

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Related Pages

---

Here is a list of all related documentation pages:

- [Error Codes](#)
- [Version History](#)
- [Using Skype4COM with Internet Explorer](#)
- [Deprecated List](#)

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Error Codes

---

- 0x80040200 E\_SKYPE\_NOT\_CONNECTED
- 0x80040201 E\_SKYPE\_WAIT\_TIMEOUT
- 0x80040202 E\_SKYPE\_WAIT\_FAILED
- 0x80040203 E\_SKYPE\_PROTOCOL\_ERROR
- 0x80040204 E\_SKYPE\_CONNECTION\_REFUSED
- 0x80040205 E\_SKYPE\_PROTOCOL\_VERSION
- 0x80040206 E\_SKYPE\_RUN\_FAILED
- 0x80040207 E\_SKYPE\_NOT\_INSTALLED

---

Copyright  2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Version History

---

## 1.0.36

- fixed: **IClient::IsRunning** property (supports Skype 5.1 and later builds).

## 1.0.35

- added UIPI message filter (Windows 7, Vista).

## 1.0.34

- fixed bug in collection enumerators.

## 1.0.33

- fixed memory leak in **Skype** class constructor.

## 1.0.32

- fixed: **IClient::Focus** and open dialog commands did not focus Skype 4.0 window

## 1.0.30

- fixed: **IClient::IsRunning** property.

## 1.0.29

- added: **\_ISkypeEvents::SilentModeStatusChanged** event.
- added: **\_ISkypeEvents::UserAuthorizationRequestReceived** event.
- added: **ISkype::PredictiveDialerCountry** property.
- bugfix: **IChatMember::CanSetRoleTo** always returned VARIANT\_FALSE:

## 1.0.28

- added: **ICall::TargetIdentity** property.
- added: **IChat::AcceptAdd** method.
- added: **IChat::ActivityTimestamp** property.
- added: **IChat::AlertString** property.
- added: **IChat::Applicants** property.

- added: **IChat::Blob** property.
- added: **IChat::ClearRecentMessages** method.
- added: **IChat::Description** property.
- added: **IChat::DialogPartner** property.
- added: **IChat::Disband** method.
- added: **IChat::EnterPassword** method.
- added: **IChat::GuideLines** property.
- added: **IChat::Join** method.
- added: **IChat::Kick** method.
- added: **IChat::KickBan** method.
- added: **IChat::MemberObjects** property.
- added: **IChat::MyRole** property.
- added: **IChat::MyStatus** property.
- added: **IChat::Options** property.
- added: **IChat::PasswordHint** property.
- added: **IChat::SetPassword** method.
- added: **IChat::TopicXML** property.
- added: **IChat::TopicXML** property.
- added: **IChat::Type** property.
- added: **IClient::OpenLiveTab** method.
- added: **ISettings::LoadAvatarFromFile** method.
- added: **ISettings::ResetIdleTimer** method.
- added: **ISettings::SaveAvatarToFile** method.
- added: **ISkype::CreateChatUsingBlob** method.
- added: **ISkype::FindChatUsingBlob** method.
- added: **ISkype::FocusedContacts** property.
- added: **IUser::IsVoicemailCapable** property.
- added: **TChatMemberRole** type.
- added: **TChatMyStatus** type.
- added: **TChatOptions** type.
- added: **TChatType** type.
- added: **\_ISkypeEvents::CallTransferStatusChanged** event.
- added: **\_ISkypeEvents::ChatMembersChanged** event.
- added: **\_ISkypeEvents::ChatMemberRoleChanged** event.
- added: interfaces **IChatMember** and **IChatMemberCollection**.
- change: **ISettings::RingToneStatus** property settable.
- change: calls to event handler are now serialised.

- added: **\_ISkypeEvents::CallVideoStatusChanged** event.
- added: **\_ISkypeEvents::CallVideoSendStatusChanged** event.
- added: **\_ISkypeEvents::CallVideoReceiveStatusChanged** event.
- bugfix: #59913 Microphone stream redirection to port and file

### 1.0.27

- added: **ICall::CanTransfer** property.
- added: **ICall::TransferStatus** property.
- added: **ICall::TransferActive** property.
- added: **ICall::TransferredBy** property.
- added: **ICall::TransferredTo** property.
- added: **ICall::Transfer** method.
- added: **ISkype::ActiveFileTransfers** property.
- added: **ISkype::FileTransfers** property.
- added: **\_ISkypeEvents::FileTransferStatusChanged** event.
- added: **IUser::RichMoodText** property.
- added: **IProfile::RichMoodText** property.
- fixed: **\_ISkypeEvents::GroupUsers** event did not fire.

### 1.0.26

- fixed: bugs in translation file.
- fixed: IDL incompatibility with version 1.0.0.22

### 1.0.25

- fixed: self-registration for LUA user.
- fixed: **ICall::InputDevice** and **ICall::OutputDevice** caching problem.
- fixed: **ISkype::Protocol** wrong value.
- added: **ISkype::ApiWrapperVersion** property.
- added: **ISkype::SilentMode** property.
- added: **IClient::Wallpaper** property.
- added: **\_ISkypeEvents::WallpaperChanged** event.

### 1.0.24.0

- added **PluginEvent** class.
- added **PluginMenuItem** class.



- added **\_ISkypeEvents::PluginEventClicked** event.
- added **\_ISkypeEvents::PluginMenuItemClicked** event.

### 1.0.23.0

- changed file versioning format.
- added **TChatMessageType::cmeEmoted**.
- added **\_ISkypeEvents::CallSeenStatusChanged** event.
- changed **ICall::Seen** property readable.
- added **ICall::ForwardedBy** property.
- changed **IUser::DisplayName** property is now writable.
- fixed: bug in **IProfile::CallSendToVM** implementation.

### 1.0.0.22

- added **ISkype::AsyncSearchUsers** method.
- added **ICall::InputDevice** property.
- added **ICall::InputStatus** property.
- added **ICall::OutputDevice** property.
- added **ICall::CaptureMicDevice** property.
- added **IUser::NumberOfAuthBuddies** property.
- added **\_ISkypeEvents::CallInputStatusChanged** event.
- added **\_ISkypeEvents::AsyncSearchUsersFinished** event.

### 1.0.0.21

- added **IClient::OpenSmsDialog** method.
- added **IProfile::ValidatedSmsNumbers** property.
- added **ISkype::CreateSms** property.
- added **ISkype::MissedSmss** property.
- added **ISkype::SendSms** method.
- added **ISkype::Smss** property.
- added **SmsChunk** class.
- added **SmsChunkCollection** class.
- added **SmsMessage** class.
- added **SmsMessageCollection** class.
- added **SmsTarget** class.
- added **SmsTargetCollection** class.
- fixed: bug in **IChatMessage::Seen** implementation.

### 1.0.0.14

- added **IProfile::IPCountry** property.

- fixed: bug in command cache implementation that caused strings to be truncated.

#### 1.0.0.12

- added **ISkype::ApiSecurityContextEnabled** property.
- added **ISkype::EnableApiSecurityContext** method.
- added **ISkype::MissedVoicemails** property.
- added **ICall::Rate** property.
- added **ICall::RateCurrency** property.
- added **ICall::RatePrecision** property.
- added **IChat::Bookmarked** property.
- added **IChat::Bookmark** method.
- added **IChat::Unbookmark** method.
- added **IChatMessage::Sender** property.
- added **IGroup::Accept** method.
- added **IGroup::Decline** method.
- added **IGroup::Share** method.
- changed: **ISettings::Language** writable.
- added **IVoicemail::SetUnplayed** property.

#### 1.0.0.11

- added **CallChannel** class.
- added **CallChannelCollection** class.
- added **CallChannelManager** class.
- added **CallChannelMessage** class.

#### 1.0.0.10

- added **\_ISkypeEvents::UserMood** event.

