when a binary feature for EHDA has changed.

The supported binary features are:

- AutoAnswerFeature
- DoNotDisturbFeature
- MakeBusyFeature
- MessageWaitingIndicator
- ReseizeTimerFeature

ForwardFeatureEvent

This event is generated when a forwarding feature on an EHDA is changed.

The supported forwarding features are:

- CFAlways
- CFBusyExternal
- CFBusyInternal
- CFNoAnswerExternal
- CFNoAnswerInternal
- CFNoAnswerBusy
- CFAAllModes.

Group Presence Feature Events

These events are generated when a presence feature of an EHDA is changed.

The supported presence features are:

- JoinGroupFeature
- LeaveGroupFeature
- JoinAllACDGroupsFeature
- LeaveAllACDGroupsFeature

UserIDFeatureEvent

This event is generated when an EHDA logs in or logs out.

Application Scenarios for EHDA

Silent Monitor

In ACD, a supervisor can initiate a silent monitor on an agent. Only a phone on same MiVoice Business as the ACD agent can be a supervisor of a silent monitor.

The following PBX programming is required:

- Agent's COS "ACD Silent Monitor Accept" – enabled

- Supervisor's COS "ACD Silent Monitor Allowed" – enabled.

The following call scenarios assume ACD call reaches MiVoice Business via trunk connection. In addition, trunk, agent and supervisor all reside in the same MiVoice Business.

Silent Monitor Initiated After Call Established

An ACD supervisor initiates silent monitor on an agent who already has an ACD call established. The call status events for the scenario are shown below.

CALLER (TRUNK)	ACD AGENT	ACD SUPERVISOR
CallEstablishedEvent / EstablishedState / Answered GCID = G1 Connected Device = agent	CallEstablishedEvent / EstablishedState / AnswerInvoked GCID = G1 Connected Device = caller	
		(invoke silent monitor)
		CallOriginatedEvent / OriginatedState / NewCallInvoked GCID = G2
		CallDeliveredEvent / DeliveredState / ACDSilentMonitorInvoked GCID = G2
CallConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G1	CallConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G1	CallConferencedEvent / EstablishedState / ACDSilentMonitorEstablished GCID = G3 / Primary = G2
		(hang up)
DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	CallClearedEvent / IdleState / ClearInvoked GCID = G3

Silent Monitor Initiated Before Call Established

An ACD supervisor initiates silent monitor on an idle agent. The agent then receives a new ACD call. The call status events for the scenario are shown below.

CALLER (TRUNK)	AGENT	SUPERVISOR
		(invoke silent monitor)
		CallOriginatedEvent / OriginatedState / NewCallInvoked GCID = G1

CALLER (TRUNK)	AGENT	SUPERVISOR
		CallDeliveredEvent / DeliveredState / ACDSilentMonitorInvoked GCID = G1
(new ACD call)		
CallOriginatedEvent / OriginatedState / NewCallInvoked GCID = G2		
CallDeliveredEvent / DeliveredState / Delivered GCID = G2 Called Dev = ACD path	CallReceivedEvent / ReceivedState / ACDCallReceived GCID = G2 Calling Dev = caller	
(answer)		
CallEstablishedEvent / EstablishedState / Answered GCID = G2 Connected Dev = agent	CallEstablishedEvent / EstablishedState / AnswerInvoked GCID = G2 Connected Dev = caller	
CallConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G2	CalConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G2	CallConferencedEvent / EstablishedState / ACDSilentMonitorEstablished GCID = G3 / Primary = G1
(hang up)		
DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	CallClearedEvent / IdleState / ClearInvoked GCID = G3

Request Help

In ACD, an agent can request help from a supervisor.

The following MiVoice Business programming is required:

- Supervisor's COS "ACD Silent Monitor Allowed" – enabled.

The following call scenarios assume ACD call reaches MiVoice Business via trunk connection. In addition, trunk, agent and supervisor all reside in the same MiVoice Business.

Call Events for Request Help

An agent who has an ACD call established requests help from a supervisor. The call status events for the scenario are shown below.

