

# SCOPIA iVIEW Management Suite Web API

Version 7.6



## NOTICE

© 2005-2011 RADVISION Ltd. All intellectual property rights in this publication are owned by RADVISION Ltd and are protected by Italy and United States copyright laws, other applicable copyright laws and international treaty provisions. RADVISION Ltd retains all rights not expressly granted.

This publication is RADVISION confidential. No part of this publication may be reproduced in any form whatsoever or used to make any derivative work without prior written approval by RADVISION Ltd.

No representation of warranties for fitness for any purpose other than what is specifically mentioned in this guide is made either by RADVISION Ltd or its agents.

RADVISION Ltd reserves the right to revise this publication and make changes without obligation to notify any person of such revisions or changes. RADVISION Ltd may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

If there is any software on removable media described in this publication, it is furnished under a license agreement included with the product as a separate document. If you are unable to locate a copy, please contact RADVISION Ltd and a copy will be provided to you.

Unless otherwise indicated, RADVISION registered trademarks are registered in the United States and other territories. All registered trademarks recognized.

For further information contact RADVISION or your local distributor or reseller.

SCOPIA iVIEW Management Suite Web API Version 7.6, March 2011

<http://www.radvision.com>

# CONTENTS

---

## *About This Manual*

Related Documentation	vii
Conventions Used in this Manual	vii
Feedback	viii

## **1** *Introduction*

Features	1
API Deployment	2
Components	3

## **2** *Service Version Management*

## **3** *Service Invoking and Definitions*

Service Invoking	7
Introduction	7
Constructing a WS Client	7
Service Definitions	8
Introduction	8
ResourceService	9
getMeetingTypes API	9
getTerminals API	10
getRooms API	11
getDeviceControllerIpPort API	12
searchTerminal API	13
getTerminalByIDs API	14

deleteTerminals API	15
setTerminals API	16
ResourceV2Service	17
ControlService	17
extendLiveConference API	17
terminateLiveConference API	19
inviteNewTerminal API	20
dropAttendedTerminal API	21
ScheduleService	22
createConference API	23
createConferenceSP API	25
modifyConference API	29
modifyConferenceSP API	32
cancelConference API	36
cancelConferences API	37
viewConference API	38
checkAlternateTerminals API	39
checkAlternativeTimes API	40
createRecurrence API	41
modifyRecurrence API	43
cancelRecurrence API	45
viewRecurrence API	46
createVirtualRoom API	47
modifyVirtualRoom API	49
deleteVirtualRoom API	51
viewVirtualRooms API	52
viewVirtualRoomsByUserID API	53
viewVirtualRoomByName API	54
viewVirtualRoomByID API	55
viewVirtualRoomByDialableNumber API	56
viewConferenceByDialableNumber API	57
getDialedNumberConferenceInfo API	58
getVersion API	60
searchConferences API	61
deleteHistoryConferences API	62
removeConference API	63
UserService	64

authUser API	64
setUsers API	65
deleteUsers API	66
searchUsers API	67
getUserByLoginID API	68
getUsers API	69
DirectoryUser	
Service	69
prepareSession API	70
prepareUsers API	71
submit API	72
cancel API	73

#### 4 *Associated Class Definitions*

ResourceService/ResourceV2Service	76
MeetingType Class	76
Layout Class	78
View Class	79
DeviceControllerIpPort Class	80
RoomResource Class	81
TerminalResource Class	82
ISDNLocation Class	85
TelePresenceInfo Class	86
TPSegment Class	87
ScheduleService	87
AvailableInfo Class	87
AlternateTerminal Class	88
AlternateTerminalsTimedSlice Class	89
EventInfo Class	90
LayoutInfo Class	91
ViewInfo Class	92
RoomInfo Class	93
TerminalInfo Class	94
ScheduleInfo Class	96
GatewayInfo Class	97
GatewayPort Class	98

TerminalLayoutInfo Class	99
ConferenceInfo Class	100
VirtualRoomInfo Class	109
SDGInfo Class	110
SDGAccessInfo Class	111
ConferenceAdvancedInfo Class	112
RecurrenceInfo Class	114
RecurrenceInstanceInfo Class	115
ScheduleResult Class	116
ConferenceInfoCondition Class	121
ControlService	122
Terminal Class	122
ControlResult Class	124
UserService	127
UserInfo Class	127
UserSearchCondition Class	128
DirectoryUser	
Service	128
DirectoryOperationSession Class	128
DirectoryUserInfo Class	130

## 5 *Location, Dependencies and Properties*

iCM Service Location	131
Dependencies	131
Configurable Properties	132
Associated WSDL Files	132

## APPENDIX A *Error Codes*

# ABOUT THIS MANUAL

---

The [iVIEW Suite Web Service API Reference Guide](#) provides information for administrators about using the iVIEW Suite to extend the functionality of SCOPIA iVIEW Management Suite.

## RELATED DOCUMENTATION

The SCOPIA iVIEW Management Suite documentation set is available on the RADVISION Utilities and Documentation CD-ROM supplied with the product and includes manuals and online helps. The manuals are in PDF format.

---

**Note** You require Adobe Reader version 5.0 or later to open the PDF files. You can download Acrobat Reader free of charge from [www.adobe.com](http://www.adobe.com).

---

## CONVENTIONS USED IN THIS MANUAL

Below is a list of conventions used throughout this manual:

- API names start with a lowercase character.
- Array class names end with a “List” string.
- Associated attribute class names for the ResourceService web service end with a “Resource” string.
- Associated attribute class names for the ScheduleService web service end with an “Info” string.
- Associated attribute class names for the ControlService web service have no suffix.

- Operation result class names for these web services end with a “Result” string.
- All time variables are calculated in milliseconds since “The Unix Epoch” (January 1st, 1970).

## FEEDBACK

The team at RADVISION constantly endeavors to provide accurate and informative documentation. If you have comments or suggestions regarding improvements to future publications, we would value your feedback.

Please send your comments to [doc\\_comments@radvision.com](mailto:doc_comments@radvision.com).

We thank you for your contribution.

# 1

## INTRODUCTION

---

- [Features](#) on page 1
- [API Deployment](#) on page 2
- [Components](#) on page 3

### FEATURES

This manual documents the following:

- **Service Versions**—This section mainly documents the versions of the iVIEW Communications Manager web service
- **Service Definitions**—This section documents three kinds of web services exposed in the iVIEW Communications Manager system:
  - iVIEW Communications Manager resource web service
  - iVIEW Communications Manager conference control web service
  - iVIEW Communications Manager conference schedule web service
- **Associated Class Definitions**—This section describes all the classes involved with these three web services mentioned above, including their parameters and constant fields
- **Associated WSDL Files**—This section sources the WSDL (Web Service Definition Language) files associated with these three web services

## API DEPLOYMENT

By default, Web Service can only be visited by the local client.

To change the Web Service permissions:



### Procedure

- 1 Browse to JBOSS\_HOME/bin/vcs-core.properties.
- 2 Open vcs-core.properties in a text editor such as Notepad.
- 3 Set the following property to false:  
com.radvision.icm.service.init.checkClientAddress=false

See [Figure 1-1](#).

**Figure 1-1** Changing the Web Service Permissions

```

context.dialstring.beforeBandwidth.splitStr=@
configtool.hidetab.tabsystem.tabUI=false
configtool.hidetab.tabsystem.tabSNMP=true
configtool.hidetab.tabcdr.tabscheduling=true
configtool.hidetab.tabcdr.tabActualInformation=true
vnex.vcms.core.enableLowLevelInfo=false
vnex.vcms.core.conference.server.enableRestrictCascadedConference=
context.inmeetingctrl.showEPSInfoInLog=false
vnex.vcms.core.addParentZonePrefix=false
vnex.vcms.oss.mechanism.force.real.mac=true
#To control the function of auto config.
vnex.vcms.core.startResourceActionListener=false
vnex.vcms.core.allowVirtualRoom4Organizer=false
context.inmeetingctrl.sendEmail=false
#For icmservice
com.radvision.icm.service.init.checkClientAddress=false
vnex.vcms.core.ra.schedule.reservedCascadePorts=0
vnex.vcms.core.authorizeEndpointInitiatedP2PCalls=true
vnex.vcms.core.startup.handler.601.ConferenceMonitorStartup=com.vi
vnex.vcms.core.conference.maxExtendTime=1440000
context.inmeetingctrl.checkConferenceModification=5000
context.inmeetingctrl.inviteConfEnabled=false
com.visionnex.vcms.core.ra.impl.rv.enableoutcall=true

# for switch the two verison of inmeetingctrl
context.inmeetingctrl.ui.switchnew=true

vnex.vcms.core.allowCascadingForVCSPro=true
Production.SDG.cascadeChannel.name=SCOPIA Desktop;Desktop Server
vnex.vcms.core.allowLoadConfInfoFromDefaultTemplate=true

# default mininum days for deleting history days
vnex.vcms.core.conferencecleaner.defaultDays=365

```

---

**Note** If the property does not exist, add this line to the file.

---

JBOSS\_HOME is the default home directory of the SCOPIA iVIEW Management Suite installation. The default path is:

C:\Program Files\RADVISION\iVIEW Suite\iCM\jboss-3.2.5\bin

---

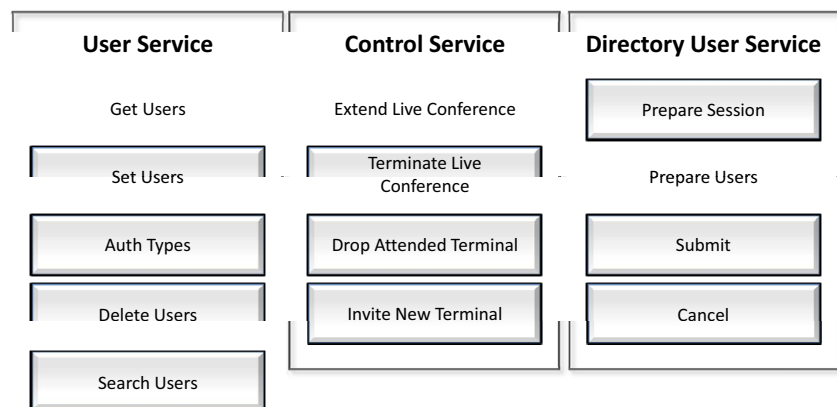
## COMPONENTS

The [iVIEW Suite Web Service API Reference Guide](#) provides information for programmers to build onto iVIEW Suite using API services provided by iVIEW Communications Manager. iVIEW Communications Manager provides six kinds of web services:

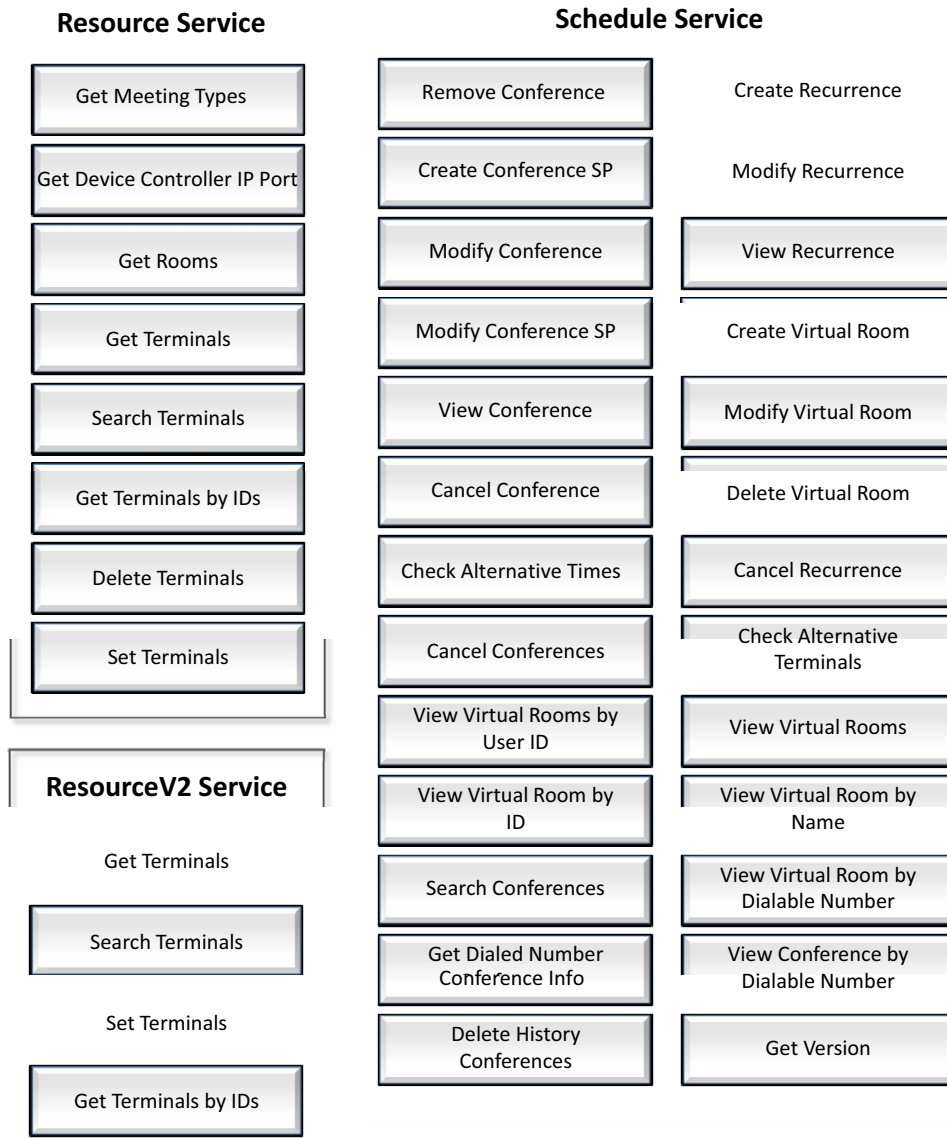
- UserService
- ControlService
- DirectoryService
- ScheduleService
- ResourceService
- ResourceV2Service

[Figure 1-2](#) and [Figure 1-3](#) show the four web services and their exposed API names.

**Figure 1-2** Web Services and their Exposed APIs



**Figure 1-3** Web Services and their Exposed APIs



# 2

## SERVICE VERSION MANAGEMENT

---

iVIEW Communications Manager web services middleware performs version management, this is beneficial for stability and compatibility. Clients need to give their version number when using the web services. The current version number of the iVIEW Communications Manager web services is 1.0 except for ResourceService that supports both 1.0 and 2.0. The API pfResourceV2Service supports Telepresence terminals.

The URL format to support version 1.0 management is:

`http://.../1.0/XXXService`

The URL format to support version 2.0 management is:

`http://.../2.0/ResourceService`

An example of using the URL format to access iVIEW Communications Manager ResourceService:

`http://icm.radvision.com:8080/icmservice/1.0/ResourceService`

An example of using the URL format to access iVIEW Communications Manager ResourceV2Service:

`http://icm.radvision.com:8080/icmservice/2.0/ResourceService-`



# 3

## SERVICE INVOKING AND DEFINITIONS

---

- [Service Invoking](#) on page 7
- [Service Definitions](#) on page 8

### SERVICE INVOKING

#### INTRODUCTION

This iVIEW Communications Manager web services API exposes the external client system iVIEW Communications Manager services.