#### 1.0.0.9

- added HTML documentation

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Using Skype4COM with Internet Explorer

---

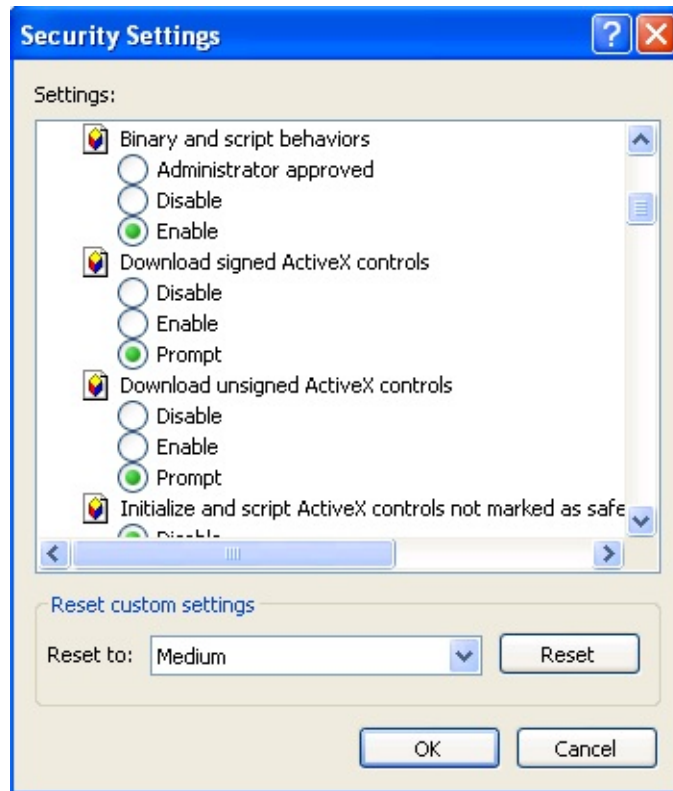
## Adding Skype4COM to a web page

To add Skype4COM controls to a web page, use the *OBJECT* tag in the *HTML* code.

```
<html>
<body>
<object id=Skype
        name=Skype
        height=0
        width=0
        classid=clsid:830690FC-BF2F-47A6-AC2D
-330BCB402664
        codebase="http://www.yoursite.com/Sky
pe4COM.dll" >
    <span style="color: red">Failed to load con
trol.</span>
</object>
</body>
</html>
```

## Installing Skype4COM with Internet Explorer

Before you begin, ensure that Internet Explorer security policies are set to download signed ActiveX controls. Select Tools menu > Internet Options > Security tab and click the Custom Level button to display the ActiveX settings.



If downloading ActiveX controls is set to Prompt, when a user first visits a web page containing a Skype4COM object, Internet Explorer displays a warning.



The user must click the Install button to start the download. After the installation, the Internet Explorer page reloads, and Skype4COM is ready for use.

## Skype4COM security policies for Internet Explorer

Internet Explorer defines five security zones for web content:

- Local Machine (this zone is not visible in the Internet Explorer security tab)
- Local Intranet
- Trusted
- Internet
- Restricted

Skype4COM defines four API security contexts:

- Voice services
- Messaging services
- Account services
- Contact services

During installation, Skype4COM creates custom policies for each API context in each Internet Explorer security zone.

<b>Security Zone</b>	<b>Voice API</b>	<b>Messaging API</b>	<b>Account API</b>	<b>Contact API</b>
Local Machine	Allow	Allow	Allow	Allow
Local Intranet	Allow	Allow	Prompt	Allow
Trusted	Allow	Allow	Prompt	Allow
Internet	Prompt	Prompt	Prompt	Prompt
Restricted	Disallow	Disallow	Disallow	Disallow

If the zone setting for a security context is set to "Prompt", Skype4COM displays a window where the user must select whether to allow this context or not. For example, if a web page

contains a Call Me button, and voice services for the Internet security zone are set to prompt, the following window appears:



The user receives multiple prompts if a web page contains more than one Skype API security context, for example a page with a Call Me button and a Chat to Me button, where the security policies for voice and messaging services are both set to Prompt.

If the user selects the Allow (first) or Disallow (last) option, the preference is saved as a host setting. When the user next visits this web page, Internet Explorer compares the security policies for the zone and the host, and applies the most secure policy. For example, if security for the zone is set to Prompt, and security for the host is set to Disable (user selected Disable on a previous visit to the web page), Internet Explorer applies the most secure policy, does not display a prompt window to the user, and does not enable the service.

## Managing security policies

Custom security policies for web content zones are stored in the Windows registry for the current user. Skype4COM provides a window where you can view and change these settings. Type <skype4com:openzonesettings> into the Internet Explorer address bar to access the zone security settings.

Custom security policies entered by the user in prompt windows are stored in Windows registry for the current user. Skype4COM provides a window where you can view and change these settings. Type [skype4com:openhostsettings](#) into the Internet Explorer address bar to access the host security settings.

---

Copyright  2006 [Skype Limited](#). All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Deprecated List

---

## **Class Conversion**

The **SKYPE4COMLib::Conversion** class is deprecated, support for multilanguage status codes support will be dropped in the future.

## **Member ICall::Status**

Set **ICall::Status** is deprecated, use the **ICall::Hold**, **ICall::Resume**, **ICall::Finish** or **ICall::Answer** instead.

## **Member IChat::Status**

**ICall::Status** is deprecated, use the **IChat::Type** and **IChat::MyStatus** instead.

## **Member ISettings::Avatar ([in, defaultvalue("1")] BSTR Id)**

Use the **ISettings::LoadAvatarFromFile** instead.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<b><a href="#">Modules</a></b>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Modules

---

Here is a list of all modules:

- [Skype4COM](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

[Typedefs](#) | [Enumerations](#)

**Skype4COM**

---

## Typedefs

typedef enum <b>TAttachmentStatus</b>	<b>TAttachmentStatus</b>
typedef enum <b>TConnectionStatus</b>	<b>TConnectionStatus</b>
typedef enum <b>TUserStatus</b>	<b>TUserStatus</b>
typedef enum <b>TCallFailureReason</b>	<b>TCallFailureReason</b>
typedef enum <b>TCallStatus</b>	<b>TCallStatus</b>
typedef enum <b>TCallType</b>	<b>TCallType</b>
typedef enum <b>TCallHistory</b>	<b>TCallHistory</b>
typedef enum <b>TCallVideoStatus</b>	<b>TCallVideoStatus</b>
typedef enum <b>TCallVideoSendStatus</b>	<b>TCallVideoSendStatus</b>
typedef enum <b>TCalloDeviceType</b>	<b>TCalloDeviceType</b>
typedef enum <b>TChatMessageType</b>	<b>TChatMessageType</b>
typedef enum <b>TChatMessageStatus</b>	<b>TChatMessageStatus</b>
typedef enum <b>TChatMemberRole</b>	<b>TChatMemberRole</b>
typedef enum <b>TUserSex</b>	<b>TUserSex</b>
typedef enum <b>TBuddyStatus</b>	<b>TBuddyStatus</b>
typedef enum <b>TOnlineStatus</b>	<b>TOnlineStatus</b>
typedef enum <b>TChatLeaveReason</b>	<b>TChatLeaveReason</b>
typedef enum <b>TChatStatus</b>	<b>TChatStatus</b>
typedef enum <b>TChatType</b>	<b>TChatType</b>
typedef enum <b>TChatMyStatus</b>	<b>TChatMyStatus</b>
typedef enum <b>TChatOptions</b>	<b>TChatOptions</b>
typedef enum <b>TVoicemailType</b>	<b>TVoicemailType</b>
typedef enum <b>TVoicemailStatus</b>	<b>TVoicemailStatus</b>
typedef enum <b>TVoicemailFailureReason</b>	<b>TVoicemailFailureReason</b>
typedef enum <b>TGroupType</b>	<b>TGroupType</b>
typedef enum <b>TCallChannelType</b>	<b>TCallChannelType</b>
typedef enum <b>TApiSecurityContext</b>	<b>TApiSecurityContext</b>
typedef enum <b>TSmsMessageType</b>	<b>TSmsMessageType</b>
typedef enum <b>TSmsMessageStatus</b>	<b>TSmsMessageStatus</b>
typedef enum <b>TSmsFailureReason</b>	<b>TSmsFailureReason</b>
typedef enum <b>TSmsTargetStatus</b>	<b>TSmsTargetStatus</b>
typedef enum <b>TPluginContext</b>	<b>TPluginContext</b>

```
typedef enum TPluginContactType TPluginContactType
```

```
typedef enum TFileTransferType TFileTransferType
```

```
typedef enum TFileTransferStatus TFileTransferStatus
```

```
typedef enum TFileTransferFailureReason TFileTransferFailureReason
```