CALLER (TRUNK)	AGENT	SUPERVISOR
CallOriginatedEvent / OriginatedState / NewCallInvoked GCID = G1		
CallDeliveredEvent / DeliveredState / Delivered GCID = G1 Called Dev = ACD path	CallReceivedEvent / ReceivedState / ACDCallReceived GCID = G1 Calling Dev = caller	
(answer)		
CallEstablishedEvent / EstablishedState / Answered GCID = G1 Connected Dev = agent	CallEstablishedEvent / EstablishedState / AnswerInvoked GCID = G1 Connected Dev = caller	
(request help)		
	CallDeliveredEvent / DeliveredState / ACDSilentMonitorInvoked GCID = G2	CallReceivedEvent / ReceivedState / ACDHelpCall GCID = G2 Calling Dev = agent
(answer)		
CallConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G1	CallConferencedEvent / EstablishedState / ACDSilentMonitorEstablished GCID = G3 / Primary = G2 / Secondary = G1	CallConferencedEvent / EstablishedState / ACDSilentMonitor GCID = G3 / Primary = G2
(hang up)		
DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	DeviceDroppedEvent / EstablishedState / ACDSilentMonitorMemberDropped GCID = G3	CallClearedEvent / IdleState / ClearInvoked GCID = G3

Agent Auto Logout on No Answer

If an agent does not answer an ACD call after the auto logout timer expires, then the agent shall be logged out and the ACD call shall be re-queued.

The following MiVoice Business programming is required:

- COS “ACD Logout Agent No Answer Timer”
- System Option “ACD – Auto Logout Last Agent on No Answer”.

Call Events for Agent Auto Logout

Let an ACD call ring an agent until the Logout-Agent-No-Answer timer expires. The call events for the scenario are shown below.

CALLER	EHDA (AGENT)	PATH
CallDeliveredEvent / DeliveredState / Delivered		ACDPathEvent / ReceivedState / ACDRequest
	CallReceivedEvent / ReceivedState / ACDCallReceived	ACDPathEvent / IdleState / ACDCallDelivered
	(Auto logout on no answer)	
	CallClearedEvent / IdleState / ACDAgentTimeout	ACDPathEvent / ReceivedState / ACDRequeueRequest
	OutOfServiceEvent / UnavailableState / DeviceOutOfService	

Feature Event for Agent Auto Logout

When an agent is auto logged out, a `UserIdFeatureEvent` is also generated if a feature monitor is enabled (for example, ACD-class feature monitor). Please see the previous section for the detail of the `UserIdFeatureEvent`.

Unsupported Features for EHDA

The features not supported for EHDA are listed below:

- AnswerCall for EHDA external user.
- ClearCall for EHDA external user.
- AssignCallerId.
- TapCall
- AlterMessage
- Auto-answer feature for EHDA external user.
- Query reseize timer feature status.
- Query ACD login feature.
- Query trunk make-busy feature.

EHDA supervisors for silent monitor – There is no interface provided to allow an EHDA supervisor to select the “next call” to monitor.

Record-a-call control - An EHDA does not have any control of its active record-a-call session. There is no API to allow an EHDA to perform the following actions:

- Pause a recording
- Resume a recording
- Stop recording and save the recording
- Stop recording and erase the recording

When trying to login/logout on an external device, Superkey and soft key interfaces for login/logout are not available to an EHDA (Feature keys are not available to an EHDA that is

logged in as a remote destination). An EHDA external device is essentially a single line device with no soft keys and no superkey. Therefore, an EHDA cannot access any features via the Superkey or soft key interfaces, including these popular features:

- Hotdesking (login/logout)
- Do not disturb
- Auto answer
- Make busy

EHDA and ACD Rules:

- A feature access code does not exist for canceling an ACD agent Work Timer.
- An EHDA cannot request help via soft key prompt, feature access code, or a programmed feature key.
- An EHDA Supervisor is not supported.
- An EHDA cannot handoff since it cannot be programmed in a Personal Ring Group.
- ACD Express Agent Groups are not resilient. Since SDS will not share ACD Express Group data, a resilient EHDA programmed in an ACD Express Group on its Primary will typically be shared as a Hot Desk Agent programmed in no groups on its Secondary. As such, the agent will be considered an ACD2 agent there until it is programmed into an ACD Express Group on the Secondary.
- An EHDA cannot be a member of a hunt group or ring group since ACD Agents cannot be programmed into a hunt group or ring group.
- An EHDA cannot be a prime member of a multicall group or key system group since ACD Agents cannot be the prime member of a multicall group or key system group.
- An EHDA cannot be a member of a PRG since ACD Agents cannot be members of a PRG.
- Application cannot set a reseize timer.
- EHDA Reseize timer is not resilient.
- An EHDA does not support reminders. When an EHDA logs in, any reminder that is set is cancelled.
- Calls to an EHDA will not support ringing discrimination.
- Tag Call is not supported on an EHDA since there are no programmable keys available on an EHDA external device.
- ACD queue status is not supported on an EHDA since the HMI is not available.
- Any attempt to perform a direct page on an EHDA will be rejected and the user will receive NOT ALLOWED and receive reorder tone.
- Currently an EHDU cannot be a member of an Emergency Hunt Group. Thus, an EHDA also cannot be a member of an Emergency Hunt Group and will not receive any Emergency Call Notifications.
- An EHDA cannot park or retrieve the parked call because it does not have access to keys.
- EHDA does not support hold on hold
- Message Waiting Ringing Notification is not supported for an externally logged in EHDA.
- An EHDA has no ability to request a messaging-callback

- Programmable speedcall line keys are not supported on an EHDA, as there are no line keys on an EHDA.
- Phone Lock is not available to an EHDA. If an EHDA attempts to use the phone lock FAC, they will receive reorder tone.

Call Flow Exceptions for Call Park and Paging

Loudspeaker Page

User lifts a MiVoice Business phone handset, dials FAC (Feature Access Code) for page, and speaks a message over the loudspeaker. A monitor on the user's phone 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 a message. A monitor on the user's phone 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 a MiVoice Business phone for an incoming call, then dials the CallPark FAC plus target DN. A monitor on the user's phone 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 a MiVoice Business phone for an incoming call, then the DirectPage FAC plus target DN. A monitor on the user's phone 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 a MiVoice Business phone for an incoming call, then dials LoudSpeaker FAC plus target DN, and speaks a message over the loudspeaker. A monitor on the user's phone 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 pick up the parked call. A monitor on the user's phone will report the following:

EVENT	STATE	CAUSE
CallOriginatedEvent	OriginatedState	NewCallInvoked
CallEstablishedEvent	EstablishedState	Answered



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

SIP Device Support

The Mitel 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. Mitel 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.

The table below provides a list of operations that work, but work differently in many cases.

OPERATION	SUPPORTED FOR SIP	SUPPORTED FOR EHDU
AlternateCall (Swap)	Yes	Yes
AnswerCall	No	No
AssignCallerId	No	No
CamponCall	No	No
CancelConsCall	Yes	Yes
ClearCall	No	No
ConferenceCall	Yes	Yes
ConsultationCall	Yes	Yes
HoldCall	Yes	Yes
IntrudeCall	Yes	Yes
ListenCall	No	No
MakeCall	Yes	Yes
NewCall	Yes	Yes
OutpulseDigits	Yes	Yes
PickupCall	Yes	Yes
RedirectCall	Yes	Yes
RetrieveCall	Yes	Yes
SendCallbackMessage	No	No

OPERATION	SUPPORTED FOR SIP	SUPPORTED FOR EHDU
SetAccountCode	Yes	Yes
SetCallMeBack	No	No
SplitConferenceCall	Yes	Yes
TapCall	No	No
TradeCall	Yes	No
TransferCall	Yes	Yes

While the use of stimulus messages can be used to update the displays, states, and LEDs of a MiVoice Business 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 MiVoice Business call process.

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 MiVoice Business has no control over what is displayed by the SIP phone. Thus, several ANI/DNIS COS fields cannot be applied to SIP phones.

The following COS options affect ANI/DNIS behavior:

- **COS: ANI/ISDN Calling Number Only.** This COS field cannot 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 MiVoice Business 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 MiVoice Business has no control over what is displayed by the SIP phone. Thus, this COS field cannot be applied to SIP phones. When a MiVoice Business 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.

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
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
Mitel 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