#### CONSTRUCTING A WS CLIENT

##### Before You Begin

Ensure that the web services component is supported or enabled in the currently equipped iVIEW Communications Manager as follows:



##### Procedure

- 1 Verify that the first web page of your iVIEW Communications Manager system is located at the following URL:  
`http://x.x.x.x:yyyy/icm`  
where the server IP address is represented by x.x.x.x and the port is represented by yyyy.  
Example: `http://127.0.0.1:8080/icm`

## Service Definitions

- 2 Enter your URL link into the browser of your client machine according to the following format:

`http://<iVIEW Communications Manager Server IP Address>:<Port>/icmservice/1.0/<Service Name>?wsdl`

Example: `http://127.0.0.1:8080/icmservice/1.0/ScheduleService?wsdl`

Regarding ResourceV2Service, the URL link is `http://x.x.x.x:yyyy//icmservice/2.0/ResourceService?wsdl`

Example: `http://127.0.0.1:8080/icmservice/2.0/ResourceService?wsdl`.

An XML (WSDL) message displays in the browser indicating that the iVIEW Communications Manager web service is functioning.

---

You must generate WS stubs before you can use the iVIEW Communications Manager web service APIs.

When we are sure the iVIEW Communications Manager web services API is functioning correctly, then we can use published WSDL files to generate WS stubs by means of a third party tool such as GSOAP for C/C++, Ant for Java or Microsoft Visual Studio for DotNET. The client system can invoke this stub to use the business method of the services published by the iVIEW Communications Manager. We can also write extra codes, namely stub, to directly invoke WS API.

## SERVICE DEFINITIONS

- [Introduction](#) on page 8
- [ResourceService](#) on page 9
- [ResourceV2Service](#) section on page 17
- [ControlService](#) section on page 17
- [ScheduleService](#) on page 22
- [UserService](#) on page 64
- [DirectoryUser Service](#) on page 69

## INTRODUCTION

Use each of the following methods for each iVIEW Communications Manager service to invoke interactive communication between them.

## RESOURCE SERVICE

The ResourceService has 4 APIs to retrieve the 4 basic kinds of resources in iVIEW Communications Manager. The 4 APIs are:

- [getMeetingTypes API](#)
- [getTerminals API](#)
- [getRooms API](#)
- [getDeviceControllerIpPort API](#)

---

### getMeetingTypes API

#### DESCRIPTION

Use the getMeetingTypes API to retrieve all uploaded and active meeting types into the MCU in iVIEW Communications Manager. MeetingType object stores the associated parameters of the MCU service. See [Table 3-1](#) for the parameter table.

**Table 3-1** MeetingType object Parameter Table

Parameter	Description
Meeting Type ID	The SCOPIA iVIEW Management Suite meeting type ID.
Name	The name of the meeting type.
Description	A description of the meeting type.
Service Prefix	
MCU Service Type	
Switching Mode	
Bandwidth	
Resolution	

The returned object contains a collection of MeetingType objects.

**ResourceService**  
getMeetingTypes API

### **DEFINITION**

```
public List<MeetingType> getMeetingTypes();
```

---

## getTerminals API

### DESCRIPTION

Use the getTerminals API to retrieve all the iVIEW Communications Manager terminals. The TerminalResource object stores the associated parameters of the conference terminal or party. The returned object contains a collection of TerminalResource information. When the classificationName parameter is equal to ResourceService.ALL\_CLASSIFICATIONID, this API returns all the terminals associated with this classification.

### DEFINITION

```
public List<TerminalResource> getTerminals(String  
classificationName) throws  
NoClassificationFoundException, InvalidArgumentException;
```

### PARAMETERS

**Table 3-2**      *API Parameter Table*

Parameter	Description
classificationName	Indicates the network in which this scheduled conference is held.

---

---

## getRooms API

### DESCRIPTION

Use the getRooms API to retrieve information on all the conference rooms of iVIEW Communications Manager. The RoomResource object stores the parameters of the conference room for the terminal. The returned object contains a collection of RoomResource.

### DEFINITION

```
public List<RoomResource> getRooms();
```

---

## getDeviceControllerIpPort API

### DESCRIPTION

Use the getDeviceControllerIpPort API to retrieve the IP address and port of the device controller. The returned object contains the IP address and port of the device controller.

### DEFINITION

```
public DeviceControllerIpPort  
getDeviceControllerIpPort() throws  
NoClassificationFoundException, InvalidArgumentException;
```

---

## searchTerminal API

### DESCRIPTION

Use the searchTerminal API to return information on the terminals that meet the condition defined by the parameters.

### DEFINITION

```
public List<TerminalResource> searchTerminal(String  
classificationName, String name, String dialingInfo)  
throws  
NoClassificationFoundException, InvalidArgumentException
```

### PARAMETERS

**Table 3-3** searchTerminal API Parameter Table

Parameter	Description
ClassificationName	The classification name. This parameter defaults to <i>all</i> when no classification is defined.
Name	The terminal name or part of the terminal name.
DialingInfo	e164, telephone number or SIP URI.

### RETURN VALUE

The terminals that meet the condition defined in the parameters.

### EXAMPLE

```
port.searchTerminal("all", "ter", "555");
```

This query returns the terminals with the names containing *ter* and the dialing strings containing *555*.

---

## getTerminalByIDs API

### DESCRIPTION

Use the getTerminalByIDs API to return a list of terminal IDs with specified identities.

### DEFINITION

```
public List<TerminalResource> getTerminalsByIDs(List<String>
terminalIDs)
```

### PARAMETERS

**Table 3-4** *getTerminalByIDs API Parameter Table*

Parameter	Description
TerminalIDs	A list of terminal IDs.

### RETURN VALUE

The terminals with specified IDs.

---

## deleteTerminals API

### DESCRIPTION

Use the deleteTerminals API to remove the specified terminals from the system.

### DEFINITION

```
public ResourceResult deleteTerminals(List <String> terminalIDs)
```

### PARAMETERS

**Table 3-5** *deleteTerminals API Parameter Table*

---

Parameter	Description
TerminalIDs	The list of terminal IDs.

---

### RETURN VALUE

ScheduleResult.success is true if the terminals are deleted, false otherwise (for example, if one of the terminals does not exist).

---

## setTerminals API

### DESCRIPTION

Use the setTerminals API to create new terminals or update existing terminals. If TerminalResource.terminalID is null, the terminal is created, otherwise the terminal is updated.

### DEFINITION

```
public ResourceResult setTerminals(List<TerminalResource>
terminals, boolean overwritten)
```

### PARAMETERS

**Table 3-6** *setTerminals API Parameter Table*

Parameter	Description
Terminals	The terminals to be created or updated.
Overwritten	If true, only the terminals listed in the parameter are retained, the other terminals are removed.

### RETURN VALUE

ResourceResult.success is true for success, false otherwise. When ResourceResult.success is false, ResourceResult.errorCode contains the error code:

- 101—The terminal list is empty
- 102—The list contains null terminal
- 103—Failed to create or update the terminals
- 104—Terminal is duplicated because of duplicate IP, ISDN number, terminal name, SIP URI etc.

**ResourceV2Service**  
extendLiveConference API

## RESOURCEV2SERVICE

This service is derived from ResourceService and compatible with ResourceService. All API methods of the service are the same as those of ResourceService. The following APIs support Telepresence terminals: getTerminals, searchTerminal, getTerminalsByIds, setTerminals. Regarding the APIs' details, refer to [ResourceService](#) above.

## CONTROLSERVICE

The iCMConferenceControl service has 3 APIs to control the in-meeting conference managed by iVIEW Communications Manager. They extend the live conference, terminate the live conference and modify the live conference.

---

### extendLiveConference API

#### DESCRIPTION

Use the extendLiveConference API to extend the in-meeting conference with the specified time.

#### DEFINITION

```
public ControlResult extendLiveConference(String  
conferenceID, long extendTime);
```

#### PARAMETERS

Both parameters are required.

**Table 3-7** *extendLiveConference API Parameter Table*

Parameter	Description
conferenceID	The unique conference identification.
extendTime	The end time in milliseconds after extending during the ongoing conference.

#### RETURN VALUE

When the extended control is completed, a control result, namely ControlResult, which contains the information related to success or failure, is returned. If the API fails, the detailed error information is enclosed in it.

Error Code in ControlResult.errorCode:

- ERROR\_UNSPECIFIED(-1)—The Required arguments (conferenceId and/or extendTime) are not defined
- ERROR\_CONFERENCE\_NOT\_IN\_MEETING(102)—The conference is not in session

For the other error code, see the [Error Codes](#) appendix.

---

## terminateLiveConference API

### DESCRIPTION

Use the terminateLiveConference API to terminate the in-meeting conference. Reasons for termination could be expired time or user intervention.

### DEFINITION

```
public ControlResult terminateLiveConference(String  
conferenceID);
```

### PARAMETERS

**Table 3-8** terminateLiveConference API Parameter Table

Parameter	Description
conferenceID	The unique conference identification.

### RETURN VALUE

Error code in ControlResult.errorCode:

- ERROR\_UNSPECIFIED(-1)—ConferenceID is not defined
- ERROR\_CONFERENCE\_NOT\_IN\_MEETING(102)—The conference is not in session

For the other error codes, see the [Error Codes](#) appendix.

---

## inviteNewTerminal API

### DESCRIPTION

Use the inviteNewTerminal API to invite a new terminal to the in-meeting conference.

### DEFINITION

```
public ControlResult inviteNewTerminal(String conferenceID,  
Terminal t);
```

### PARAMETERS

**Table 3-9** *inviteNewTerminal API Parameter Table*

Parameter	Description
Conference ID	The unique conference identification.
Terminal Object	Is invited newly; each internal terminal has a unique terminal ID. Terminal IDs are automatically assigned by the iVIEW Communications Manager system to external terminals after inviting them to the conference.

### RETURN VALUE

Error code in ControlResult.errorCode:

- ERROR\_UNSPECIFIED(-1)—ConferenceId is not defined
- ERROR\_CONFERENCE\_NOT\_IN\_MEETING(102)—The Conference is not in session

For the other error code, see the [Error Codes](#) appendix.

---

## dropAttendedTerminal API

### DESCRIPTION

Use the dropAttendedTerminal API to drop an attended terminal from the in-meeting conference.

### DEFINITION

```
public ControlResult dropAttendedTerminal(String  
conferenceID, String terminalID);
```

### PARAMETERS

**Table 3-10** *dropAttendedTerminal API Parameter Table*

Parameter	Description
conferenceID	The unique conference identification.
terminalID	The terminal ID.

### RETURN VALUE

Error code in ControlResult.errorCode:

- ERROR\_UNSPECIFIED(-1)—The Required arguments (conferenceId and/or terminalID) are not defined
- ERROR\_CONFERENCE\_NOT\_IN\_MEETING(102)—The conference is not in session

For the other error code, see the [Error Codes](#) appendix.

## **SCHEDULESERVICE**

The schedule service exposes 11 APIs to schedule the upcoming conferences or recurrences managed by iVIEW Communications Manager. These schedule operations are:

- Creating the conference
- Modifying the conference
- Canceling the conference
- Viewing the conference
- Checking the available conference times
- Creating
- Modifying
- Canceling
- Viewing the recurrent conference

The following is a brief summary of the API classifications for the schedule service module:

- Conference:
  - Create (or schedule)
  - Modify (or reschedule)
  - Cancel
  - View
- Recurrence:
  - Create (or schedule)
  - Modify (or reschedule)
  - Cancel
  - View
- Checking:
  - Check alternate terminals
  - Check alternate times

---

## createConference API

### DESCRIPTION

Use the createConference API to create a new conference with the specified conference information object, ConferenceInfo.Definition. This API has support to create a non-video conference.

### DEFINITION

```
public ScheduleResult createConference(ConferenceInfo info);
```

### PARAMETERS

ConferenceInfo must set the following mandatory parameters:

- Start Time
- End Time
- Meeting Type ID
- Terminals
- TestSchedule (mandatory for non-video conferences only)

**Table 3-11** createConference API Parameter Table

Parameter	Description
Start Time	The conference start time in milliseconds.
End Time	The conference end time in milliseconds.
Meeting Type ID	The conference meeting type ID, referred to as Service Template ID.
Terminals	The terminals invited by the conference. The conference makes no sense without them.
Conference ID	iVIEW Communications Manager uses the ID given by the client. If the Conference ID contains an empty string (“”) or NULL, iVIEW Communications Manager assigns an ID automatically. A conflict may happen as the client does not check if the conference ID is already used by the iVIEW Communications Manager.

---

**Table 3-11** *createConference API Parameter Table (continued)*

Parameter	Description
Classification Name	If the classification parameter contains an empty string or NULL, iVIEW Communications Manager treats the API as if the client does not want a classification and does not return a non-found error code.
TestSchedule	The TestSchedule parameter is mandatory for non-video conferences. It is not used for video conferences. If the parameter is true, iVIEW Communications Manager checks the creation of a conference without committing the scheduling result to iVIEW Communications Manager and creating a conference. The default is false.

## RETURN VALUE

ScheduleResult.errorStatus contains the information related to success or failure. If the API fails, the detailed error information is enclosed in it.

When scheduling a conference, iVIEW Communications Manager first checks if the specified classification name is invalid or does not exist in iVIEW Communications Manager. If it is invalid or does not exist, the ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME = 111 error is returned, or else it is ignored.

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NULL\_MEETING\_TYPE\_ID\_ARGUMENT(107)—The Meeting type ID is not defined.
- ERROR\_NOT\_FOUND\_MEETING\_TYPE(22)—The specified meeting type is not found.
- ERROR\_INACTIVE\_MEETING\_TYPE(23)—The specified meeting type is inactive.
- ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME(111)—The classification is not found.
- ERROR\_MEETING\_CONTROL(99)—The conference cannot be created because there are no available resources in the devices (such as MCU or SCOPIA Gateway), or the conference has already been created.

For the other error code, see the [Error Codes](#) appendix.

---

## createConferenceSP API

### DESCRIPTION

Use the createConferenceSP API to create a new conference with the specified conference information object, ConferenceInfo.Definition. This API has support to create a non-video conference.

### DEFINITION

```
public ScheduleResult createConferenceSP(ConferenceInfo  
info);
```

### PARAMETERS

createConferenceSP must set the following mandatory parameters:

- Start Time
- End Time
- Meeting Type ID
- Organization ID
- User ID
- Terminals
- TestSchedule (mandatory for non-video conferences only)

**Table 3-12** createConferenceSP API Parameter Table

Parameter	Description
Start Time	The conference start time in milliseconds.
End Time	The conference end time in milliseconds.
Meeting Type ID	The conference meeting type ID, referred to as Service Template ID.
Organization ID	Organization ID should not be NULL or “999”. This parameter must be set or iVIEW Communications Manager will throw an exception.
User ID	User ID should not be NULL. In iCM SP, the user needs to be listed in iCM.