## Enumerations

enum	<pre>TAttachmentStatus {     apiAttachUnknown = -1, apiAttachSuccess = 0,     apiAttachPendingAuthorization = 1, apiAttachRefused =     2,     apiAttachNotAvailable = 3, apiAttachAvailable = 4 }</pre>
enum	<pre>TConnectionStatus {     conUnknown = -1, conOffline = 0, conConnecting = 1,     conPausing = 2,     conOnline = 3 }</pre>
enum	<pre>TUserStatus {     cusUnknown = -1, cusOffline = 0, cusOnline = 1,     cusAway = 2,     cusNotAvailable = 3, cusDoNotDisturb = 4, cusInvisible =     5, cusLoggedOut = 6,     cusSkypeMe = 7 }</pre>
enum	<pre>TCallFailureReason {     cfrUnknown = -1, cfrMiscError = 0, cfrUserDoesNotExist     = 1, cfrUserIsOffline = 2,     cfrNoProxyFound = 3, cfrSessionTerminated = 4,     cfrNoCommonCodec = 5, cfrSoundIOError = 6,     cfrRemoteDeviceError = 7, cfrBlockedByRecipient = 8,     cfrRecipientNotFriend = 9, cfrNotAuthorizedByRecipient =     10,     cfrSoundRecordingError = 11 }</pre>
	<pre>TCallStatus {     clsUnknown = -1, clsUnplaced = 0, clsRouting = 1,     clsEarlyMedia = 2,     clsFailed = 3, clsRinging = 4, clsInProgress = 5,     clsOnHold = 6,     clsFinished = 7, clsMissed = 8, clsRefused = 9, clsBusy =     10,</pre>

enum    **clsCancelled = 11, clsLocalHold = 12, clsRemoteHold = 13, clsVoicemailBufferingGreeting = 14, clsVoicemailPlayingGreeting = 15, clsVoicemailRecording = 16, clsVoicemailUploading = 17, clsVoicemailSent = 18, clsVoicemailCancelled = 19, clsVoicemailFailed = 20, clsTransferring = 21, clsTransferred = 22**  
}

enum    **TCallType { cltUnknown = -1, cltIncomingPSTN = 0, cltOutgoingPSTN = 1, cltIncomingP2P = 2, cltOutgoingP2P = 3**  
}

enum    **TCallHistory { chsAllCalls = 0, chsMissedCalls = 1, chsIncomingCalls = 2, chsOutgoingCalls = 3 }**

enum    **TCallVideoStatus { cvsUnknown = -1, cvsNone = 0, cvsSendEnabled = 1, cvsReceiveEnabled = 2, cvsBothEnabled = 3**  
}

enum    **TCallVideoSendStatus { vssUnknown = -1, vssNotAvailable = 0, vssAvailable = 1, vssStarting = 2, vssRejected = 3, vssRunning = 4, vssStopping = 5, vssPaused = 6**  
}

enum    **TCallloDeviceType { callloDeviceTypeUnknown = -1, callloDeviceTypeSoundcard = 0, callloDeviceTypePort = 1, callloDeviceTypeFile = 2 }**

enum    **TChatMessageType { cmeUnknown = -1, cmeCreatedChatWith = 0, cmeSawMembers = 1, cmeAddedMembers = 2, cmeSetTopic = 3, cmeSaid = 4, cmeLeft = 5, cmeEmoted = 6, cmePostedContacts = 7, cmeGapInChat = 8, cmeSetRole = 9, cmeKicked = 10, cmeSetOptions = 11, cmeKickBanned = 12, cmeJoinedAsApplicant = 13, cmeSetPicture = 14,**

```
    cmsSetGuidelines = 15  
}
```

```
enum TChatMessageStatus {  
    cmsUnknown = -1, cmsSending = 0, cmsSent = 1,  
    cmsReceived = 2,  
    cmsRead = 3  
}
```

```
enum TChatMemberRole {  
    chatMemberRoleUnknown = -1, chatMemberRoleCreator  
= 0, chatMemberRoleMaster = 1, chatMemberRoleHelper =  
2,  
    chatMemberRoleUser = 3, chatMemberRoleListener = 4,  
    chatMemberRoleApplicant = 5  
}
```

```
enum TUserSex { usexUnknown = -1, usexMale = 0, usexFemale  
= 1 }
```

```
enum TBuddyStatus {  
    budUnknown = -1, budNeverBeenFriend = 0,  
    budDeletedFriend = 1, budPendingAuthorization = 2,  
    budFriend = 3  
}
```

```
enum TOnlineStatus {  
    olsUnknown = -1, olsOffline = 0, olsOnline = 1, olsAway  
= 2,  
    olsNotAvailable = 3, olsDoNotDisturb = 4, olsSkypeOut =  
5, olsSkypeMe = 6  
}
```

```
enum TChatLeaveReason {  
    leaUnknown = -1, leaUserNotFound = 0,  
    leaUserIncapable = 1, leaAdderNotFriend = 2,  
    leaAddedNotAuthorized = 3, leaAddDeclined = 4,  
    leaUnsubscribe = 5  
}
```

```
enum TChatStatus {  
    chsUnknown = -1, chsLegacyDialog = 0, chsDialog = 1,  
    chsMultiNeedAccept = 2,  
    chsMultiSubscribed = 3, chsUnsubscribed = 4  
}
```

enum **TChatType** {  
    **chatTypeUnknown** = -1, **chatTypeDialog** = 0,  
    **chatTypeLegacyDialog** = 1, **chatTypeLegacyUnsubscribed**  
    = 2,  
    **chatTypeMultiChat** = 3, **chatTypeSharedGroup** = 4  
}

enum **TChatMyStatus** {  
    **chatStatusUnknown** = -1, **chatStatusConnecting** = 0,  
    **chatStatusWaitingRemoteAccept** = 1,  
    **chatStatusAcceptRequired** = 2,  
    **chatStatusPasswordRequired** = 3, **chatStatusSubscribed**  
    = 4, **chatStatusUnsubscribed** = 5, **chatStatusDisbanded** =  
    6,  
    **chatStatusQueuedBecauseChatsFull** = 7,  
    **chatStatusApplicationDenied** = 8, **chatStatusKicked** = 9,  
    **chatStatusBanned** = 10,  
    **chatStatusRetryConnecting** = 11  
}

enum **TChatOptions** {  
    **chatOptionJoiningEnabled** = 1,  
    **chatOptionJoinersBecomeApplicants** = 2,  
    **chatOptionJoinersBecomeListeners** = 4,  
    **chatOptionHistoryDisclosed** = 8,  
    **chatOptionUsersAreListeners** = 16,  
    **chatOptionTopicAndPictureLockedForUsers** = 32  
}

enum **TVoicemailType** {  
    **vmtUnknown** = -1, **vmtIncoming** = 0, **vmtDefaultGreeting**  
    = 1, **vmtCustomGreeting** = 2,  
    **vmtOutgoing** = 3  
}

enum **TVoicemailStatus** {  
    **vmsUnknown** = -1, **vmsNotDownloaded** = 0,  
    **vmsDownloading** = 1, **vmsUnplayed** = 2,  
    **vmsBuffering** = 3, **vmsPlaying** = 4, **vmsPlayed** = 5,  
    **vmsBlank** = 6,  
    **vmsRecording** = 7, **vmsRecorded** = 8, **vmsUploading** = 9,  
    **vmsUploaded** = 10,  
}

```
    vmsDeleting = 11, vmsFailed = 12
}
```

```
enum TVoicemailFailureReason {
    vmrUnknown = -1, vmrNoError = 0, vmrMiscError = 1,
    vmrConnectError = 2,
    vmrNoPrivilege = 3, vmrNoVoicemail = 4,
    vmrFileReadError = 5, vmrFileWriteError = 6,
    vmrRecordingError = 7, vmrPlaybackError = 8
}
```

```
enum TGroupType {
    grpUnknown = -1, grpCustomGroup = 0, grpAllUsers = 1,
    grpAllFriends = 2,
    grpSkypeFriends = 3, grpSkypeOutFriends = 4,
    grpOnlineFriends = 5, grpPendingAuthorizationFriends =
    6,
    grpRecentlyContactedUsers = 7,
    grpUsersWaitingMyAuthorization = 8,
    grpUsersAuthorizedByMe = 9, grpUsersBlockedByMe =
    10,
    grpUngroupedFriends = 11, grpSharedGroup = 12,
    grpProposedSharedGroup = 13
}
```

```
enum TCallChannelType { cctUnknown = -1, cctDatagram = 0,
    cctReliable = 1 }
```

```
enum TApiSecurityContext {
    apiContextUnknown = 0x0000, apiContextVoice =
    0x0001, apiContextMessaging = 0x0002,
    apiContextAccount = 0x0004,
    apiContextContacts = 0x0008
}
```

```
enum TSmsMessageType {
    smsMessageTypeUnknown = -1,
    smsMessageTypeIncoming = 0,
    smsMessageTypeOutgoing = 1,
    smsMessageTypeCCRequest = 2,
    smsMessageTypeCCSubmit = 3
}
```

```
TSmsMessageStatus {
```

```
enum    smsMessageStatusUnknown = -1,  
        smsMessageStatusReceived = 0,  
        smsMessageStatusRead = 1,  
        smsMessageStatusComposing = 2,  
        smsMessageStatusSendingToServer = 3,  
        smsMessageStatusSentToServer = 4,  
        smsMessageStatusDelivered = 5,  
        smsMessageStatusSomeTargetsFailed = 6,  
        smsMessageStatusFailed = 7  
}
```

```
enum    TSmsFailureReason {  
        smsFailureReasonUnknown = -1,  
        smsFailureReasonMiscError = 0,  
        smsFailureReasonServerConnectFailed = 1,  
        smsFailureReasonNoSmsCapability = 2,  
        smsFailureReasonInsufficientFunds = 3,  
        smsFailureReasonInvalidConfirmationCode = 4,  
        smsFailureReasonUserBlocked = 5,  
        smsFailureReasonIPBlocked = 6,  
        smsFailureReasonNodeBlocked = 7  
}
```

```
enum    TSmsTargetStatus {  
        smsTargetStatusUnknown = -1,  
        smsTargetStatusUndefined = 0,  
        smsTargetStatusAnalyzing = 1,  
        smsTargetStatusAcceptable = 2,  
        smsTargetStatusNotRoutable = 3,  
        smsTargetStatusDeliveryPending = 4,  
        smsTargetStatusDeliverySuccessful = 5,  
        smsTargetStatusDeliveryFailed = 6  
}
```

```
enum    TPluginContext {  
        pluginContextUnknown = -1, pluginContextChat = 0,  
        pluginContextCall = 1, pluginContextContact = 2,  
        pluginContextMyself = 3, pluginContextTools = 4  
}
```

```
enum    TPluginContactType { pluginContactTypeUnknown = -1,  
        pluginContactTypeAll = 0, pluginContactTypeSkype = 1,
```

	<b>pluginContactTypeSkypeOut = 2 }</b>
enum	<b>TFileTransferType { fileTransferTypeIncoming = 0, fileTransferTypeOutgoing = 1 }</b>
enum	<b>TFileTransferStatus { fileTransferStatusNew = 0, fileTransferStatusConnecting = 1, fileTransferStatusWaitingForAccept = 2, fileTransferStatusTransferring = 3, fileTransferStatusTransferringOverRelay = 4, fileTransferStatusPaused = 5, fileTransferStatusRemotelyPaused = 6, fileTransferStatusCancelled = 7, fileTransferStatusCompleted = 8, fileTransferStatusFailed = 9 }</b>
enum	<b>TFileTransferFailureReason { fileTransferFailureReasonSenderNotAuthorized = 1, fileTransferFailureReasonRemotelyCancelled = 2, fileTransferFailureReasonFailedRead = 3, fileTransferFailureReasonFailedRemoteRead = 4, fileTransferFailureReasonFailedWrite = 5, fileTransferFailureReasonFailedRemoteWrite = 6, fileTransferFailureReasonRemoteDoesNotSupportFT = 7, fileTransferFailureReasonRemoteOfflineTooLong = 8 }</b>

---

## Detailed Description

Skype4COM is a COM library that enables you to develop applications using the [Skype API](#) in a range of ActiveX programming environments and scripting languages.

---



## Enumeration Type Documentation

### enum **TApiSecurityContext**

contains API security contexts for Internet Explorer.

#### Enumerator:

<i>apiContextUnknown</i>	indicates that the API security context is unknown.
<i>apiContextVoice</i>	relates to call and voicemail commands and events
<i>apiContextMessaging</i>	relates to text and app2app messaging commands and events
<i>apiContextAccount</i>	relates to user profile and global settings commands and events
<i>apiContextContacts</i>	relates to contacts commands and events

### enum **TAttachmentStatus**

contains the status of an attachment with the Skype API.

#### Enumerator:

<i>apiAttachUnknown</i>	indicates that the attachment status is unknown.
-------------------------	--

<i>apiAttachSuccess</i>	indicates that the client is successfully attached.
<i>apiAttachPendingAuthorization</i>	indicates that Skype has acknowledged a connection request and is waiting for user authorisation.
<i>apiAttachRefused</i>	indicates that a user has explicitly denied access to the client.
<i>apiAttachNotAvailable</i>	indicates that the Skype API is not available.
<i>apiAttachAvailable</i>	indicates that the Skype API is available.

## **enum TBuddyStatus**

contains the status of a user.

### **Enumerator:**

<i>budUnknown</i>	indicates the status of a contact is unknown.
<i>budNeverBeenFriend</i>	indicates the contact has never been in the user's contactlist.

<i>budDeletedFriend</i>	indicates the contact was deleted from the contact list.
<i>budPendingAuthorization</i>	indicates the contact is pending authorization.
<i>budFriend</i>	indicates the contact is in a contact list.

### enum TCallChannelType

contains the call channel type.

#### Enumerator:

<i>cctUnknown</i>	indicates that the call channel type is unknown.
<i>cctDatagram</i>	indicates that the call channel is a datagram type channel.
<i>cctReliable</i>	indicates that the call channel is a reliable type channel.

### enum TCallFailureReason

contains the reason a call fails.

#### Enumerator:

<i>cfrUnknown</i>	indicates the reason for call failure is unknown.
-------------------	---

<i>cfrMiscError</i>	indicates the call failed because of a miscellaneous error.
<i>cfrUserDoesNotExist</i>	indicates the call failed because the user or phone number does not exist.
<i>cfrUserIsOffline</i>	indicates the call failed because the user is offline.
<i>cfrNoProxyFound</i>	indicates the call failed because no proxy was found.
<i>cfrSessionTerminated</i>	indicates the call failed because the session terminated.
<i>cfrNoCommonCodec</i>	indicates the call failed because no common codec was found.
<i>cfrSoundIOError</i>	indicates the call failed because of an audio I/O error.
<i>cfrRemoteDeviceError</i>	indicates the call failed because of a problem with the remote sound device.
<i>cfrBlockedByRecipient</i>	indicates the recipient blocked the call.

<i>cfrRecipientNotFriend</i>	indicates the call failed because the recipient is not a friend.
<i>cfrNotAuthorizedByRecipient</i>	indicates the call failed because the user is not authorized by the recipient.
<i>cfrSoundRecordingError</i>	indicates the call failed because of a sound recording error.

### enum TCallHistory

defines call types in call history.

#### Enumerator:

<i>chsAllCalls</i>	indicates all calls in call history.
<i>chsMissedCalls</i>	indicates all missed calls in call history.
<i>chsIncomingCalls</i>	indicates all incoming calls in call history.
<i>chsOutgoingCalls</i>	indicates all outgoing calls in call history.

### enum TCallIoDeviceType

contains the call io device types.

#### Enumerator:

*callloDeviceTypeUnknown* indicates that device is unknown.

*callloDeviceTypeSoundcard* indicates soundcard.

*callloDeviceTypePort* indicates local port.

*callloDeviceTypeFile* indicates local file.

## enum TCallStatus

contains the status of a call.

### Enumerator:

*clsUnknown* indicates the call status is unknown.

*clsUnplaced* indicates the call was not placed.

*clsRouting* indicates the call is in the process of being routed.

*clsEarlyMedia* indicates that early media is currently being played, such as recorded speech.

*clsFailed* indicates the call failed.