**Table 3-12** *createConferenceSP API Parameter Table (continued)*

<b>Parameter</b>	<b>Description</b>
Terminals	The terminals invited by the conference.
Conference ID	iVIEW Communications Manager uses the ID given by the client. If the Conference ID contains an empty string (“”) or NULL, iVIEW Communications Manager assigns an ID automatically. A conflict may happen as the client does not check if the conference ID is already used by the iVIEW Communications Manager.
Classification Name	If the classification parameter contains an empty string or NULL, iVIEW Communications Manager treats the API as if the client does not want a classification and does not return a non-found error code.
TestSchedule	The TestSchedule parameter is mandatory for non-video conferences. It is not used for video conferences. If the parameter is true, the iVIEW Communications Manager system checks the creation of a conference without committing the scheduling result to the iVIEW Communications Manager system and creating a conference. The default is false.
Allow Recording	<p>The Allow Recording parameter allows or disallows recording. This is an old parameter. If it is NULL, iVIEW Communications Manager sets its value according to the iVIEW Communications Manager type:</p> <ul style="list-style-type: none"> <li>■ Enterprise version—Defaults to false</li> <li>■ SP version—iVIEW Communications Manager sets the value according to user policy with specified Organization ID and User ID</li> </ul> <p>Organization ID is 999, and User ID is passed by the application that calls the API.</p>
Record Meeting When Start	Use the Record Meeting When Start parameter to record the conference when it starts. This is a new parameter. If it is NULL, the parameter defaults to false. If the conference is not allowed to be recorded but is recorded when the conference starts, iVIEW Communications Manager throws an exception.

**Table 3-12** *createConferenceSP API Parameter Table (continued)*

Parameter	Description
Allow Streaming	The Allow Streaming parameter allows or disallows streaming. This is a new parameter. If it is NULL, the parameter defaults to false.
Streaming Status	The Streaming Status parameter enables streaming when the conference starts. This is an old parameter.
Recording Quality	The Recording Quality parameter specifies the recording quality (rate in Kbps) of the conference. This is a new parameter. Its value is one of the following enumerations: 128, 192, 256, 384, 512, 768, 1024. If the user does not specify this parameter, SCOPIA Desktop uses the default value (configurable in SCOPIA Desktop admin GUI).
Recording Duration	The Recording Duration parameter specifies the recording duration (in minutes) of the conference. Maximum recording duration is 397 minutes. If the user does not specify this parameter, SCOPIA Desktop uses the default value (configurable in SCOPIA Desktop admin GUI).

## RETURN VALUE

ScheduleResult.errorStatus contains the information related to success or failure, is returned; if failed, the detailed error information is enclosed in it.

When scheduling a conference, the iVIEW Communications Manager system first checks if the specified classification name is invalid or does not exist in the iVIEW Communications Manager system. If it is invalid or does not exist, the ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME = 111 error is returned, or else it is ignored.

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NULL\_MEETING\_TYPE\_ID\_ARGUMENT(107)—The meeting type ID is not defined.
- ERROR\_NOT\_FOUND\_MEETING\_TYPE(22)—The specified meeting type is not found.
- ERROR\_INACTIVE\_MEETING\_TYPE(23)—The specified meeting type is inactive.

- **ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME(111)**—The classification is not found.
- **ERROR\_MEETING\_CONTROL(99)**—The conference cannot be created because there are no available resources in the devices (such as MCU or SCOPIA Gateway), or the conference has already been created.

For the other error code, see the [Error Codes](#) appendix.

---

## modifyConference API

### DESCRIPTION

Use the modifyConference API to modify the next scheduled conference in the system with the specified conference information object, ConferenceInfo. iVIEW Communications Manager re-schedules the existing conference with the new conference information. The modifyConference API supports non-video conferences.

### DEFINITION

```
public ScheduleResult modifyConference(ConferenceInfo  
newinfo);
```

### PARAMETERS

The ConferenceInfo object must set the mandatory parameters:

- [Conference ID](#) (mandatory for non-video conferences only)
- [Start Time](#)
- [End Time](#)
- [Meeting Type ID](#)
- [Terminals](#)
- [TestSchedule](#) (mandatory for non-video conferences only)

---

**Note** The default value or null may be used for other parameters not mentioned above, for example, the required level parameter uses a default value of 4 (REQUIRED\_ALL\_ROOMS\_AND\_PARTIES) if not specified, and the layout list parameter is set to null if not specified.

---

**Table 3-13** *modifyConference API Parameter Table*

Parameter	Description
Conference ID	If the conference ID is not found, the operation fails and returns the error to the client.
Start Time	The conference start time in milliseconds.

---

**Table 3-13**     *modifyConference API Parameter Table (continued)*

<b>Parameter</b>	<b>Description</b>
End Time	The conference end time in milliseconds.
Meeting Type ID	The conference meeting type ID, referred to as Service Template ID.
Terminals	The terminals invited by the conference. The conference makes no sense without them.
Classification Name	If the classification parameter contains an empty string or NULL, iVIEW Communications Manager treats the API as if the client does not want a classification and does not return a non-found error code.
TestSchedule	The TestSchedule parameter is mandatory for non-video conferences. It is not used for video conferences. If the parameter is true, iVIEW Communications Manager checks the creation of a conference without committing the scheduling result to iVIEW Communications Manager and creating a conference. The default is false.

## RETURN VALUE

When modifying a conference, iVIEW Communications Manager first checks if the specified classification name is invalid or does not exist in iVIEW Communications Manager. If it is invalid or does not exist, the `ERROR_SVR_INVALID_CLASSIFICATION_NAME = 111` error is returned, or else it is ignored.

Error code in `ScheduleResult.errorStatus`:

- `ERROR_NOT_SUPPORT(7)`—The conference specified by the conference ID does not exist.
- `ERROR_SVR_NULL_MEETING_TYPE_ID_ARGUMENT(107)`—The meeting type ID is not defined.
- `ERROR_NOT_FOUND_MEETING_TYPE(22)`—The specified meeting type is not found.
- `ERROR_INACTIVE_MEETING_TYPE(23)`—The specified meeting type is inactive.

**ScheduleService**  
modifyConference API

- **ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME(111)**—The classification is not found.
- **ERROR\_MEETING\_CONTROL(99)**—The conference cannot be created because there are no available resources in the devices (such as MCU or SCOPIA Gateway), or the conference has already been created.

For the other error code, see the [Error Codes](#) appendix.

---

## modifyConferenceSP API

### DESCRIPTION

Use the modifyConferenceSP API to modify the next scheduled conference in the system with the specified conference information object, ConferenceInfo. iVIEW Communications Manager reschedules the existing conference with the new conference information. The modifyConference API supports non-video conferences. The modifyConferenceSP API has more parameters than the modifyConference API.

### DEFINITION

```
public ScheduleResult modifyConferenceSP(ConferenceInfo
newinfo);
```

### PARAMETERS

The ConferenceInfo object must set the mandatory parameters:

- Organization ID
- User ID
- Conference ID
- Start Time
- End Time
- Meeting Type ID
- Terminals
- TestSchedule (mandatory for non-video conferences only)

**Table 3-14** *modifyConferenceSP API Parameter Table*

Parameter	Description
Organization ID	Should not be NULL or “999”.
User ID	User ID should not be NULL. In iCM SP, the user needs to be listed in iCM.
Conference ID	If the conference ID is not found, the operation fails and returns the error to the client.

---

**Table 3-14** *modifyConferenceSP API Parameter Table (continued)*

Parameter	Description
Start Time	The conference start time in milliseconds.
End Time	The conference end time in milliseconds.
Meeting Type ID	The conference meeting type ID, referred to as Service Template ID.
Terminals	The terminals invited by the conference. The conference makes no sense without them.
Classification Name	If the classification parameter contains an empty string or NULL, iVIEW Communications Manager treats the API as if the client does not want a classification and does not return a non-found error code.
TestSchedule	The TestSchedule parameter is mandatory for non-video conferences. It is not used for video conferences. If the parameter is true, iVIEW Communications Manager checks the creation of a conference without committing the scheduling result to iVIEW Communications Manager and creating a conference. The default is false.
Allow Recording	<p>The Allow Recording parameter allows or disallows recording. This is an old parameter. If it is NULL, iVIEW Communications Manager sets its value according to iVIEW Communications Manager types:</p> <ul style="list-style-type: none"><li>■ Enterprise version—Defaults to false</li><li>■ SP version—iVIEW Communications Manager sets the value according to user policy with specified Organization ID and User ID</li></ul> <p>Organization ID is 999, and User ID is passed by the application that calls the API</p>

**Table 3-14**     *modifyConferenceSP API Parameter Table (continued)*

<b>Parameter</b>	<b>Description</b>
Record Meeting When Start	The Record Meeting When Start parameter causes the conference to be recorded when it starts. This is a new parameter. If it is NULL, the parameter defaults to false. If the conference is not allowed to be recorded but is recorded when the conference starts, iVIEW Communications Manager throws an exception.
Allow Streaming	The Allow Streaming parameter allows or disallows streaming. This is a new parameter. If it is NULL, the parameter defaults to false.
Streaming Status	The Streaming Status parameter enables streaming when the conference starts. This is an old parameter.
Recording Quality	The Recording Quality parameter specifies the recording quality (rate in Kbps) of the conference. This is a new parameter. Its value is one of the following enumerations: 128, 192, 256, 384, 512, 768, 1024. If the user does not specify this parameter, SCOPIA Desktop uses the default value (configurable in SCOPIA Desktop admin GUI).
Recording Duration	The Recording Duration parameter specifies recording duration (rate in minutes) of the conference. Max recording duration is 397 minutes. If the user does not specify this parameter, SCOPIA Desktop uses the default value (configurable in SCOPIA Desktop admin GUI).

### **RETURN VALUE**

When modifying a conference, iVIEW Communications Manager first checks if the specified classification name is invalid or does not exist in iVIEW Communications Manager. If it is invalid or does not exist, the `ERROR_SVR_INVALID_CLASSIFICATION_NAME = 111` error is returned, or else it is ignored.

Error code in ScheduleResult.errorStatus:

- ERROR\_NOT\_SUPPORT(7)—The conference specified by the conference ID does not exist.
- ERROR\_SVR\_NULL\_MEETING\_TYPE\_ID\_ARGUMENT(107)—The meeting type ID is not defined.
- ERROR\_NOT\_FOUND\_MEETING\_TYPE(22)—The specified meeting type is not found.
- ERROR\_INACTIVE\_MEETING\_TYPE(23)—The specified meeting type is inactive.
- ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME(111)—The classification is not found.
- ERROR\_MEETING\_CONTROL(99)—The conference cannot be created because there are no available resources in the devices (such as MCU or SCOPIA Gateway), or the conference has already been created.

For the other error code, see the [Error Codes](#) appendix.

---

## cancelConference API

### DESCRIPTION

Use the cancelConference API to cancel the next scheduled conference in the system with the specified conference ID.

### DEFINITION

```
public boolean cancelConference(String conferenceID);
```

### PARAMETERS

**Table 3-15** *cancelConference API Parameter Table*

Parameter	Description
conferenceID	

### RETURN VALUE

Return true if the conference is cancelled successfully, false otherwise.

---

## cancelConferences API

### DESCRIPTION

Use the cancelConferences API to cancel the following scheduled conferences in the system with the specified conference IDs.

### DEFINITION

```
public List<ScheduleResult> cancelConferences(List<String>  
conferenceIDs);
```

### PARAMETERS

**Table 3-16** *cancelConferences API Parameter Table*

Parameter	Description
conferenceIDs	A list that contains the conference IDs

### RETURN VALUE

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NOT\_FOUND\_CONFERENCE(112)—The conference does not exist
- ERROR\_SVR\_CANCEL\_ALREADY\_CANCELLED\_CONFERENCE(113)—The conference has been already canceled
- ERROR\_SVR\_CANCEL\_ALREADY\_FINISHED\_CONFERENCE(114)—The Conference has been terminated
- ERROR\_SVR\_CANCEL\_BACKUP\_CONFERENCE(115)—This conference is a backup conference and cannot be cancelled
- ERROR\_SVR\_CANCEL\_SCHEDULE\_FAILED\_CONFERENCE(116)—This conference is failed to be launched at the start time and cannot be cancelled.

For the other error code, see the [Error Codes](#) appendix.

---

## viewConference API

### DESCRIPTION

Use the viewConference API to display the following scheduled conferences in the system with the specified conference IDs. An existing conference information object is received.

### DEFINITION

```
public ConferenceInfo viewConference(String  
conferenceID) throws  
InvalidArgumentException, NotConferenceFoundException;
```

### PARAMETERS

**Table 3-17** *viewConference API Parameter Table*

---

Parameter	Description
conferenceID	

---

### RETURN VALUE

Exception:

- InvalidAugrmentException—The conferenceID is null
- NotConferenceFoundException—The specified conference not found.

---

## checkAlternateTerminals API

### DESCRIPTION

Use the checkAlternateTerminals API to check the alternative terminals for the given terminals in the specified time frame.

### DEFINITION

```
public List<AlternateTerminalsTimedSlice>  
checkAlternateTerminals(List<String> terminalIDs, long  
startTime, long endTime, long cycleInterval) throws  
InvalidArgumentException, InvalidTerminalIdException;
```

### PARAMETERS

**Table 3-18** *checkAlternateTerminals API Parameter Table*

Parameter	Description
terminalIDs	
startTime	
endTime	
cycleInterval	

### RETURN VALUE

Exception:

- `InvalidArgumentException`—The required parameter is not supplied. Using `InvalidArgumentException.getMessage()` to know the name of the parameter.
- `InvalidTerminalIdException`—The terminals are not found.

---

## checkAlternativeTimes API

### DESCRIPTION

Use the checkAlternateTimes API to check if an upcoming conference can be held with the specified conference information object, the start time, the end time and the cycle interval.

### DEFINITION

```
public List<AvailableInfo>  
checkAlternativeTimes(ConferenceInfo newConference, long  
startTime, long endTime, long cycleInterval)throws  
InvalidArgumentException;
```

### PARAMETERS

**Table 3-19** *checkAlternativeTimes API Parameter Table*

Parameter	Description
ConferenceInfo	
newConference	
startTime	
endTime	
cycleInterval	

### RETURN VALUE

Exception:

- **InvalidArgumentException**—The required parameter is not supplied. Using `InvalidArgumentException.getMessage()` to know the name of the parameter.

---

## createRecurrence API

### DESCRIPTION

Use the createRecurrence API to create an upcoming recurrent conference by only specifying the recurrence information. Each recurrent conference instance uses the same conference information except start and end times. The recurrence information specifies the necessary parameters of the recurrent conference, such as the conference information template object, the recurrence instance times array object and its ID. After executing this operation, iVIEW Communications Manager returns the schedule result list that contains all the schedule results for each recurrent conference and the recurrence information is passed in from the argument. The recurrence information has been filled in all the conference instances modified by iVIEW Communications Manager and its ID is generated anew.

The recurrence information object as the input parameter must specify the attributes as follows:

- Conference information template object in which the start time and end time settings are not included, refer to the createConference API for settings to other mandatory parameters.
- Recurrence instance information array object in which only the start time and end times are specified for each instance.