<i>clsRinging</i>	indicates the call is in the process of ringing.
<i>clsInProgress</i>	indicates the call is in progress.
<i>clsOnHold</i>	indicates the call is placed on hold.
<i>clsFinished</i>	indicates the call is finished.
<i>clsMissed</i>	indicates the call was missed.
<i>clsRefused</i>	indicates the call was refused.
<i>clsBusy</i>	indicates the call destination was busy.
<i>clsCancelled</i>	indicates the call was cancelled.
<i>clsLocalHold</i>	indicates the call is placed on hold by the local user.
<i>clsRemoteHold</i>	indicates the call is placed on hold by the remote user.
<i>clsVoicemailBufferingGreeting</i>	indicates that voicemail is preparing a greeting.
	indicates that voicemail is

<i>clsVoicemailPlayingGreeting</i>	playing a greeting.
<i>clsVoicemailRecording</i>	indicates that voicemail is recording a message.
<i>clsVoicemailUploading</i>	indicates that voicemail is being loaded to the server.
<i>clsVoicemailSent</i>	indicates that voicemail has been sent.
<i>clsVoicemailCancelled</i>	indicates that voicemail has been cancelled.
<i>clsVoicemailFailed</i>	indicates that voicemail failed.
<i>clsTransferring</i>	indicates that call is being transferred.
<i>clsTransferred</i>	indicates that call has been transferred.

## enum **TCallType**

contains information about call types.

### **Enumerator:**

*cltUnknown* indicates the call type is unknown.



*cltIncomingPSTN* indicates the call is an incoming call from the PSTN.

*cltOutgoingPSTN* indicates the call is an outgoing call to the PSTN.

*cltIncomingP2P* indicates the call is an incoming Skype call.

*cltOutgoingP2P* indicates the call is an outgoing Skype call.

## enum TCallVideoSendStatus

contains the status for video send streams.

### Enumerator:

*vssUnknown* indicates that video send status is unknown.

*vssNotAvailable* indicates that video send is not available because video is disabled or a webcam is unplugged.

*vssAvailable* indicates that video send is available but the video is not running.

*vssStarting* indicates that video is sending but not yet running at full speed.

*vssRejected* indicates that video send was rejected by the receiver.

<i>vssRunning</i>	indicates that video send is actively running.
<i>vssStopping</i>	indicates that video send is in the process of stopping but not halted yet.
<i>vssPaused</i>	indicates that video send is placed on hold.

### **enum TCallVideoStatus**

contains the video status of a call.

#### **Enumerator:**

<i>cvsUnknown</i>	indicates that call video status is unknown.
<i>cvsNone</i>	indicates that call video status is that video is not enabled.
<i>cvsSendEnabled</i>	indicates that call video status is that video send is enabled.
<i>cvsReceiveEnabled</i>	indicates that call video status is that video receive is enabled.
<i>cvsBothEnabled</i>	indicates that call video status is that video send and receive are enabled.

## enum TChatLeaveReason

contains the reason a user leaves a chat.

### Enumerator:

<i>leaUnknown</i>	indicates the reason for leaving a chat is unknown.
<i>leaUserNotFound</i>	indicates the reason for leaving a chat is that a user was not found.
<i>leaUserIncapable</i>	indicates the reason for leaving a chat is that a user has an old version of Skype which does not support multichat.
<i>leaAdderNotFriend</i>	indicates the reason for leaving a chat is that the recipient accepts messages from contacts only and the sender is not in this contact list.
<i>leaAddedNotAuthorized</i>	indicates the reason for leaving a chat is the recipient accepts authorised contacts only and has not authorised the sender.
<i>leaAddDeclined</i>	indicates that the invited user declined the chat invitation.

### Note:

This message type existed in earlier versions of Skype, and is not used any more.

*leaUnsubscribe*

indicates the reason for leaving is the user unsubscribed from the chat.

## enum TChatMemberRole

Chat member roles.

### Enumerator:

*chatMemberRoleUnknown* Unknown role.

*chatMemberRoleCreator* Chat creator.

*chatMemberRoleMaster* Chat master.

*chatMemberRoleHelper* Chat helper.

*chatMemberRoleUser* Chat user.

*chatMemberRoleListener* Chat listener.

*chatMemberRoleApplicant* Chat applicant.

## enum TChatMessageStatus

contains the status of a chat message.

### Enumerator:

*cmsUnknown* indicates the status of the chat message is unknown.

*cmsSending* indicates the message is in the process of being sent.

*cmsSent* indicates the message has been sent.

*cmsReceived* indicates the message has been received.

*cmsRead* indicates the message has been read.

## enum TChatMessageType

Chat message types.

### Enumerator:

*cmeUnknown* Unknown message type.

*cmeCreatedChatWith* Chat created.

*cmeSawMembers* Saw chat members.

*cmeAddedMembers* Added chat members.

*cmeSetTopic* Chat topic changed.

<i>cmeSaid</i>	Sent message.
<i>cmeLeft</i>	Left the chat.
<i>cmeEmoted</i>	Sent emoted message.
<i>cmePostedContacts</i>	Posted contacts.
<i>cmeGapInChat</i>	Indicates gap in chat.
<i>cmeSetRole</i>	Member role changed.
<i>cmeKicked</i>	Member kicked from chat.
<i>cmeSetOptions</i>	Chat options changed.
<i>cmeKickBanned</i>	Member kicked and banned from chat.
<i>cmeJoinedAsApplicant</i>	Member joined as applicant.
<i>cmeSetPicture</i>	Chat picture changed.
<i>cmeSetGuidelines</i>	Chat guidelines changed.

## **enum TChatMyStatus**

---

Chat status.

**Enumerator:**

<i>chatStatusUnknown</i>	Unknown status.
<i>chatStatusConnecting</i>	Connecting to chat.
<i>chatStatusWaitingRemoteAccept</i>	Waiting for accept.
<i>chatStatusAcceptRequired</i>	Accept is required.
<i>chatStatusPasswordRequired</i>	Password is required.
<i>chatStatusSubscribed</i>	Subscribed to chat.
<i>chatStatusUnsubscribed</i>	Unsubscribed from chat.
<i>chatStatusDisbanded</i>	Chat was disbanded.
<i>chatStatusQueuedBecauseChatIsFull</i>	Queued because the chat is full.
<i>chatStatusApplicationDenied</i>	Application was denied.
<i>chatStatusKicked</i>	Kicked from chat.
<i>chatStatusBanned</i>	Banned from chat.

*chatStatusRetryConnecting*

Trying to reconnect.

## enum TChatOptions

Chat options.

### Enumerator:

*chatOptionJoiningEnabled*

Joining enabled.

*chatOptionJoinersBecomeApplicants*

Joiners become applicants.

*chatOptionJoinersBecomeListeners*

Joiners become listeners.

*chatOptionHistoryDisclosed*

History disclosed.

*chatOptionUsersAreListeners*

Users are listeners.

*chatOptionTopicAndPictureLockedForUsers*

Topic and picture are locked for users.

## enum TChatStatus

contains the status of a chat.



### Enumerator:

<i>chsUnknown</i>	indicates the status of a chat is unknown.
<i>chsLegacyDialog</i>	indicates the chat is in a legacy window mode which does not support multichats.
<i>chsDialog</i>	indicates the chat is in a chat window.
<i>chsMultiNeedAccept</i>	indicates the chat is waiting for acceptance by multiple users.
<i>chsMultiSubscribed</i>	indicates the chat is subscribed to by multiple users.
<i>chsUnsubscribed</i>	indicates the chat is unsubscribed.

### enum TChatType

Chat types.

### Enumerator:

<i>chatTypeUnknown</i>	Unknown type.
<i>chatTypeDialog</i>	Chat with two participants.
<i>chatTypeLegacyDialog</i>	Legacy dialog.

*chatTypeLegacyUnsubscribed* Legacy dialog (unsubscribed).

*chatTypeMultiChat* Chat with multiple members.

*chatTypeSharedGroup* Chat used to synchronize shared groups.

### enum TConnectionStatus

indicates the status of a network connection.

#### Enumerator:

*conUnknown* indicates that the status of a connection is unknown.

*conOffline* indicates that a connection does not exist.

*conConnecting* indicates that a connection is in the process of being established.

*conPausing* indicates that a connection is pausing.

*conOnline* indicates that a connection exists.

### enum TFileTransferFailureReason

File transfer failure reason codes.

## Enumerator:

<i>fileTransferFailureReasonSenderNotAuthorized</i>	Sender is not authorized by remote user.
<i>fileTransferFailureReasonRemotelyCancelled</i>	Remote user cancelled the file transfer.
<i>fileTransferFailureReasonFailedRead</i>	File read error.
<i>fileTransferFailureReasonFailedRemoteRead</i>	Remote file read error.
<i>fileTransferFailureReasonFailedWrite</i>	File write error.
<i>fileTransferFailureReasonFailedRemoteWrite</i>	Remote file write error.
<i>fileTransferFailureReasonRemoteDoesNotSupportFT</i>	Remote party does not support file

	transfer.
<i>fileTransferFailureReasonRemoteOfflineTooLong</i>	Remote party has been offline too long.

## enum TFileTransferStatus

File transfer status.

### Enumerator:

<i>fileTransferStatusNew</i>	A new file transfer.
<i>fileTransferStatusConnecting</i>	Connecting to user.
<i>fileTransferStatusWaitingForAccept</i>	Waiting user to accept the file transfer.
<i>fileTransferStatusTransferring</i>	The file is beeing transferred.
<i>fileTransferStatusTransferringOverRelay</i>	The file is beeing transferred over relay.
<i>fileTransferStatusPaused</i>	The file transfer is paused.

*fileTransferStatusRemotelyPaused*

The file transfer is paused by remote user.

*fileTransferStatusCancelled*

The file transfer has been cancelled.

*fileTransferStatusCompleted*

The file transfer completed.

*fileTransferStatusFailed*

The file transfer failed.

### enum TFileTransferType

File transfer types.

#### Enumerator:

*fileTransferTypeIncoming* Incoming file transfer.

*fileTransferTypeOutgoing* Outgoing file transfer.

### enum TGroupType

contains the group type.

#### Enumerator:

*grpUnknown*

indicates the group type is

unknown.

*grpCustomGroup*

indicates the group type is user-defined.

*grpAllUsers*

indicates the group type is all other users a user knows about, including users in the contactlist, users recently contacted, and blocked users.

*grpAllFriends*

indicates the group type is all contacts in the user's contactlists.

*grpSkypeFriends*

indicates the group type is Skype contacts in the contactlist.

*grpSkypeOutFriends*

indicates the group type is SkypeOut contacts in the contactlist.

*grpOnlineFriends*

indicates the group type is Skype contacts in the contactlist who are online.

*grpPendingAuthorizationFriends*

indicates the group type is contacts in the contactlist who have not yet authorized the user.

<i>grpRecentlyContactedUsers</i>	indicates the group type is contacts the user has conversed with recently, including non-friends.
<i>grpUsersWaitingMyAuthorization</i>	indicates the group type is contacts who are awaiting response to an authorisation request, including non-friends.
<i>grpUsersAuthorizedByMe</i>	indicates the group type is all authorised contacts, including non-friends.
<i>grpUsersBlockedByMe</i>	indicates the group type is all blocked contacts, including non-friends.
<i>grpUngroupedFriends</i>	indicates the group type is all contacts in the contactlist that do not belong to any custom group.
<i>grpSharedGroup</i>	indicates the group type is shared group.
<i>grpProposedSharedGroup</i>	indicates the group type is proposed shared group.

## enum TOnlineStatus

contains the online status of a user.

### Enumerator:

<i>olsUnknown</i>	indicates the online status of the user is unknown.
<i>olsOffline</i>	indicates the user is offline (not connected).
<i>olsOnline</i>	indicates the user is online.
<i>olsAway</i>	indicates the user is away (has been inactive for certain period).
<i>olsNotAvailable</i>	indicates the user is not available.
<i>olsDoNotDisturb</i>	indicates the user is in do not disturb mode.
<i>olsSkypeOut</i>	indicates the user is in the SkypeOut contact list.
<i>olsSkypeMe</i>	indicates the user is in SkypeMe mode.

## enum TPluginContactType

This property contains the contact type for the TPluginContext::pluginContextContact



### Enumerator:

<i>pluginContactTypeUnknown</i>	This value indicates that the plug-in contact type is undefined.
<i>pluginContactTypeAll</i>	This value indicates that the plug-in is available for all contacts.
<i>pluginContactTypeSkype</i>	This value indicates that the plug-in is available for Skype contacts.
<i>pluginContactTypeSkypeOut</i>	This value indicates that the plug-in is available for SkypeOut contacts.

### enum TPluginContext

contains the plug-in context.

### Enumerator:

<i>pluginContextUnknown</i>	This value indicates that the plug-in context is undefined.
<i>pluginContextChat</i>	This value indicates that the plug-in works with chats.
<i>pluginContextCall</i>	This value indicates that the plug-in works with calls.
	This value indicates that the plug-in

<i>pluginContextContact</i>	works with contact list items.
<i>pluginContextMyself</i>	This value indicates that the plug-in is a personal plug-in that works for a user who is logged into Skype.
<i>pluginContextTools</i>	This value indicates that the plug-in is available as a generic tool.

## enum TSmsFailureReason

contains the reason an SMS message failed.

### Enumerator:

<i>smsFailureReasonUnknown</i>	indicates the SMS message failed for an unknown reason.
<i>smsFailureReasonMiscError</i>	indicates the SMS message failed due to a miscellaneous error.
<i>smsFailureReasonServerConnectFailed</i>	indicates a failure to connect to the server.
	indicates the user

*smsFailureReasonNoSmsCapability*

has no SMS capability.

*smsFailureReasonInsufficientFunds*

indicates the user does not have sufficient funds to send an SMS message.

*smsFailureReasonInvalidConfirmationCode*

indicates the SMS message has an invalid confirmation code.

*smsFailureReasonUserBlocked*

indicates the SMS message failed because the user is blocked by the SMS service.

*smsFailureReasonIPBlocked*

indicates the SMS message failed because the IP address is blocked by the SMS service.

*smsFailureReasonNodeBlocked*

indicates the SMS message failed because the node is blocked by the SMS service.

## enum TSmsMessageStatus

contains the status of an SMS message.

### Enumerator:

<i>smsMessageStatusUnknown</i>	indicates the SMS message status is unknown.
<i>smsMessageStatusReceived</i>	indicates the SMS message was received.
<i>smsMessageStatusRead</i>	indicates the SMS message was read.
<i>smsMessageStatusComposing</i>	indicates the user is composing a SMS message.
<i>smsMessageStatusSendingToServer</i>	indicates that Skype is sending an SMS message to the server.
<i>smsMessageStatusSentToServer</i>	indicates an SMS message was sent to the server.
<i>smsMessageStatusDelivered</i>	indicates an SMS message was delivered.

*smsMessageStatusSomeTargetsFailed* indicates an SMS message failed to reach some targets.

*smsMessageStatusFailed* indicates an SMS message failed.

### enum TSmsMessageType

contains the type of the SMS message.

#### Enumerator:

*smsMessageTypeUnknown* indicates the SMS message is of an unknown type.

*smsMessageTypeIncoming* indicates an incoming SMS message.

*smsMessageTypeOutgoing* indicates an outgoing SMS message.

*smsMessageTypeCCRequest* indicates the SMS message is a confirmation code request.

*smsMessageTypeCCSubmit* indicates the SMS message is a confirmation code reply.

### enum TSmsTargetStatus

contains the status of the target of an SMS message.

**Enumerator:**

<i>smsTargetStatusUnknown</i>	indicates the status of the target is unknown.
<i>smsTargetStatusUndefined</i>	indicates the status of the target is undefined.
<i>smsTargetStatusAnalyzing</i>	indicates that Skype is analyzing the target number.
<i>smsTargetStatusAcceptable</i>	Indicates the target number is acceptable.
<i>smsTargetStatusNotRoutable</i>	indicates that there is no route to the target.
<i>smsTargetStatusDeliveryPending</i>	indicates that delivery to the target is pending.
<i>smsTargetStatusDeliverySuccessful</i>	indicates the message was delivered to the target.
<i>smsTargetStatusDeliveryFailed</i>	indicates the SMS message failed to be delivered to the target.

## enum TUserSex

contains information about the sex of a user.