### DEFINITION

```
public List<ScheduleResult> createRecurrence(RecurrenceInfo rinfo);
```

## PARAMETERS

**Table 3-20** *createRecurrence API Parameter Table*

Parameter	Description
RecurrenceInfo	<p>The recurrence info object as the input parameter must specify these attributes:</p> <ul style="list-style-type: none"> <li>■ Conference info template object in which the start/end time setting is not included. Refer to the createConference API for settings to other mandatory fields.</li> <li>■ Recurrence instance infos array object in which only the start/end time is specified for each instance.</li> </ul>

## RETURN VALUE

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NULL\_RECURRENCE\_INSTANCE\_INFOS\_ARGUMENT(103)—No Recurrences are supplied
- ERROR\_SVR\_GET\_NEXT\_RECURRENCE\_ID(105)—Failed to generate the identifier for the recurrence conference
- ERROR\_SVR\_NULL\_CONFERENCE\_INFO\_TEMPLATE\_ARGUMENT(102)—No Conference is supplied
- ERROR\_SVR\_NULL\_CONFERENCE\_INFO\_ARGUMENT(108)—
- ERROR\_SVR\_NULL\_MEETING\_TYPE\_ID\_ARGUMENT(107)—The Meeting type ID is not defined
- ERROR\_NOT\_FOUND\_MEETING\_TYPE(22)—The specified meeting type is not found
- ERROR\_INACTIVE\_MEETING\_TYPE(23)—The Specified meeting type is inactive
- ERROR\_SVR\_INVALID\_CLASSIFICATION\_NAME(111)—The classification is not found
- ERROR\_SVR\_SET\_PUBLIC\_LOCAL\_ACCESS\_URL(104)—The URL to access this recurrence is not set

For the other error code, see the [Error Codes](#) appendix.

---

## modifyRecurrence API

### DESCRIPTION

Use the modifyRecurrence API to modify an upcoming recurrent conference by only specifying the recurrence information and giving the old recurrent conference ID. This method will have no effect if you do not give the old recurrent conference ID.

After executing this operation, iVIEW Communications Manager returns the schedule result list that contains all the schedule results for each recurrent conference and the recurrence information is passed in from the argument. The recurrence information has been filled in all the conference instances modified by iVIEW Communications Manager and its old ID.

The recurrence information object as the input parameter must specify the attributes as follows:

- Recurrence ID must be specified
- Conference information template object in which the start/end time setting is not included, refer to the createConference API for settings to other mandatory parameters
- Recurrence instance information array object in which only the start/end time and instance ID are specified for each instance

---

**Note** It supports two features: one is to modify recurrence and commit its result, and the other is only to check to modify recurrence but not commit its result. The switch flag supporting these two features is the TestSchedule parameter in the ConferenceInfo template class.

---

### DEFINITION

```
public List<ScheduleResult> modifyRecurrence(RecurrenceInfo rinfo);
```

## PARAMETERS

**Table 3-21** *modifyRecurrence API Parameter Table*

Parameter	Description
RecurrenceInfo	<p>The recurrence info object as the input parameter must specify these attributes:</p> <ul style="list-style-type: none"> <li>■ Recurrence id must be specified.</li> <li>■ Conference info template object in which the start/end time setting is not included. Refer to the createConference API for the settings to other mandatory fields.</li> <li>■ Recurrence instance infos array object in which the start/end time and instance id are specified for each instance.</li> </ul>

## RETURN VALUE

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NULL\_RECURRENCE\_INFO\_ARGUMENT(100)—The parameter is null.
- ERROR\_SVR\_NULL\_RECURRENCE\_ID\_ARGUMENT(101)—The recurrence ID is null.
- ERROR\_SVR\_NULL\_CONFERECE\_INFO\_TEMPLATE\_ARGUMENT(102)—No conference data provided.
- ERROR\_SVR\_NULL\_RECURRENCE\_INSTANCE\_INFOS\_ARGUMENT(103)—No recurrences are supplied.

For the other error code, see the [Error Codes](#) appendix.

---

## cancelRecurrence API

### DESCRIPTION

Use the cancelRecurrence API to cancel an upcoming recurrent conference by specifying the recurrence information and conference ID. After executing this operation, iVIEW Communications Manager returns the schedule result list that contains all the schedule results for each recurrent conference. Each schedule result has a flag attribute to identify if canceling this recurrent conference succeeded or failed.

### DEFINITION

```
public List<ScheduleResult> cancelRecurrence(String  
recurrenceID);
```

### PARAMETERS

**Table 3-22** *cancelRecurrence API Parameter Table*

---

Parameter	Description
recurrenceID	

---

### RETURN VALUE

Error code in ScheduleResult.errorStatus:

- ERROR\_SVR\_NULL\_RECURRENCE\_ID\_ARGUMENT(101)—The recurrence ID is null

For the other error code, see the [Error Codes](#) appendix.

---

## viewRecurrence API

### DESCRIPTION

Use the viewRecurrence API to view an upcoming recurrent conference by specifying the recurrence information and conference ID. After executing this operation, iVIEW Communications Manager returns the schedule result list that contains all the schedule results for each recurrent conference. Each schedule result has a flag attribute to identify if canceling this recurrent conference succeeded or failed.

### DEFINITION

```
public RecurrenceInfo viewRecurrence (String  
recurrenceID)throws IllegalArgumentException;
```

### PARAMETERS

**Table 3-23** *viewRecurrence API Parameter Table*

---

Parameter	Description
recurrenceID	

---

### RETURN VALUE

---

## createVirtualRoom API

This API method aims for creating a new virtual room with the specified virtual room information object (namely VirtualRoomInfoEx) and this argument object needs to set those mandatory attributes such as meeting type id, member id, user id and etc. When this operation is done, a schedule result (namely VirtualRoomResult), which contains the information related to success or failure, will be returned; if failed, the detailed error information will be enclosed in it.

As for the generic conference, below list the common fields or attributes need to be specified:

- Virtual room Id [Optional]: not needed, the iCM will assign one automatically.
- Virtual room Name [Mandatory]: the virtual room name.
- Virtual room Number [Mandatory]: the virtual room number.
- Meeting Type ID [Mandatory]: the meeting type id (called as "service template id") of the virtual room.
- Member ID [Optional]: the member id. If it is not specified, the value will be set as default value in iCM. When iView is for ENT version, the value will be set as "999".
- User ID [Mandatory]: the user id.
- Description [Optional]: the virtual room description.
- Time Zone [Optional]: the time zone id; if an empty string "" or NULL, the iCM will assign the local time zone id; otherwise, the iCM uses the id given by the client.
- Outsiders [Optional]: the outsiders invited by the virtual room; this virtual room makes no sense if none.
- Attendees [Optional]: the attendees invited by the virtual room; this virtual room makes no sense if none.

---

**Note** The default value or null may be used for other fields not mentioned above.

---

### DEFINITION

```
public VirtualRoomResult createVirtualRoom(VirtualRoomInfoEx  
virtualRoomInfo);
```

VirtualRoomResult.errorStatus:

- ERROR\_SVR\_NULL\_SERVICE\_ID\_ARGUMENT(4): The meeting type id is not defined.
- ERROR\_INACTIVE\_MEETING\_TYPE(1): The specified meeting type is not found.
- ERROR\_MEMBER\_NOT\_FOUND\_IN\_ICM(12): The specified member id is not found in iCM.
- ERROR\_USERID\_NOT\_FOUND\_IN\_ICM (13): The specified user id is not found in iCM.

For the other error codes, see the [Error Codes](#) appendix.

---

## modifyVirtualRoom API

This API method aims for modifying a virtual room with the specified virtual room information object (namely VirtualRoomInfoEx) and this argument object needs to set those mandatory attributes such as virtual room id, meeting type id, member id, user id and etc. When this operation is done, a schedule result (namely VirtualRoomResult), which contains the information related to success or failure, will be returned; if failed, the detailed error information will be enclosed in it.

As for the generic conference, below list the common fields or attributes need to be specified:

- Virtual room Id [Mandatory]: the virtual room id; If not found the conference with the given id, the operation to modify the virtual room fails and returns the error to the client.
- Virtual room Name [Mandatory]: the virtual room name.
- Virtual room Number [Mandatory]: the virtual room number.
- Meeting Type ID [Mandatory]: the meeting type id (called as “service template id”) of the virtual room.
- Member ID [Optional]: the member id. If it is not specified, the value will be set as default value in iCM. When iView is for ENT version, the value will be set as “999”.
- User ID [Mandatory]: the user id.
- Description [Optional]: the virtual room description.
- Time Zone [Optional]: the time zone id; if an empty string “” or NULL, the iCM will assign the local time zone id; otherwise, the iCM uses the id given by the client.
- Outsiders [Optional]: the outsiders invited by the virtual room; this virtual room makes no sense if none.
- Attendees [Optional]: the attendees invited by the virtual room; this virtual room makes no sense if none.

---

**Note** The default value or null may be used for other fields not mentioned above.

---

### Definition

```
public VirtualRoomResult modifyVirtualRoom(VirtualRoomInfoEx  
virtualRoomInfo);
```

VirtualRoomResult.errorStatus:

- ERROR\_SVR\_NULL\_SERVICE\_ID\_ARGUMENT(4): The meeting type id is not defined.
- ERROR\_INACTIVE\_MEETING\_TYPE(1): The specified meeting type is not found.
- ERROR\_MEMBER\_NOT\_FOUND\_IN\_ICM(12): The specified member id is not found in iCM.
- ERROR\_USERID\_NOT\_FOUND\_IN\_ICM (13): The specified user id is not found in iCM.

For the other error codes, see the [Error Codes](#) appendix Appendix Error Code.

---

## **deleteVirtualRoom API**

This API method aims for deleting a virtual room with the specified virtual room number.

### **DEFINITION**

```
public VirtualRoomResult deleteVirtualRoom(String  
virtualRoomNumber);
```

---

## viewVirtualRooms API

### DESCRIPTION

Use the viewVirtualRooms API to view virtual rooms of a specific user. The user's login-ID and member-ID are used.

### DEFINITION

```
public Collection<VirtualRoomInfo> viewVirtualRooms(String  
loginID, String memberID)throws InvalidArgumentException,  
VirtualRoomNotFoundException, UserNotExistException;
```

### PARAMETERS

**Table 3-24** *viewVirtualRooms API Parameter Table*

---

Parameter	Description
loginID	The login id of the user whose virtual room will be returned.
memberID	The member id of the user whose virtual room will be returned.

---

### RETURN VALUE

It returns a collection of virtual room objects.

---

## viewVirtualRoomsByUserID API

### DESCRIPTION

Use the viewVirtualRoomsByUserID API to view virtual rooms of a specific user ID.

### DEFINITION

```
public Collection<VirtualRoomInfo>  
viewVirtualRoomsByUserId(String userID, String memberID)  
throws InvalidArgumentException,  
VirtualRoomNotFoundException;
```

### PARAMETERS

**Table 3-25** *viewVirtualRoomsByUserID API Parameter Table*

Parameter	Description
userID	The login id of the user whose virtual room will be returned.
memberID	The member id of the user whose virtual room will be returned.

### RETURN VALUE

It returns a collection of virtual room objects.

---

## viewVirtualRoomByName API

### DESCRIPTION

Use the viewVirtualRoomsByName API to view the data of a virtual room according to the room's ID.

### DEFINITION

```
public VirtualRoomInfo viewVirtualRoomById(String  
virtualRoomID) throws InvalidArgumentException,  
VirtualRoomNotFoundException;
```

### PARAMETERS

**Table 3-26** *viewVirtualRoomByName API Parameter Table*

---

Parameter	Description
virtualRoomID	

---

### RETURN VALUE

It returns a VirtualRoomInfo object.

---

## viewVirtualRoomByID API

### DESCRIPTION

Use the viewVirtualRoomByID API to view the data of a specific virtual room according to the room's ID.

### DEFINITION

```
public VirtualRoomInfo viewVirtualRoomByID(String  
virtualRoomID) throws InvalidArgumentException,  
VirtualRoomNotFoundException;
```

### PARAMETERS

**Table 3-27** *viewVirtualRoomByID API Parameter Table*

---

Parameter	Description
virtualRoomID	

---

### RETURN VALUE

It returns a VirtualRoomInfo object.

---

## viewVirtualRoomByDialableNumber API

### DESCRIPTION

Use the viewVirtualRoomByDialableNumber API to view the data of a specific virtual room according to the room's dialable number (virtual ID). You can include the service prefix in the dialable number but it is not required.

### DEFINITION

```
public VirtualRoomInfo  
viewVirtualRoomByDialableNumber(String dialableNumber)  
throws InvalidArgumentException,  
VirtualRoomNotFoundException;
```

### PARAMETERS

**Table 3-28** *viewVirtualRoomByDialableNumber API Parameter Table*

Parameter	Description
dialableNumber	

### RETURN VALUE

It returns a VirtualRoomInfo object.

---

## viewConferenceByDialableNumber API

### DESCRIPTION

Use the viewConferenceByDialableNumber API to view the data of a specific conference according to the conference's dialable number (virtual ID). You can include the service prefix in the dialable number but it is not required.

### DEFINITION

```
public Collection viewConferenceByDialableNumber(String  
dialableNumber) throws  
InvalidArgumentException, NotConferenceFoundException;
```

### PARAMETERS

**Table 3-29** *viewConferenceByDialableNumber API Parameter Table*

---

Parameter	Description
dialableNumber	

---

### RETURN VALUE

It returns a VirtualRoomInfo object.

---

## getDialedNumberConferenceInfo API

### DESCRIPTION

Use the getDialedNumberConferenceInfo API to view the potential conference information of a dialed number.

The dialable number is any number that can be dialed to join the conference. It may include the service prefix and a zone prefix.

This API replaces the API viewConferenceOrVirtualRoomByDialableNumber.

---

### Note

- This API does not return any information concerning past or upcoming conferences.
- This API is used by the SCOPIA Desktop Server.

---

### DEFINITION

```
public ConferenceInfo getDialedNumberConferenceInfo (String  
dialableNumber) throws InvalidArgumentException,  
ServiceMismatchException, NoDefaultServiceException,  
UpcomingConferenceException,  
UpcomingConferenceConflictException,  
NotConferenceFoundException;
```

### PARAMETERS

**Table 3-30** *getDialedNumberConferenceInfo API Parameter Table*

---

Parameter	Description
dialableNumber	

---

## RETURN VALUE

- If the dialed number matches an ongoing conference, it returns a ConferenceInfo object with the conference information.
- If the dialed number matches a virtual room, it returns a VirtualRoomInfo object with the virtual room's information (VirtualRoomInfo is derived from ConferenceInfo).
- If the dialed number does not match either, and the dialed number does not conflict with other conferences, it returns a ConferenceInfo object with the default ad-hoc conference settings.

## Exceptions:

- InvalidArgumentException—If no dialableNumber was provided.
- ServiceMismatchException—If the dialed service prefix does not match the conference or virtual room.
- UpcomingConferenceException—If a matching conference is just about to start (configurable, default is 5 minutes).
- UpcomingConferenceConflictException—If an upcoming scheduled conference that uses this dialableNumber or some dialableNumber that conflicts with it was found (this depends on the default length of ad-hoc conferences, default is 30 minutes).
- NoDefaultServiceException—If the dialableNumber does not start with a service prefix and there is no default service configured in iVIEW Communications Manager.
- NotConferenceFoundException—If none of the above occurs and the number still can't initiate a new conference, for example it does not start with a service prefix or the iVIEW Communications Manager prefix.

---

## getVersion API

### DESCRIPTION

Use the getVersion API to get the web services version. This version can be set in icm-ws.properties.

### DEFINITION

```
Public String getVersion()
```

---

## searchConferences API

### DESCRIPTION

Use the searchConferences API to search for the conferences that meet the specified condition.

### DEFINITION

```
public List<ConferenceInfo>  
searchConferences(ConferenceInfoCondition condition)
```

### PARAMETERS

#### Condition

The condition for the search, see [ConferenceInfoCondition Class](#) on page 121.

**Table 3-31** *searchConferences API Parameter Table*

---

Parameter	Description
Condition	The condition for the search, see <a href="#">ConferenceInfoCondition Class</a> on page 121.

---

### RETURN VALUE

The conferences that meet the specified condition.

---

## deleteHistoryConferences API

### DESCRIPTION

Use the deleteHistoryConferences API to delete the conference history.

### DEFINITION

```
public ScheduleResult deleteHistoryConferences(List<String>  
conferenceIDs)
```

### PARAMETERS

**Table 3-32** *deleteHistoryConferences API Parameter Table*

---

Parameter	Description
ConferenceIDs	The conference histories of this list of IDs is deleted.

---

### RETURN VALUE

Error code in ScheduleResult.success:

- true—If Successful
- false—If not successful

Error code in ScheduleResult.errorStatus:

- 109—Some of the conference IDs are null
- 117—Failed, no conferences are removed

---

## removeConference API

Remove conference from system(including database).

### DEFINITION

```
public boolean removeConference(String conferenceId) throws  
InvalidArgumentException
```

### PARAMETERS

**Table 3-33** *removeConference API Parameter Table*

Parameter	Description
ConferenceIDs	

### RETURN VALUE

True if the conference is removed successfully, false otherwise.

### Exceptions

- `InvalidArgumentException`—`conferenceId` is empty

## USER SERVICE

---

### authUser API

#### DESCRIPTION

Use the authUser API to authenticate users with iVIEW Suite.

#### DEFINITION

```
public boolean authUser(String loginId,String pwd);
```

#### PARAMETERS

**Table 3-34** *authUser API Parameter Table*

Parameter	Description
loginId	The user's log-in ID.
Pwd	The user's password.

#### RETURN VALUE

Error code

- True—If the user is authenticated successfully.
- False—If the user is not authenticated successfully.

---

## setUsers API

### DESCRIPTION

Use the setUsers API to create or update user profiles. The created and updated user profiles are echoed back in the instance of UserResult so that the client is informed of the user IDs of the newly created user profiles.

This operation will be broken when one of the users of the input list failed to be added/updated to iVIEW. Once one fails, all will roll back and return error for this user.

### DEFINITION

```
public UserResult setUsers (List<UserInfo> userInfos)
```

### PARAMETERS

**Table 3-35**     *setUsers API Parameter Table*

Parameter	Description
UserInfos	The user profiles to be created or updated.
UserInfo.userID	If null, the user profile is created, otherwise the user profile is updated.

### RETURN VALUE

Error code in UserResult.success:

- true—Success
- false—Fail

Error code in UserResult.errorCode:

- 100—The default terminal does not existed
- 101—The last name not specified
- 107—User can not be created
- 108—User can not be updated

---

## deleteUsers API

### DESCRIPTION

Use the deleteUsers API to delete a list of specified user profiles.

This operation will be broken when one of the users of the input list failed to be deleted from iVIEW. Once one fails, all will roll back and return error for this user.

### DEFINITION

```
public UserResult deleteUsers (List<String> userIDs)
```

### PARAMETERS

**Table 3-36** *deleteUsers API Parameter Table*

Parameter	Description
UserIDs	The list of the identifiers of user profiles to be deleted.

### RETURN VALUE

Error code in UserResult.success:

- true—Success
- false—Fail

---

## searchUsers API

### DESCRIPTION

Use the searchUsers API to search for a specific first name, last name and login ID.

### DEFINITION

```
public List<UserInfo> searchUsers (UserSearchConditon  
condition)
```

### PARAMETERS

**Table 3-37** *searchUsers API Parameter Table*

---

Parameter	Description
Condition	Define what to search for. See <a href="#">UserSearchConditon Class</a> on page 128.

---

### RETURN VALUE

A list of users that meets the condition.

---

## getUserByLoginID API

### DESCRIPTION

Use the getUserByLoginID API to get a user's detailed information by the login ID.

### DEFINITION

```
public UserInfo getUserByLoginID(String loginID)
    throws UserNotExistedException
```

### PARAMETERS

**Table 3-38** *getUserByLoginID API Parameter Table*

Parameter	Description
LoginID	The login ID of the user.

### RETURN VALUE

The detailed information of the user.

### Exceptions:

- UserNotExistedException—Thrown when the user with the log-in ID does not exist

---

## getUsers API

### DESCRIPTION

Use the getUsers API to retrieve a list of user profiles with specified user identifiers.

If an error occurs at the iVIEW side while retrieving the users, the list of users will not be returned. The returned result only includes those users that are retrieved from iVIEW.

### DEFINITION

```
public List<UserInfo> getUsers (List<String> userIDs)
```

### PARAMETERS

**Table 3-39** *getUsers API Parameter Table*

Parameter	Description
UserIDs	The list of user identifiers.

### RETURN VALUE

The list of users.

## DIRECTORYUSER SERVICE

This set of API is used for the client to add a set of users from a directory server to iVIEW. Since there might be lot of users from a specific directory server, in order to prevent transferring too many users in one web server request to iVIEW the set of API firstly let the client cache part of the users in iVIEW. When all the users from the directory server have been sent to iVIEW, the client then tells iVIEW to synchronize its database with the supplied directory users.

As the client is supposed to use several requests to send the users to the client, the client firstly uses prepareSession() method to get an operation session. Then the client uses prepareUsers() to send users to iVIEW. iVIEW will cache the users in its memory. The client can invoke prepareUsers() many times until all the users from a directory server are sent to iVIEW. At last, the client invokes submit() to make iVIEW synchronize the users in its cache with its database or cancel() to cancel the operation. iVIEW will clean up its cache and invalidate the session.

---

## prepareSession API

Establish a session for the LDAP operation

### DEFINITION

```
public DirectoryOperationSession prepareSession(  
DirectoryOperationSession session ) throws  
DirectoryServerNotFoundException,  
OrganizationNotFoundException, CookieDuplicateException ;
```

### PARAMETERS

session - the data to establish a session. See the description of DirectoryOperatoinSession.

### RETURN

A data that indicate the established session.

### EXCEPTION

- DirectoryServerNotFoundException - the directory server in the session data is not found in iVIEW.
- OrganizationNotFoundException - the organization in the session data is not found in iVIEW.
- CookieDuplicateException - a cookie value supplied in the session data is duplicated.

---

## prepareUsers API

The API enables iVIEW to cache the supplied users.

### DEFINITION

```
public void prepareUsers( DirectoryOperationSession session,  
List<DirectoryUserInfo> users ) throws  
SessionNotExistedException ;
```

### PARAMETERS

- session - the session data returned by prepareSession() .
- users - the users downloaded by the client from the directory server indicated by the session.

### EXCEPTION

SessionNotExistedException - the session is invalid.

---

## submit API

Tell iVIEW to synchronize its database with the users supplied by `prepareUsers()`. This invocation also invalidates the session.

When iVIEW synchronizes its database with the cached users in its memory:

- If a user does not exist in DB but in the cache, the user is added.
- If a user exists in DB but not in the cache, the user is deleted.
- If a user exists both in the cache and DB, the user is updated if it is changed.

### DEFINITION

```
public void submit( DirectoryOperationSession session )  
throws DirectoryOperationFailedException ;
```

### PARAMETERS

`session` - the session returned by `prepareSession()`.

### EXCEPTION

`DirectoryOperatoinFailedException` - error occurs when iVIEW synchronize its database with the supplied users.

---

## cancel API

Cancel the session. This will invalidate the operation session and clean up the cached data.

### DEFINITION

```
public void cancel(DirectoryOperationSession session)
```

### PARAMETERS

session - the session returned by prepareSession().

---

**Note** When these APIs are used to synchronize users from a directory:

- The directory server should first be configured in iVIEW by the LDAP Configuration page. Only base information (host, login and password) needs to be configured. Because the user will not be downloaded by iVIEW, the search bases and the other settings need not to be set.
  - The users will be updated if the users have been downloaded by iVIEW.
-

# 4

## ASSOCIATED CLASS DEFINITIONS

---

This section introduces all the associated classes with these four web services, including the descriptions of their parameters and constant fields.

- [ResourceService/ResourceV2Service](#) on page 76
- [ScheduleService](#) on page 87
- [ControlService](#) on page 122
- [UserService](#) on page 127
- [DirectoryUser Service](#) on page 128

**RESOURCESERVICE/  
RESOURCEV2SERVICE**

---

## MeetingType Class

The meeting type is actually the service supported by the MCU. It contains parameters such as the available screen layouts and the available views.

---

**Note** The default MCU version is RADVISION version 3.0. At present each meeting type has the same available screen layouts.

---

**Table 4-1** MeetingType Class Parameter Table

Parameter	Data Type	Default Value	Description
ID	String (32 characters)	NULL	iVIEW Communications Manager has the ID of the meeting type, not the physical MCU.
Name	String (64 characters)	NULL	The meeting type name.
ServicePrefix	String (32 characters)	NULL	The service prefix known by physical MCU is to identify this meeting type (MCU service).
Description	String (255 characters)	NULL	The meeting type description.
McuService	Integer	AUDIO_VIDEO_SERVICE	This parameter distinguishes the kind of service type. See <a href="#">Table 4-2</a> for descriptions.
Resolution	String (32 characters)	NONE	This parameter specifies service resolution or picture size.
SwitchingMode	Integer	NONE_SWITCHING_MODE	Switching mode of TP service: the first one is switching by site, the second one is switching by segment and the last is none mode.
Bandwidth	Integer	0	Service bandwidth in KB, for example, 2048 KB.
List<Layout>	List<Layout>	NULL	All the available screen layouts supported by this meeting type.

**Table 4-1** MeetingType Class Parameter Table (continued)

Parameter	Data Type	Default Value	Description
List<View>	List<View>	NULL	All the available views supported by this meeting type.

**Table 4-2** MeetingType Class Constant Field Table

Constant Field	Value	Description
BUILT_IN_SERVICE	0	Meeting type supporting built-in service.
AUDIO_ONLY_SERVICE	1	Meeting type supporting audio-only service.
AUDIO_VIDEO_SERVICE	2	Meeting type supporting audio-video service.
TELE_PRESENCE_SERVICE	3	Meeting type supporting TelePresence service.
NONE_SWITCHING_MODE	1	None switching mode.
SWITCHING_BY_SITE	2	Switching mode by site.
SWITCHING_BY_SEGMENT	3	Switching mode by segment.

---

## Layout Class

This object is the screen layout supported by the MCU. The current RADVISION MCU is version 3.0. Note that each version of the same MCU has different screen layouts, and the LayoutType parameter is used to identify a screen layout type.

**Table 4-3**      *Layout Class Parameter Table*

Parameter	Data Type	Default Value	Description
Index	Integer	-1	The screen layout index.
LayoutType	String (32 characters)	NULL	The screen layout type.
Description	String (60 characters)	NULL	The screen layout description.

---

## View Class

Each vendor's MCU, for example RADVISION's MCU version 3.0, has a series of services (meeting types), and each meeting type has some views which define the parameters such as initial video layout and maximum video layout.

**Table 4-4** *View Class Parameter Table*

Parameter	Data Type	Default Value	Description
ViewID	String (32 characters)	NULL	The view ID.
InitialVideoLayout	String (32 characters)	NULL	The initial video layout. The value is taken from the layout type parameter in Layout object.
MaxVideoLayout	String (32 characters)	NULL	The max video layout. The value is taken from the layout type parameter in Layout object.

---

---

## DeviceControllerIpPort Class

This class defines the IP address and port for the device controller.

**Table 4-5** *DeviceControllerIpPort Class Parameter Table*

Parameter	Data Type	Default Value	Description
IP	String (20 characters)	NULL	The IP address.
Port	Integer	3336	The port.

---

## RoomResource Class

This object is the conference room providing the room for the terminal.

**Table 4-6** *RoomResource Class Parameter Table*

Parameter	Data Type	Default Value	Description
ID	String (32 characters)	NULL	The conference room ID.
Name	String (60 characters)	NULL	The conference room name.
StatusID	Integer	RESOURCE_TYPE_IN_USE	The conference room status. See <a href="#">Table 4-7</a> for a description.
Location	String (128 characters)	NULL	The conference room location.

**Table 4-7** *RoomResource Constant Field Table*

Constant Field	Value	Description
RESOURCE_TYPE_IN_USE	1	Resource type definition for conference room usage.

---

## TerminalResource Class

This object stores the terminal parameters of the terminal resource managed by iVIEW Communications Manager. iVIEW Communications Manager can retrieve all the internal terminals from its database. The external terminals are not stored in the iVIEW Communications Manager database, therefore no external terminal can be retrieved by the iVIEW Communications Manager ResourceService.

**Table 4-8** TerminalResource Class Parameter Table

Parameter	Data Type	Default Value	Description
TerminalID	String (32 characters)	NULL	The terminal ID.
TerminalName	String (60 characters)	NULL	The terminal name.
TerminalNumber	String (40 characters)	NULL	When the terminal is an ISDN terminal then TerminalNumber contains the telephone number. When the terminal is an IP terminal then TerminalNumber contains the IP address.
NodeID	String (32 characters)	NULL	This value can be null if it does not belong to any network. This parameter indicates that it is in the network topology.
CountryCode	String (10 characters)	NULL	When IsISDN contains true then CountryCode contains the country code of the ISDN terminal.
AreaCode	String (10 characters)	NULL	When IsISDN contains true then AreaCode contains the area code of the ISDN terminal.
List<ISDNLocation>	List<ISDNLocation>	NULL	When DetailProtocol contains SERIAL then the ISDN terminal has multiple ISDN locations.

---

**Table 4-8** TerminalResource Class Parameter Table (continued)

Parameter	Data Type	Default Value	Description
ClassificationNames	List<String> (64 characters)	NULL	A terminal can belong to multiple network classifications, and each classification name is not the primary key of the NetClassification object, but it can uniquely identify one record. If the terminal does not belong to any network, this parameter is null.
ClassificationName Count	Integer	0	The count of the classification names.
TerminalEmail	String (120 characters)	NULL	The terminal's e-mail address.
IP	String (20 characters)	NULL	The IP address.
E164	String (50 characters)	NULL	Telephone number or SIP URI.
RegisterGKID	String (32 characters)	NULL	The ID of the gate keeper in which this terminal is registered.
ZonePrefix	String (10 characters)	NULL	The prefix of the network zone which this terminal belongs to.
DefaultRoomID	String (32 characters)	NULL	The ID of the default conference room where this terminal is located.
StatusID	Integer	1	The terminal status ID.
DetailProtocol	Integer	NONE	See <a href="#">Table 4-9</a> for descriptions.
TimezoneID	String (100 characters)	NULL	The time zone ID specified in Java.
IsVoiceOnly	Boolean	False	The flag which identifies it only uses audio port to attend this conference.
MaxBandwidth	Number(9)	0	The maximum bandwidth
SDNMaxBandwidth	Number(9)	0	The ISDN bandwidth if the terminal is dual (H.323 and H.320).

**ResourceService/ResourceV2Service**  
TerminalResource Class

**Table 4-8** *TerminalResource Class Parameter Table (continued)*

<b>Parameter</b>	<b>Data Type</b>	<b>Default Value</b>	<b>Description</b>
TelePresenceInfo	TelePresenceInfo	NULL	Tele-presence detail if the terminal is tele-presence. Note that only ResourceV2Service supports it.