### Enumerator:

*usexUnknown* indicates the sex of the user is unknown.

*usexMale* indicates the user is male.

*usexFemale* indicates the user is female.

## enum TUserStatus

contains the Skype user status.

### Enumerator:

*cusUnknown* indicates the user status is unknown.

*cusOffline* indicates the user is offline.

*cusOnline* indicates the user is online.

*cusAway* indicates the user is away.

*cusNotAvailable* indicates the user is not available.

*cusDoNotDisturb* indicates the user is in do not disturb mode.

*cusInvisible* indicates the user is invisible to others.

*cusLoggedOut* indicates the user is logged out.

*cusSkypeMe* indicates the user is in SkypeMe mode.

## enum **TVoicemailFailureReason**

contains the reason for voicemail failure.

Voicemail failure reasons.

### **Enumerator:**

*vmrUnknown* indicates the reason for failure is unknown.

*vmrNoError* indicates the reason for failure is none.

*vmrMiscError* indicates the reason for failure is due to a miscellaneous error.

*vmrConnectError* indicates the reason for failure is due to a connection error.

*vmrNoPrivilege* indicates the reason for failure is the user has no voicemail service.

*vmrNoVoicemail* indicates the reason for failure is there is no such voicemail.



*vmrFileReadError* indicates the reason for failure is due to an error reading the file.

*vmrFileWriteError* indicates the reason for failure is due to an error writing the file.

*vmrRecordingError* indicates the reason for failure is due to a recording error.

*vmrPlaybackError* indicates the reason for failure is due to a playback error.

## **enum TVoicemailStatus**

contains the status of a voicemail.

Voicemail status.

### **Enumerator:**

*vmsUnknown* indicates that voicemail status is unknown.

*vmsNotDownloaded* indicates that voicemail status is not downloaded.

*vmsDownloading* indicates that voicemail status is in the process of downloading.

*vmsUnplayed* indicates that voicemail status is not played.

<i>vmsBuffering</i>	indicates that voicemail status is buffering.
<i>vmsPlaying</i>	indicates that voicemail status is playing.
<i>vmsPlayed</i>	indicates that voicemail status has been played.
<i>vmsBlank</i>	indicates that voicemail status is blank.
<i>vmsRecording</i>	indicates that voicemail status is recording.
<i>vmsRecorded</i>	indicates that voicemail status has been recorded.
<i>vmsUploading</i>	indicates that voicemail status is uploading.
<i>vmsUploaded</i>	indicates that voicemail status has been uploaded.
<i>vmsDeleting</i>	indicates that voicemail status is in the process of being deleted.
<i>vmsFailed</i>	indicates that voicemail status is failed.

## enum TVoicemailType

defines voicemail types.

### Enumerator:

<i>vmtUnknown</i>	indicates the voicemail type is unknown.
<i>vmtIncoming</i>	indicates the voicemail type is incoming.
<i>vmtDefaultGreeting</i>	indicates the voicemail type is the default voicemail greeting.
<i>vmtCustomGreeting</i>	indicates the voicemail type is a custom voicemail greeting.
<i>vmtOutgoing</i>	indicates the voicemail type is an outgoing voicemail.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>_ICallChannelManagerEvents</b>	<b>CallChannelManager</b> event source interface
<b>_ISkypeEvents</b>	<b>Skype</b> event source interface
<b>Application</b>	This class enables two applications sharing the same name to communicate over the <b>Skype</b> network
<b>ApplicationStream</b>	This class manages application streams
<b>ApplicationStreamCollection</b>	This class manages application stream collections
<b>Call</b>	This class manages calls and call properties
<b>CallChannel</b>	This class forms a logical connection between the <b>ApplicationStream</b> and <b>Call</b> objects
<b>CallChannelCollection</b>	Container class for <b>CallChannel</b> objects
<b>CallChannelManager</b>	This class manages <b>Call</b> data channels
<b>CallChannelMessage</b>	This class manages <b>CallChannel</b> messages
<b>CallCollection</b>	This class manages <b>Call</b> collections
<b>Chat</b>	This class manages chats
<b>ChatCollection</b>	This class manages <b>Chat</b> collections

<b>ChatMember</b>	This class manages chat member
<b>ChatMemberCollection</b>	This class manages chat member collections
<b>ChatMessage</b>	This class manages chat messages
<b>ChatMessageCollection</b>	This class manages chat message collections
<b>Client</b>	This class controls features related to the <b>Skype</b> user interface
<b>Command</b>	This class manages <b>Skype</b> API command objects
<b>Conference</b>	This class manages conference calls
<b>ConferenceCollection</b>	This class manages conference collections
<b>Conversion</b>	This class converts text to constant values and converts constant values to text
<b>Group</b>	This class manages contact groups
<b>GroupCollection</b>	This class manages collections of <b>Group</b> objects
<b>IApplication</b>	<b>Application</b> class default interface
<b>IApplicationStream</b>	<b>ApplicationStream</b> class default interface
<b>IApplicationStreamCollection</b>	<b>ApplicationStreamCollection</b> class default interface
<b>ICall</b>	<b>Call</b> class default interface
<b>ICallChannel</b>	<b>CallChannel</b> class default interface
	<b>CallChannelCollection</b> class

<b>ICallChannelCollection</b>	default interface
<b>ICallChannelManager</b>	<b>CallChannelManager</b> class default interface
<b>ICallChannelMessage</b>	<b>CallChannelMessage</b> class default interface
<b>ICallCollection</b>	<b>CallCollection</b> class default interface
<b>IChat</b>	<b>Chat</b> class default interface
<b>IChatCollection</b>	<b>ChatCollection</b> class default interface
<b>IChatMember</b>	<b>ChatMember</b> class default interface
<b>IChatMemberCollection</b>	<b>ChatMemberCollection</b> class default interface
<b>IChatMessage</b>	<b>ChatMessage</b> class default interface
<b>IChatMessageCollection</b>	<b>ChatMessageCollection</b> class default interface
<b>IClient</b>	<b>Client</b> class default interface
<b>ICommand</b>	<b>Command</b> class default interface
<b>IConference</b>	<b>Conference</b> class default interface
<b>IConferenceCollection</b>	<b>ConferenceCollection</b> class default interface
<b>IConversion</b>	<b>Conversion</b> class default interface
<b>IFileTransfer</b>	File transfer object default interface
<b>IFileTransferCollection</b>	File transfer collection class default interface
<b>IGroup</b>	<b>Group</b> class default interface
	<b>GroupCollection</b> class default

<b>IGroupCollection</b>	interface
<b>IParticipant</b>	<b>Participant</b> class default interface
<b>IParticipantCollection</b>	<b>ParticipantCollection</b> class default interface
<b>IPluginEvent</b>	This is the default interface for the <b>PluginEvent</b> class
<b>IPluginMenuItem</b>	Class <b>PluginMenuItem</b> default interface
<b>IProfile</b>	<b>Profile</b> class default interface
<b>ISettings</b>	<b>Settings</b> class default interface
<b>ISkype</b>	<b>Skype</b> is the default interface for the class
<b>ISmsChunk</b>	<b>SmsChunk</b> is the default interface for this class
<b>ISmsChunkCollection</b>	<b>SmsChunkCollection</b> is the default interface for this class
<b>ISmsMessage</b>	<b>SmsMessage</b> is the default interface for this class
<b>ISmsMessageCollection</b>	<b>SmsMessageCollection</b> is the default interface for this class
<b>ISmsTarget</b>	<b>SmsTarget</b> is the default interface for this class
<b>ISmsTargetCollection</b>	<b>SmsTargetCollection</b> is the default interface for this class
<b>IUser</b>	<b>User</b> class default interface. This interface enables you to read user properties
<b>IUserCollection</b>	<b>UserCollection</b> class default interface
<b>IVoicemail</b>	<b>Voicemail</b> class default interface
<b>IVoicemailCollection</b>	<b>VoicemailCollection</b> class



	default interface
<b>Participant</b>	This class manages participants in a conference call
<b>ParticipantCollection</b>	This class manages collections of participants in a conference call
<b>PluginEvent</b>	This class manages plug-in event
<b>PluginMenuItem</b>	Plugin menu item
<b>Profile</b>	This class manages the user profile
<b>Settings</b>	This class manages the <b>Skype</b> API general parameters
<b>Skype</b>	<b>Skype</b> API COM wrapper
<b>SmsChunk</b>	SMS message chunk
<b>SmsChunkCollection</b>	Collection class for <b>SmsChunk</b> objects
<b>SmsMessage</b>	SMS message
<b>SmsMessageCollection</b>	Collection class for SMS messages
<b>SmsTarget</b>	SMS message target
<b>SmsTargetCollection</b>	Collection class for SMS targets
<b>User</b>	This class manages user properties
<b>UserCollection</b>	<b>User</b> collection
<b>Voicemail</b>	This class manages <b>Skype</b> voicemail
<b>VoicemailCollection</b>	<b>Voicemail</b> collection