**Table 4-9** *TerminalResource Class Field Table*

<b>Constant Field</b>	<b>Value</b>	<b>Description</b>
NONE	0	No support mobile protocol.
MOBILE_DEFAULT	1	Traditional mobile protocol.
MOBILE_3G	2	3G protocol.
SERIAL	3	Traditional serial protocol.

---

## ISDNLocation Class

This object is defined for storing the ISDN location of the ISDN terminal. Each ISDN location contains the node ID and the telephone number. An ISDN location is also a node in the network topology.

**Table 4-10** *ISDNLocation Class Parameter Table*

Parameter	Data Type	Default Value	Description
NodeID	String (32 characters)	NULL	The node ID of the ISDN location in the network.
CountryCode	String (10 characters)	NULL	The country code.
AreaCode	String (10 characters)	NULL	The area code.
PhoneNumber	String (40 characters)	NULL	The phone number.

---

## TelePresenceInfo Class

This object is defined for the Telepresence terminal. Each Telepresence terminal has at least segments and no more than 4 segments.

**Table 4-11** *TelePresenceInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
TPId	String (32 characters)	NULL	The Telepresence ID of the Telepresence terminal.
TPTemplateId	String (32 characters)	NULL	The Telepresence template ID. Please refer to following table.
List<TPSegment>	List<TPSegment>	NULL	Telepresence segment list

**Table 4-12** *TelePresenceInfo Value Table*

Value	Description
TP_GENERIC	Generic
TP_POLYCOM_TPX_306M	Polycom TPX HD 306
TP_POLYCOM_RPX_200	Polycom RPX 200
TP_POLYCOM_ATX_300	Polycom ATX 300
TP_POLYCOM_RPX_400	Polycom RPX 400
TP_TANDBERG_T3	Tandberg T3
TP_LIFESIZE_CONFERENCE_200	LifeSize Conference

---

## TPSegment Class

This object is defined for one segment of a Telepresence terminal.

**Table 4-13** *TPSegment Class Parameter Table*

Parameter	Data Type	Default Value	Description
SegmentNumber	String (64 characters)	NULL	The segment number

---

## SCHEDULESERVICE AvailableInfo Class

An AvailableInfo object enclosed in AvailableInfoList array object. It stores the information of the checked conference during the specified time span.

**Table 4-14** *AvailableInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
StartTime	Long	0L	The start time of the checked conference.
EndTime	Long	0L	The end time of the checked conference.
Available	Boolean	False	It indicates if the checked conference can be held between StartTime and EndTime.

---

---

## AlternateTerminal Class

This object describes the available status of the given terminal in the specified time slice. If the terminal is available in the specified time frame, the *Available* parameter is true. This indicates this terminal resource may be used in this specified time slice.

**Table 4-15** *AlternateTerminal Class Parameter Table*

Parameter	Data Type	Default Value	Description
TerminalID	String (32 characters)	NULL	The alternative terminal ID.
Available	Boolean	False	It indicates if the checked terminal can be used to attend the conference in the given time slice.

---

---

## AlternateTerminalsTimedSlice Class

This object describes the available statuses for the given terminals in the specified time slices.

**Table 4-16** *AlternateTerminalsTimedSlice Class Parameter Table*

Parameter	Data Type	Default Value	Description
StartTime	Long	0L	The start time of the checked time.
EndTime	Long	0L	The end time of the checked time.
AlternateTerminals	List<AlternateTerminal >	NULL	The availability of the candidate terminal.

---

---

## EventInfo Class

An EventInfo object contains the event scheduled in this conference.

**Table 4-17** *EventInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
EventType	Integer	EVENT_TERMINATION_WARNING	The event type. See <a href="#">Table 4-18</a> for a description.
RelativeTime	Long	0L	The relative time to the end time of the conference.
Memo	String (64 characters)	NULL	The memo information associated with this event.

**Table 4-18** *EventInfo Class Constant Field Table*

Constant Field	Value	Description
EVENT_TERMINATION_WARNING	0	Conference termination warning event type. For example, a termination warning event that this conference will be terminated after five minutes.

---

## LayoutInfo Class

A LayoutInfo object contains the information of the screen layout scheduled in this conference.

**Table 4-19** *LayoutInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
ViewID	String(32 characters)	NULL	The view ID which is the primary key of the View object.
LayoutType	String(32 characters)	NULL	The layout type which is the primary key of the Layout object.
LayoutID	String	NULL	If the conference is currently in session, this is the layout ID assigned by the MCU. Otherwise, it is NULL.
VideoRole	String	NULL	If the conference is currently in session, this is the video role as set by the MCU. Otherwise, it is NULL.
frameTitleShowedState	String	None	If the conference is currently in session, this is the frame title showed state. The returned value will be None, On or Off.

---

---

## ViewInfo Class

Extends LayoutInfo. Holds information of a single view. Views are configured in the MCU.

**Table 4-20** *ViewInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
InitialLayout	String	NULL	The initial video layout.
MaxLayout	String	NULL	The max video layout.
Dynamic	Boolean	False	Whether or not this is a dynamic layout.

---

## RoomInfo Class

The RoomInfo object contains the information of a conference room scheduled in this conference.

**Table 4-21** *RoomInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
RoomID	String (32 characters)	NULL	The conference room ID.
TerminateID	String (32 characters)	NULL	The associated terminate ID.

---

## TerminalInfo Class

A TerminalInfo object stores the related terminal parameters required when scheduling the conference. For an internal terminal, TerminalInfo, TerminalResource and Terminal have the same terminal ID. For an external terminal, only TerminalInfo and Terminal are used because it is not stored in the iVIEW Communications Manager database. TerminalNumber is used instead of TerminalID for TerminalResource. It also has its ID after being invited into the conference scheduled by iVIEW Communications Manager.

**Table 4-22** TerminalInfo Class Parameter Table

Parameter	Data Type	Default Value	Description
TerminalID	String (32 characters)	NULL	The terminal ID.
TerminalNumber	String (40 characters)	NULL	When the terminal is an ISDN terminal then TerminalNumber contains the telephone number. When the terminal is an IP terminal then TerminalNumber contains the IP address.
CountryCode	String (10 characters)	NULL	When IsISDN contains true then CountryCode contains the country code of the ISDN terminal.
AreaCode	String (10 characters)	NULL	When IsISDN contains true then AreaCode contains the area code of the ISDN terminal.
IsOutsider	Boolean	False	The flag which identifies this terminal is outside iVIEW Communications Manager. iVIEW Communications Manager does not store this terminal information.
IsDual	Boolean	False	If this flag contains true then the terminal is an ISDN and an IP terminal.
IsISDN	Boolean	False	If this flag contains true then the terminal is an ISDN terminal. If the flag contains false then the terminal is an IP terminal.

---

**Table 4-22** TerminalInfo Class Parameter Table

Parameter	Data Type	Default Value	Description
IsHost	Boolean	False	Set to true when the terminal is a host.
ScheduleInfo	ScheduleInfo	NULL	After a successful scheduling, this parameter contains the scheduling information and the dialing number.
IsAudioOnly	Boolean	False	<p>The IsAudioOnly and IsTelePresence parameters distinguish the type of terminal.</p> <p>There are three combinations:</p> <ul style="list-style-type: none"> <li>■ AudioOnly=false, TelePresence=false, therefore: Audio Terminal</li> <li>■ AudioOnly=true, TelePresence=false therefore: Audio+Video Terminal</li> <li>■ TelePresence=true therefore: H.D. Terminal</li> </ul>
IsTelePresence	Boolean	False	
AudioVideoPortCount	Integer	1	<p>These three parameters specify how many ports are used for each terminal. One terminal requires only one audio and video port by default. H.D. terminals need multiple ports (it may be a combination of these three kinds of ports).</p>
AudioOnlyPortCount	Integer	0	
VideoOnlyPortCount	Integer	0	
isSip	Boolean	False	<p>The flag which identifies this terminal is an SIP one.</p> <p>@Since V7.0</p>
ProtocolType	Integer	0	<p>0 : supports H.323            1 : supports H.320            2 : supports both H.323 and H.320            3 : supports SIP            4 : supports PSTN</p>

---

## ScheduleInfo Class

Each terminal takes a schedule information parameter which is a null value before scheduling the conference. This parameter will be set after successfully completing the schedule. At present, this class only records the dialing number information for each terminal invited to attend this conference, further, there are two ways to get the dialing number, one is given directly by the DialingExpression parameter in the iVIEW Communications Manager format, and the other is that the client manually composes it with the parameters except for the DialingExpression parameter.

**Table 4-23**      *ScheduleInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
ZonePrefix	String (10 characters)	NULL	The zone prefix.
DialableConferenceID	String (50 characters)	NULL	The dialable conference ID defined in the MCU. Maybe the physical conference ID or the virtual conference ID.
DIDNumber	String (40 characters)	NULL	The DID number to call the SCOPIA Gateway in DID mode not IVR mode.
GWInfo	GatewayInfo	NULL	The associated SCOPIA Gateway info. This parameter is not null only if an ISDN terminal.
DialingExpression	String (255 characters)	False	The expression of dialing number composed by the above parameters according to the specific logic.

---

## GatewayInfo Class

An ISDN terminal needs a SCOPIA Gateway to attend the conference, so this object is used to store the call number of the SCOPIA Gateway.

**Table 4-24** SCOPIA GatewayInfo Class Parameter Table

Parameter	Data Type	Default Value	Description
GatewayPorts	List<GatewayPort>	NULL	The multiple SCOPIA Gateway ports of the SCOPIA Gateway.

---

---

## GatewayPort Class

A SCOPIA Gateway port object stores a complete call number of this SCOPIA Gateway.

**Table 4-25** SCOPIA GatewayPort Class Parameter Table

Parameter	Data Type	Default Value	Description
CountryCode	String (10 characters)	NULL	The country code of the SCOPIA Gateway port.
AreaCode	String(10 characters)	NULL	The area code of the SCOPIA Gateway port.
TelephoneNumber	String (40 characters)	NULL	The phone number of the SCOPIA Gateway port.

---

---

## TerminalLayoutInfo Class

This object takes care of the layout settings for each terminal in the conference.

**Table 4-26**      *TerminalLayoutInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
ViewID	String (32 characters)	NULL	The view ID which is the primary key of the View object
LayoutType	String (32 characters)	NULL	The layout type which is the primary key of the Layout object
PositionID	String (32 characters)	NULL	The grid position of the screen layout in this conference.

---

---

## ConferenceInfo Class

This class consists of parameters that are used for scheduling the conference. The mandatory parameters for scheduling a new conference are:

- Either [MeetingTypeID](#)
- [StartTime](#)
- [EndTime](#)
- TerminalInfos (If not any terminals specified and not any kinds of ports reserved, this conference is actually a blank one)

The only mandatory parameter for modifying, getting or cancelling an existing conference is [ConferenceID](#).

The following table also gives the corresponding user interface item for each conference parameter in the last column if shown on the web page.

**Table 4-27** ConferenceInfo Class Parameter Table

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
ConferenceID	String (32 characters)	NULL	The ID of the conference. This parameter acts as the primary key to the conference information. This can be manually assigned but it must be unique, otherwise the iVIEW Communications Manager will assign it automatically. Avoid manually assigning the ConferenceID when creating a newly scheduled conference to avoid primary key conflicts.	N/A
DialableConferenceID	String (32 characters)	NULL	The virtual conference ID. It is sufficient to specify this parameter to hold an ad-hoc conference.	N/A
Subject	String (80 characters)	NULL	The subject of the scheduled conference.	User > Meeting Scheduling > Meeting Tab > Subject field

**Table 4-27** ConferenceInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
Description	String (255 characters)	NULL	The description of the scheduled conference.	User > Meeting Scheduling > Meeting Tab > Description field
Password	String (32 characters)	NULL	The PIN required for each participant to enter the conference. If the value is NULL, the conference is open to any participant without a password.	User > Meeting Scheduling > Advanced Tab > Password field
StartTime	Long	0L	The conference start time in milliseconds.	User > Meeting Scheduling > Meeting Tab > Start Time field
EndTime	Long	0L	The conference end time in milliseconds.	User > Meeting Scheduling > Meeting Tab > Duration field and its value is equal to EndTime-StartTime.
MeetingTypeID	String (32 characters)	NULL	The meeting type refers to the MCU service of a meeting. A meeting type includes parameters like maximum bandwidth, or whether it supports video or audio-only. All meeting types can be retrieved with the getMeetingTypes API in the iCMResourceService. Each meeting type object includes parameters such as meeting type ID, meeting type name, and service prefix.	User > Meeting Scheduling > Meeting Tab > Meeting Type field gives all the names of candidate Meeting Type.

**Table 4-27** ConferencelInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
Priority	Integer	PRIORITY_ DELAY	The iVIEW Communications Manager schedule engine allocates the required resources for this conference in reference to the Priority value. For example, if Priority is set to Delay the iVIEW Communications Manager will adopt a delay first policy to allocate this conference in the MCU servers. For possible values, see <a href="#">Table 4-28</a> on page 106.	User > Meeting Scheduling > Advanced Tab > Prioritize field
RequiredLevel	Integer	REQUIRED_ ALL_ROOMS _AND_ PARTIES	The iVIEW Communications Manager scheduling engine checks the required resources for this conference based on the defined RequiredLevel. For example, you could specify all participants but not require all rooms, or both, or neither. The iVIEW Communications Manager will only schedule a conference if all required resources are available. For possible values, see <a href="#">Table 4-28</a> on page 106.	User > Meeting Scheduling > Advanced Tab > Required field
List<TerminalInfo>	List<TerminalInfo>	NULL	All the terminals needed to be invited in this conference.	User > Meeting Scheduling > Invite Tab > Selected Participants list box
List<RoomInfo>	List<RoomInfo>	NULL	All the conference rooms required to include each terminal in this conference.	User > Meeting Scheduling > Attendee Settings Tab > Meeting Room column

**Table 4-27** ConferenceInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
List<EventInfo>	List<EventInfo>	NULL	All the events needed to be reserved in this conference.	User > Meeting Scheduling > Advanced Tab > At scheduled time Alert XXX minutes before termination radio button
List<LayoutInfo>	List<LayoutInfo>	NULL	All the screen layouts needed to be set in this conference.	User > Meeting Scheduling > Attendee Settings Tab > Layout select box
ReservedIPPorts	Integer	0	Set number of extra IP endpoints which may join the conference. Reserved IP ports can be used by external or internal endpoints.	User > Meeting Scheduling > Advanced Tab > IP Participants field
ReservedISDNPorts	Integer	0	Set number of extra ISDN endpoints which may join the conference. Reserved ISDN ports can be used by external or internal endpoints.	User > Meeting Scheduling > Advanced Tab > PSTN/ISDN Participants field
ReservedTPAudio OnlyPorts	Integer	0	Reserved audio-only ports for additional audio ports of a Telepresence endpoint. This is only used for the Telepresence service.	N/A
ReservedSPAudio OnlyPorts	Integer	0	Reserved audio-only ports for standard endpoints, because each SCOPIA Desktop endpoint only consumes 1 port. This is only used for Telepresence service.	N/A
BlockDialin	Boolean	False	Internal and external terminals cannot dial into the conference if this parameter is set to false.	User > Meeting Scheduling > Advanced Tab > Block Dial-in check box

**Table 4-27** ConferenceInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
AdmitUnresolved Calls	Boolean	True	External terminals that are unresolved by iVIEW Communications Manager are not permitted to dial into the conference if this parameter is specified to false.	User > Meeting Scheduling > Advanced Tab > Admit Unresolved PSTN/ISDN Calls check box
WaitingRoom	Boolean	True	When this parameter is set to true, all participants are held in a virtual 'waiting room' until the conference host joins the call.	User > Meeting Scheduling > Advanced Tab > Start only when host joins check box
ClassificationName	String (64 characters)	NULL	This classification name indicates in which network this scheduled conference is held.	N/A
IsTestSchedule	Boolean	False	This parameter is a flag to test the conference scheduling without committing the schedule result.	User > Meeting Scheduling > Test button
ForcedDialable ConferenceID	Boolean	false	When this parameter is set to true, the system issues an exception when two virtual conference IDs are in the same or overlapping time slot. When set to false, a new virtual conference ID is assigned automatically if there is a conflict.	N/A
ConferenceStatus	Integer	STATUS_ NOT_ START	The conference status. For possible values, see <a href="#">Table 4-28</a> on page 106. It is set to STATUS_UNKNOWN when scheduling a conference.	N/A
StreamingStatus	Integer	STREAMING _ON	Retrieves the streaming settings of this conference. See <a href="#">Table 4-28</a> for possible values. Relevant for SCOPIA Desktop Clients.	User > Meeting Scheduling > Advanced Tab

**Table 4-27** ConferenceInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
Dialin	Boolean	false	Checks if all the invited terminals' dialing modes are dial-in or dial-out. If they are set to dial-out mode, Dialin is set to false.	N/A
AutoExtend	Boolean	false	It checks if the conference created is extended automatically after its default endtime has passed.	User > Meeting Scheduling > Advanced Tab > Auto Extend radio button
AdvancedInfo	Conference AdvancedInfo	NULL	This parameter stores the advanced values. It can be used to enable cascading.	N/A
SDGInfo	SDGInfo	NULL	If SDGs are configured in the iVIEW Communications Manager, this object stores SDG access information and SDG streaming access information.	Admin > Resource Management > SCOPIA Desktop
FullControlPassword	String	NULL	The control password of the conference.	User > Meeting Scheduling > Advanced Tab
timeZoneID	String	Server local time zone	The local time zone (for example: Asia/Shanghai). If this is not supplied at the time of scheduling, the iVIEW Communications Manager sets it to the default time zone.	
allowRecording	Boolean	False	This parameter determines if the conference created is allowed to be recorded.	<b>User &gt; Profile &gt; Allow or User &gt; Advanced Settings &gt; Default User Settings</b>
recordMeetingWhen Start	Boolean	False	This parameter determines whether to automatically start recording when the conference starts.	N/A

**Table 4-27** ConferenceInfo Class Parameter Table (continued)

Parameter	Data Type (Max length)	Default Value	Description	User Interface Item
allowStreaming	Boolean	False	This parameter determines if the conference to be created is allowed to open streaming.	N/A
recordingQuality	Int	Configurable in SCOPIA Desktop admin GUI.	Recording quality (rate in Kbps) of the conference. Choose one of the following values: 128/192/256/384/512/768/1024.	N/A
recordingDuration	Int	Configurable in SCOPIA Desktop admin GUI.	Recording duration of the conference. The maximum recording duration is 397 minutes.	N/A
orgID	String	NULL	Fake organization ID. It is ignored in scheduled meetings; but it is retrieved as a part of ConferenceInfo.	N/A
userID	String	NULL	Fake user ID. It is ignored in scheduled meetings; but it is retrieved as a part of ConferenceInfo.	N/A
isLPPOn	Boolean	False	It determines whether layout per participant is enabled or not.	N/A @Since V7.0
isVirtualRoom	Boolean	False	It determines whether the conference is a virtual room or not.	N/A @Since V7.1.0
isOneTimePIN Required	Boolean	False	It determines whether the conference requires one time access PIN.	N/A @Since V7.1.0

**Table 4-28**      *ConferenceInfo Class Constant Field Table*

<b>Constant Field</b>	<b>Value</b>	<b>Description</b>
PRIORITY_BANDWIDTH	1	Gives priority to network bandwidth.
PRIORITY_DELAY	2	Gives priority to delay.
PRIORITY_LOCAL	3	Gives priority to local.
REQUIRED_NONE	1	Schedule engine ignores the resources conflict.
REQUIRED_ALL_ROOMS	2	Schedule engine checks conference room conflict.
REQUIRED_ALL_PARTIES	3	Schedule engine checks party conflict.
REQUIRED_ALL_ROOMS_AND_PARTIES	4	Schedule engine checks conference room and party conflict.
REQUIRED_ALL_RESOURCES	5	Schedule engine checks all resources conflict.
TYPE_MCU	0	Indicates MCU device, value is 0.
TYPE_GK	1	Indicates gatekeeper device, value is 1.
TYPE_PARTY	2	Indicates party device, value is 2.
TYPE_GW	3	Indicates SCOPIA Gateway device, value is 3.
TYPE_NETWORK	4	Indicates network path, value is 4.
TYPE_ROOM	5	Indicates conference room, value is 5.
TYPE_STR_GW	6	Indicates streaming SCOPIA Gateway, value is 6.
TYPE_ACCESSORY	7	Indicates Accessory, value is 7.
STATUS_UNKOWN	-1	Indicates that the status is unknown.
STATUS_NOT_START	0	Indicates that the conference has not yet started.

**Table 4-28** *ConferencelInfo Class Constant Field Table (continued)*

<b>Constant Field</b>	<b>Value</b>	<b>Description</b>
STATUS_IN_SESSION	1	Indicates that the conference is in progress.
STATUS_FINISHED	2	Indicates that the conference has finished.
STATUS_START_FAILED	3	Indicates that the conference has failed to start.
STATUS_CANCELLED	4	Indicates that the conference has been cancelled.
STATUS_SCHEDULE_FAILED	5	Indicates that conference scheduling has failed.
STATUS_BACKUP	6	Indicates that the conference plan is a backup plan.
STATUS_STOP_FAILED	7	Indicates that the conference failed to stop. This is an interim status.
STATUS_ABNORMAL_STOPPED	8	Indicates that the conference stopped abnormally. This is an interim status.
STREAMING_OFF	0	Streaming is disabled.
STREAMING_ON	1	Streaming is enabled.

---

## VirtualRoomInfo Class

Extends ConferenceInfo. Holds information of a virtual room instance.

**Table 4-29** *VirtualRoomInfo Class Parameter Table*

Parameter	Data Type (Max length)	Default Value	Description	UI Item
Name	String (32 characters)	NULL	The name of the virtual room.	User > My Profile > Virtual Room Setting > Virtual Room Name field
Public	Boolean	false	When set to true, users who do not own the room can also schedule meetings for this room.	N/A
GatewayInfo	GatewayInfo	NULL	Lists dial in information for SCOPIA Gateways configured into the system. See <a href="#">GatewayInfo Class</a> for the object description.	Admin > Resource Management > Gateway

---

---

## SDGInfo Class

This object holds the SDG access information and streaming access information of the conference.

**Table 4-30**     *SDGInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
SDGAccessInfos	List<SDGAccessInfo>	NULL	The access information for every SDG configured in the iVIEW Communications Manager.

---

---

## SDGAccessInfo Class

This object holds the SDG access information of the conference.

Currently it only contains the access URL, there might be more information in the future.

**Table 4-31** *SDGAccessInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
SDGAccessURL	String	NULL	The access URL for SDG clients for the conference.

---

---

## ConferenceAdvancedInfo Class

This object lists those parameters that are not frequently used or not exposed to the client but to the server.

**Table 4-32** *ConferenceAdvancedInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
AllowDynamicGrow	Boolean	True	If true, it indicates that the iVIEW Communications Manager allows the meeting to grow in size after it has started.
DurationAfterLeft	Integer	10 mins	It indicates after how many minutes the conference is terminated after all parties or the host has left. Note that it takes effect when TerminalCondition is set to TERMINATION_AFTER_ALL_PARTIES_LEFT or TERMINATION_AFTER_HOST_LEFT .
TerminationCondition	Integer	NORMAL_TERMINATION	This parameter specifies the termination condition of the conference. See <a href="#">Table 4-33</a> for a description.
EnableMCU Cascading	Boolean	True	If true, the iVIEW Communications Manager tries to allocate resources in more than one level MCU when scheduling the conference.

**Table 4-33** *ConferenceAdvancedInfo Class Constant Field Table*

Constant Field	Value	Description
NORMAL_TERMINATION	0	Terminates the conference at a scheduled time.

**Table 4-33** *ConferenceAdvancedInfo Class Constant Field Table*

Constant Field	Value	Description
TERMINATION_AFTER_ALL_PARTIES_LEFT	1	Terminates the conference the specified time in DurationAfterLeft after all parties leave.
TERMINATION_AFTER_HOST_LEFT	2	Terminates the conference the specified time in DurationAfterLeft after the host leaves.

---

## RecurrenceInfo Class

The recurrence information consists of parameters and is used for scheduling the recurrent conference. The mandatory parameters for the scheduling of a new recurrent conference by the client are:

- [ConferenceInfoTemplate](#)
- [List<RecurInstanceInfo>](#)

The mandatory parameters for modifying an old recurrent conference with the client are:

- [ConferenceInfoTemplate](#)
- [List<RecurInstanceInfo>](#)
- [RecurrenceID](#)

Note that the RecurrenceID and List<ConferenceInfo> parameters are filled in and this new recurrence information is returned into the ScheduleResultList object after executing to create the recurrent conference. The List<ConferenceInfo> parameter is filled in and this recurrence information is returned to the List<ScheduleResult> object after executing to modify the old recurrent conference.

**Table 4-34**      *RecurrenceInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
ConferenceInfoTemplate	ConferenceInfo	NULL	The conference information object as a template in which the start/end time is included.
List<RecurInstanceInfo>	List<RecurInstanceInfo>	NULL	The array object of recurring instance information and each element is a RecurInstanceInfo object which includes the start/end time and the instance ID.
RecurrenceID	String (32 characters)	NULL	The recurrent conference ID.
List<ConferenceInfo>	List<ConferenceInfo>	NULL	The recurrence instances.

---

## RecurInstanceInfo Class

A RecurInstanceInfo object is used to express the basic information of the recurrence instance.

**Table 4-35** *RecurInstanceInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
InstanceID	String (32 characters)	NULL	The instance conference ID.
StartTime	Long	0L	The start time of the recurring instance.
EndTime	Long	0L	The end time of the recurring instance.

---

## ScheduleResult Class

A ScheduleResult object contains the successful or failed result information after executing an iVIEW Communications Manager schedule operation.

**Table 4-36** ScheduleResult Class Parameter Table

Parameter	Data Type	Default Value	Description
IsSuccess	Boolean	False	The flag which indicates this schedule operation executed successfully.
ErrorStatus	Integer	ERROR_NOT_SUPPORT	The error status or code if the schedule operation fails to execute. See <a href="#">Table 4-37</a> for error descriptions.
ErrorInfo	String (255 characters)	NULL	The detailed error information in a message string format.
ResourceType	Integer	-1	The resource type of the conflict resource and its value is defined by the iVIEW Communications Manager. For example, a MCU has a lot of resource types, such as connections (its value is 2), conference numbers (its value is 3).
ResourceID	String (32 characters)	NULL	The conflict resource ID, an internal ID of the iVIEW Communications Manager, is used to identify a resource managed by the iVIEW Communications Manager.
ResourceDeviceType	Integer	-1	The device type of the conflict resource managed by the iVIEW Communications Manager, and it may be a terminal, a MCU, a Gatekeeper, a SCOPIA Gateway and so on. This constant definition can be seen in the ConferenceInfo class.
ConferenceID	String (32 characters)	NULL	The ID of the conference being operated.

**Table 4-36**     *ScheduleResult Class Parameter Table (continued)*

<b>Parameter</b>	<b>Data Type</b>	<b>Default Value</b>	<b>Description</b>
ConferenceInfo	ConferenceInfo	NULL	The returned ConferenceInfo object after executing this operation.
PublicAccessUrl	String (adjustable length)	NULL	The in-meeting control access URL for the master iVIEW Communications Manager.
LocalAccessUrl	String (adjustable length)	NULL	The in-meeting control access URL for the device control.

**Table 4-37**     *ScheduleResult Class Constant Field Table*

<b>Constant Field</b>	<b>Value</b>	<b>Description</b>
ERROR_MCU_RESOURCE_SHORTAGE	1	MCU resource shortage, the resource type is vendor dependent.
ERROR_GK_RESOURCE_SHORTAGE	2	Gatekeeper resource shortage, including registration and connection.
ERROR_GW_RESOURCE_SHORTAGE	3	SCOPIA Gateway resource shortage.
ERROR_PARTY_NOT_AVAILABLE	4	A terminal or some terminal is not available or conflict.
ERROR_NETWORK_CONGESTION	5	Network bandwidth shortage.
ERROR_CONFERENCE_ROOM_CONFLICT	6	Conference room conflict.
ERROR_NOT_SUPPORT	7	Unknown exception.
ERROR_NETWORK_NOT_ARRIVE	8	Network not arrive exception.
ERROR_STORAGE	9	Storage exception, maybe database exception.
ERROR_MCU_SERVICE_NOT_SUPPORT	10	MCU service not support exception.

**ScheduleService**  
ScheduleResult Class

**Table 4-37** ScheduleResult Class Constant Field Table (continued)

Constant Field	Value	Description
ERROR_GW_SERVICE_NOT_SUPPORT	11	SCOPIA Gateway service not support exception.
ERROR_SERVICE_NOT_ACTIVE	12	Service template not active exception.
ERROR_GW_PARTY_RATE_NOT_MATCH_TCH	13	Terminal connect rate can not match with SCOPIA Gateway service definitions.
ERROR_NO_RESERVED_PORT_RESOURCE	14	No reserved port resource for this conference in port limited mode.
ERROR_INVALID_PARTIES	15	The input party data is invalid.
ERROR_NO_GK_SUPPORTS_P2P	16	No gatekeeper currently supports point to point conference.
ERROR_SCHEDULE_FAILED	17	Schedule failed with an unknown reason.
ERROR_NO_STREAMING_RESOURCE	18	Schedule failed with no available streaming resource.
ERROR_PERSONAL_CONFERENCE_ID_CONFLICTED	19	Schedule failed because personal conference ID conflicted.
ERROR_VIRTUAL_CONFERENCE_ID_CONFLICTED	20	Schedule failed because virtual conference ID conflicted.
ERROR_LICENSE_SHORTAGE	21	Schedule failed because license is shortage.
ERROR_MEETING_CONTROL	99	If error status or code is equal to 99, it indicates that the schedule error derived from the meeting control; please refer to the error code definitions in <a href="#">ControlResult Class</a> on page 124.
ERROR_SVR_NULL_RECURRENCE_INFO_ARGUMENT	100	The RecurrenceInfo object passed in the argument is null.
ERROR_SVR_NULL_RECURRENCE_ID_ARGUMENT	101	The recurrence ID passed directly or indirectly in the argument is null.

**Table 4-37**     *ScheduleResult Class Constant Field Table (continued)*

Constant Field	Value	Description
ERROR_SVR_NULL_CONFERENCE_IN FO_TEMPLATE_ARGUMENT	102	The ConferenceInfo template passed in the argument is null when scheduling or rescheduling the recurring conference.
ERROR_SVR_NULL_RECURRENCE_ INSTANCE_INFOS_ARGUMENT	103	The RecurInstanceInfo list object passed directly or indirectly in argument is null when scheduling or rescheduling the recurring conference.
ERROR_SVR_SET_PUBLIC_LOCAL_ ACCESS_URL	104	The error is caught for setting the public or local access URL.
ERROR_SVR_GET_NEXT_ RECURRENCE_ID	105	The error is caught on getting the next recurrence ID.
ERROR_SVR_NULL_CONFERENCE_ID _ARGUMENT	106	The conference ID passed directly or indirectly in the argument is null.
ERROR_SVR_NULL_MEETING_TYPE_ ID_ARGUMENT	107	The meeting type ID passed directly or indirectly in the argument is null.
ERROR_SVR_NULL_CONFERENCE_IN FO_ARGUMENT	108	The ConferenceInfo object passed directly or indirectly in the argument is null.
ERROR_SVR_NULL_CONFERENCE_ID S_ARGUMENT	109	The conference ID list object passed directly or indirectly in the argument is null.
ERROR_SVR_NONE_INSTANCE_FOR_ RECURRENCE	110	No instance associated with the specified recurrence is found.
ERROR_SVR_INVALID_ CLASSIFICATION_NAME	111	The specified classification name is invalid or does not exist in the iVIEW Communications Manager.
ERROR_SVR_NOT_FOUND_ CONFERENCE	112	The specified conference with the given ID is not found.
ERROR_SVR_CANCEL_ALREADY_ CANCELLED_CONFERENCE	113	Failed to cancel a conference which is already cancelled.

**Table 4-37** ScheduleResult Class Constant Field Table (continued)

Constant Field	Value	Description
ERROR_SVR_CANCEL_ALREADY_FINISHED_CONFERENCE	114	Failed to cancel a conference which is already finished.
ERROR_SVR_CANCEL_BACKUP_CONFERENCE	115	Failed to cancel the backup conference.
ERROR_SVR_CANCEL_SCHEDULE_FAILED_CONFERENCE	116	Failed to cancel a scheduled conference.
ERROR_MODIFY_NON_NOT_STARTED_CONFERENCE	117	Failed to modify a conference which is not upcoming.
ERROR_NULL_ORG_OR_USER_ID	200	Organization ID cannot be null and 999, User ID cannot be null.
ERROR_OID_NOT_FOUND_IN_ICM	201	Organization ID: XXX cannot be found in SCOPIA iVIEW Management Suite.
ERROR_USERID_NOT_FOUND_IN_ICM	202	User ID: XXX cannot be found in SCOPIA iVIEW Management Suite's Organization: XXX.
ERROR_USER_CANNOT_SCHEDULE_CONFERENCE	203	User: XXX cannot schedule a conference in SCOPIA iVIEW Management Suite.
ERROR_STREAM_CONFLICTION_NOT_ALLOWED	204	Streaming: not allowed, but enabled (default: allowed, not enabled).
ERROR_RECORD_CONFLICTION_NOT_ALLOWED	205	Recording: not allowed, but enabled (default: not allowed, not enabled).
ERROR_RECORD_QUALITY	206	Recording quality (rate in Kbps) should be according to enumeration: 128/192/256/384/512/768/1024.
ERROR_RECORD_DURATION	207	Max recording duration is 397 minutes.

---

## ConferenceInfoCondition Class

An instance of ConferenceInfoCondition defines the search condition for searchConferences.

**Table 4-38** *ConferenceInfoCondition Class Parameter Table*

Parameter	Data Type	Default Value	Description
subject	String	NULL	The subjects of conferences.
conferenceID	String	NULL	The conference ID.
conferenceStatus	Integer	0	Values: <ul style="list-style-type: none"> <li>■ 0—Not started (upcoming)</li> <li>■ 1—In session</li> <li>■ 2—Finished (history)</li> </ul>
startTime	long	0	Measured in milliseconds, between the current time and midnight, January 1, 1970 coordinated universal time (UTC).
endTime	long	0	Measured in milliseconds, between the current time and midnight, January 1, 1970 coordinated universal time (UTC).
partyE164	String	null	The terminal number.

---

## **CONTROLSERVICE** Terminal Class

A Terminal object contains the terminal information when controlling the ongoing conference. It uses a terminal ID for an internal terminal, but not for an external one. A terminal ID is enough to identify an internal terminal. To identify an external terminal, you need the telephone number or IP and the related parameters.

**Table 4-39** Terminal Class Parameter Table

<b>Parameter</b>	<b>Data Type</b>	<b>Default Value</b>	<b>Description</b>
TerminalID	String (32 characters)	NULL	The terminal ID.
TerminalName	String (60 characters)	NULL	The terminal name.
TerminalNumber	String (40 characters)	NULL	The terminal number or telephone number.
IP	String (20 characters)	NULL	The IP address for the IP terminal.
countryCode	String (32 characters)	NULL	Country code.
areaCode	String (32 characters)	NULL	Area code.
terminalProtocol	Integer	0	Terminal protocol: <ul style="list-style-type: none"><li>■ 0—H323</li><li>■ 1—H320</li><li>■ 3—SIP</li></ul>
PhysicsID	String (32 characters)	NULL	The physics ID.
Description	String (255 characters)	NULL	The terminal description.
ConnectRate	Integer	0	The rate in bps to connect this terminal.
IsDialin	Boolean	False	The flag which identifies it is allowed to dial in.

**Table 4-39** Terminal Class Parameter Table (continued)

Parameter	Data Type	Default Value	Description
IsHost	Boolean	False	The flag which identifies it as a host. It denotes that this terminal has control power for the current meeting. If this terminal is not in this conference, it has a pending status even if others are in this conference.
IsOutside	Boolean	False	The flag which identifies it is an external terminal not managed by the iVIEW Communications Manager, namely not having a internal terminal ID.
IsAudioOnly	Boolean	False	These two boolean parameters distinguish the terminal type.
	AudioOnly=false, TelePresence=false:		Audio Terminal
	AudioOnly=true, TelePresence=false:		Audio+Video Terminal
	TelePresence=true:		HD Terminal
IsTelePresence	Boolean	False	
AudioVideoPortCount	Integer	1	These three parameters specify how many ports are occupied for each terminal, and one terminal requires only one audio and video port by default. H.D terminals need multiple ports (it may be the combination of these three kinds of ports).
AudioOnlyPortCount	Integer	0	
VideoOnlyPortCount	Integer	0	

---

## ControlResult Class

A ControlResult object contains the successful or failed result information after executing an iVIEW Communications Manager control operation.

**Table 4-40** ControlResult Class Parameter Table

Parameter	Data Type	Default Value	Description
IsSuccess	Boolean	False	The flag which identifies if this control operation is executed successfully.
OperationFailedCode	Integer	UNKNOWN	When the control operation fails to execute, this parameter contains the failed code. See <a href="#">ControlResult Class Constant Field Table</a> on page 125.
ErrorCode	Integer	ERROR_UNSPECIFIED	The error code. See <a href="#">ControlResult Class Constant Field Table</a> on page 125. The error code takes effect only if IsSuccess is false; the client should ignore this value (error code) if IsSuccess is true.
ErrorInfo	String (255 characters)	NULL	When ErrorCode is ERROR_UNSPECIFIED, namely for those errors that cannot be covered by the ErrorCode constant definitions, it stores this error information.
DeviceID	Integer	NULL	The ID of the device in which an error is captured is an internal ID in the iVIEW Communications Manager for defining a physical device.
DeviceName	String (60 characters)	NULL	The name of the device in which an error is captured.
DeviceIp	String (20 characters)	NULL	The IP address of the device in which an error is captured.

**Table 4-40** ControlResult Class Parameter Table (continued)

Parameter	Data Type	Default Value	Description
ConferenceID	String (32 characters)	NULL	The ID of the conference controlled currently by the iVIEW Communications Manager.
TerminalID	String (32 characters)	NULL	When inviting a new and external terminal, the iVIEW Communications Manager automatically assigns an ID for this terminal and returns it to the client.
			<b>Note</b> An external terminal does not have a terminal ID.

**Table 4-41** ControlResult Class Constant Field Table

Constant Field	Value	Description
ERROR_CONFERENCE_ALREADY_STARTED	101	Conference module exception error code definition, value is 101. The conference is already in meeting.
ERROR_CONFERENCE_NOT_IN_MEETING	102	The conference not in meeting.
ERROR_DEVICE_OPERATION_FAILED	103	Device operation failed.
ERROR_RESOURCE_ALLOCATION_FAILED	104	Resource allocation failed.
ERROR_MCU_CONNECTION_UNAVAILABLE	105	MCU connection not available.
ERROR_PARTY_NOT_READY	106	Party status not ready for change.
ERROR_CONFERENCE_NOT_IN_QUEUE	107	Conference Not in Queue.

**ControlService**  
ControlResult Class

**Table 4-41** ControlResult Class Constant Field Table (continued)

<b>Constant Field</b>	<b>Value</b>	<b>Description</b>
ERROR_GK_CONNECTION_UNAVAILABLE	108	GateKeeper connection not available.
ERROR_PARTY_ALREADY_EXISTED	109	Party Already in meeting.
ERROR_PARTY_NOT_FOUND	110	Party does not exist.
ERROR_UNSPECIFIED	-1	Unspecified error status.
UNKNOWN	-1	An unknown code.
NO_RESOURCES	1	No resource.
NO_PERMISSION	2	No permission.
CONF_NOT_FOUND	3	Conference not found.
PARTY_NOT_FOUND	4	Party not found.
CHANNEL_NOT_FOUND	5	Channel not found.
NO_RESPONSE_TIME_OUT	6	Request sent but no response received before the time out.

**USERSERVICE**

**UserInfo Class**

**Table 4-42** *UserInfo Class Parameter Table*

Parameter	Data Type	Default Value	Description
userID	String	null	The identifier of a user entity.
loginID	String	required	The login ID of the user.
password	String	null	The password.
roleID	int	3	Values: <ul style="list-style-type: none"> <li>■ 1—Admin</li> <li>■ 2—Conference Organizer</li> <li>■ 3—Regular user</li> </ul>
userLastName	String	required	The last name of the user.
userFirstName	String	null	The first name of the user.
branch	String	null	
department	String	null	
officePhone Number	String	null	
mobilePhone Number	String	null	
email	String	required	
defaultTerminalID	String	null	The ID of the user’s default terminal. Use when scheduling conferences by users.

---

## UserSearchConditon Class

**Table 4-43** *UserSearchCondition Class Parameter Table*

Parameter	Data Type	Default Value	Description
firstName	String	null	The pattern that the first name should match.
lastName	String	null	The pattern that the last name should match.
loginID	String	null	The pattern that the login ID should match.
name	String	null	(Deprecated)

The above search condition is an *and* operation.

---

## DIRECTORYUSER SERVICE

---

## DirectoryOperationSession Class

**Table 4-44** *DirectoryOperationSession Class Table*

Parameter	Data Type	Default Value	Description
organization	String(128)	NULL	The organization name that used when login to multiple-tenant iVIEW Pro edition. For Non-Pro edition, do not need to set it.
cookie	String	NULL	Session cookie ID
host	String(64)	NULL	The host of the directory server. it should be the same as what is input in the LDAP configuration page

**Table 4-44** *DirectoryOperationSession Class Table (continued)*

Parameter	Data Type	Default Value	Description
organizationId	String(128)	NULL	The internal ID of the organization. Only one of organization or organizationId should be set. If they all are set, organizationId is used.

---

## DirectoryUserInfo Class

DirectoryUserInfo is a subclass of UserInfo (see the UserInfo definition in The Programmer Guide for iCM Web Services).

**Table 4-45**     *DirectoryUserInfo Class*

Parameter	Data Type	Default Value	Description
distinguishedName	String(128)	NULL	The distinguished name of the user
virtualRoomNumber	String(32)	NULL	The virtual room number. If it is set, the default virtual room will be created with this number for the user. The number will be prefixed with the default prefix defined in iVIEW.
alternativeLoginIds	A list of String(120)	NULL	A user might use more than one login ids to log in a directory server, for example, the domino server allows the user to use the common name or the abbreviated name. One of them is set in the loginId attribute of the userInfo, and the others are set here. Using this attribute, we can define how the user can log in a directory server.
loginIdsLdapAttrNames	A list of String(32)	NULL	A list of the attribute name on LDAP server for login ID. There must be a one-to-one Correspondence between the login ID in alternativeLoginIds and the name in loginIdsLdapAttrNames.

---

# 5

## LOCATION, DEPENDENCIES AND PROPERTIES

---

- [iCM Service Location](#) on page 131
- [Dependencies](#) on page 131
- [Configurable Properties](#) on page 132

### ICM SERVICE LOCATION

Since SCOPIA iVIEW Management Suite version 5.3 the location of icmservice.war is located at `deploy\wcs.ear\icmservice.war`.

### DEPENDENCIES

Since the addition of virtual room APIs, the web services are dependant on the following oss jars:

- `template.jar`
- `member-ejb.jar`
- `vnexclient.jar` (for logging)

### CONFIGURABLE PROPERTIES

The following lists all core properties associated with the web services component:

- `com.radvision.icm.service.init.checkClientAddress`—Default is true. If true it requires that the web services client and its server are located in the same computer, otherwise a SOAP exception is thrown
- `com.radvision.icm.service.init.logWebServicesTimes`—Default is false. If true, time related information is logged in WARN level for the newly added APIs in order to allow better analysis of the web service response time. The newly added APIs are:
  - `getDialedNumberConferenceInfo`
  - `getVersion`
  - `searchVirtualRooms`
  - `viewConference`
  - `viewConferenceByDialableNumber`
  - `viewVirtualRoomByDialableNumber`
  - `viewVirtualRoomById`
  - `viewVirtualRoomByName`
  - `viewVirtualRoomByStrippedDialableNumber`
  - `viewVirtualRooms`
  - `viewVirtualRoomsByUserId`

Suggestion: Add this kind of logging in other API methods, where relevant.

---

**Note** These core properties are placed in the `vcs-core.properties` file in the following directory `...\jboss\bin`

---

### ASSOCIATED WSDL FILES

The following services are defined in the accompanying WSDL (Web Service Definition Language) files:

- `ResourceService`
- `ResourceV2Service`
- `ScheduleService`
- `ControlService`
- `UserService`
- `DirectoryUserService`

# APPENDIX A

## ERROR CODES

---

**Table A-1**      *Error Codes*

<b>Code</b>	<b>Description</b>
1	MCU resource shortage
2	Gatekeeper resource shortage
3	SCOPIA Gateway resource shortage
4	A party is not available or a conflict exists
5	Network bandwidth shortage
6	Conference room conflict
7	Unknown exception
8	Network connection not available
9	Storage or database access exception
10	The meeting service is not supported
11	The SCOPIA Gateway service is not supported
12	The service template is not active

**Table A-1**      *Error Codes (continued)*

<b>Code</b>	<b>Description</b>
13	The party connect rate cannot match the SCOPIA Gateway service definitions
14	No reserved ports for this conference in port limited mode
15	The party data is invalid
16	No gatekeeper in the system supports point to point conferencing
17	Schedule failed—Reason unknown
18	Schedule failed—Streaming resource unavailable
19	Schedule failed—Personal conference ID conflict
20	Schedule failed—Virtual conference ID conflict
33	No available Gateway for ISDN EPs.