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#)

# **\_ICallChannelManagerEvents Interface Reference**

---

**CallChannelManager** event source interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Channels</b> ([in] <b>ICallChannelManager</b> *pManager,[in] <b>ICallChannelCollection</b> *pChannels) This event indicates a change in the call channels.
HRESULT	<b>Message</b> ([in] <b>ICallChannelManager</b> *pManager,[in] <b>ICallChannel</b> *pChannel,[in] <b>ICallChannelMessage</b> *pMessage) This event indicates that a call channel message has been received.
HRESULT	<b>Created</b> () This event indicates that the <b>Application</b> context was successfully created.

## Detailed Description

**CallChannelManager** event source interface.

---

## Member Function Documentation

**HRESULT Channels** ( [in] **ICallChannelManager** \* **pManager**,  
[in] **ICallChannelCollection** \* **pChannels**  
)

This event indicates a change in the call channels.

**Parameters:**

[in] **pManager** contains the interface pointer for the **CallChannelManager** object.

[in] **pChannels** contains the interface pointer for the **CallChannelCollection** object.

**HRESULT Message** ( [in] **ICallChannelManager** \* **pManager**,  
[in] **ICallChannel** \* **pChannel**,  
[in] **ICallChannelMessage** \* **pMessage**  
)

This event indicates that a call channel message has been received.

**Parameters:**

[in] **pManager** contains the interface pointer for the **CallChannelManager** object.

[in] **pChannel** contains the interface pointer for the **CallChannel** object.

[in] **pMessage** contains the interface pointer for the **CallChannelMessage** object.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Public Member Functions



## **\_ISkypeEvents Interface Reference**

---

**Skype** event source interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Command</b> ([in] <b>ICommand</b> *pCommand) This event indicates the command object to use with the Skype API.
HRESULT	<b>Reply</b> ([in] <b>ICommand</b> *pCommand) This event indicates the API response to the command object.
HRESULT	<b>Error</b> ([in] <b>ICommand</b> *pCommand,[in] LONG Number,[in] BSTR Description) This event indicates the error number and description associated with a bad command object.
HRESULT	<b>AttachmentStatus</b> ([in] <b>TAttachmentStatus</b> Status) This event is caused by a change in the status of an attachment to the Skype API.
HRESULT	<b>ConnectionStatus</b> ([in] <b>TConnectionStatus</b> Status) This event is caused by a connection status change.
HRESULT	<b>UserStatus</b> ([in] <b>TUserStatus</b> Status) This event is caused by a user status change.
HRESULT	<b>OnlineStatus</b> ([in] <b>IUser</b> *pUser,[in] <b>TOnlineStatus</b> Status) This event is caused by a change in the online status of a user.
HRESULT	<b>CallStatus</b> ([in] <b>ICall</b> *pCall,[in] <b>TCallStatus</b> Status) This event is caused by a change in call status.
HRESULT	<b>CallHistory</b> () This event is caused by a change in call history.
HRESULT	<b>Mute</b> ([in] VARIANT_BOOL Mute) This event is caused by a change in mute status.
HRESULT	<b>MessageStatus</b> ([in] <b>IChatMessage</b> *pMessage,[in] <b>TChatMessageStatus</b> Status) This event is caused by a change in chat message status.
HRESULT	<b>MessageHistory</b> ([in] BSTR Username)

		This event is caused by a change in message history.
HRESULT	<b>AutoAway</b> ([in] VARIANT_BOOL Automatic)	This event is caused by a change of auto away status.
HRESULT	<b>CallDtmfReceived</b> ([in] <b>ICall</b> *pCall,[in] BSTR Code)	This event is caused by a call DTMF event.
HRESULT	<b>VoicemailStatus</b> ([in] <b>IVoicemail</b> *pMail,[in] <b>TVoicemailStatus</b> Status)	This event is caused by a change in voicemail status.
HRESULT	<b>ApplicationConnecting</b> ([in] <b>IApplication</b> *pApp,[in] <b>IUserCollection</b> *pUsers)	This event is caused by users connecting to an application.
HRESULT	<b>ApplicationStreams</b> ([in] <b>IApplication</b> *pApp,[in] <b>IApplicationStreamCollection</b> *pStreams)	This event is caused by a change in application streams.
HRESULT	<b>ApplicationDatagram</b> ([in] <b>IApplication</b> *pApp,[in] <b>IApplicationStream</b> *pStream,[in] BSTR Text)	This event is caused by the arrival of an application datagram.
HRESULT	<b>ApplicationSending</b> ([in] <b>IApplication</b> *pApp,[in] <b>IApplicationStreamCollection</b> *pStreams)	This event is caused by a change of application sending streams.
HRESULT	<b>ApplicationReceiving</b> ([in] <b>IApplication</b> *pApp,[in] <b>IApplicationStreamCollection</b> *pStreams)	This event is caused by a change of application receiving streams.
HRESULT	<b>ContactsFocused</b> ([in] BSTR Username)	This event is caused by the contacts tab gaining or losing focus.
HRESULT	<b>GroupVisible</b> ([in] <b>IGroup</b> *pGroup,[in] VARIANT_BOOL Visible)	This event is caused by a user hiding/showing a group in the contacts tab.
HRESULT	<b>GroupExpanded</b> ([in] <b>IGroup</b> *pGroup,[in] VARIANT_BOOL Expanded)	

	This event is caused by a user expanding or collapsing a group in the contacts tab.
HRESULT	<b>GroupUsers</b> ([in] <b>IGroup</b> *pGroup,[in] <b>IUserCollection</b> *pUsers) This event is caused by a change in a contact group.
HRESULT	<b>GroupDeleted</b> ([in] LONG GroupId) This event is caused by a user deleting a custom contact group.
HRESULT	<b>UserMood</b> ([in] <b>IUser</b> *pUser,[in] BSTR MoodText) This event is caused by a change in the mood text of the user.
HRESULT	<b>SmsMessageStatusChanged</b> ([in] <b>ISmsMessage</b> *pMessage,[in] <b>TSmsMessageStatus</b> Status) This event is caused by a change in the SMS message status.
HRESULT	<b>SmsTargetStatusChanged</b> ([in] <b>ISmsTarget</b> *pTarget,[in] <b>TSmsTargetStatus</b> Status) This event is caused by a change in the SMS target status.
HRESULT	<b>CallInputStatusChanged</b> ([in] <b>ICall</b> *pCall,[in] VARIANT_BOOL Status) This event is caused by a change in the Call voice input status change.
HRESULT	<b>AsyncSearchUsersFinished</b> ([in] LONG Cookie,[in] <b>IUserCollection</b> *pUsers) This event occurs when a search is completed.
HRESULT	<b>CallSeenStatusChanged</b> ([in] <b>ICall</b> *pCall, VARIANT_BOOL Status) This event occurs when the seen status of a call changes.
HRESULT	<b>PluginEventClicked</b> ([in] <b>IPluginEvent</b> *pEvent) This event occurs when a user clicks on a plug-in event.
HRESULT	<b>PluginMenuItemClicked</b> ([in] <b>IPluginMenuItem</b> *pMenuItem,[in] <b>IUserCollection</b> *pUsers,[in] <b>TPluginContext</b> PluginContext,[in] BSTR ContextId) This event occurs when a user clicks on a menu item.

HRESULT	<b>WallpaperChanged</b> ([in] BSTR Path) This event occurs when a wallpaper changes.
HRESULT	<b>FileTransferStatusChanged</b> ([in] IFileTransfer *pTransfer,[in] TFileTransferStatus Status) This event occurs when a file transfer status changes.
HRESULT	<b>CallTransferStatusChanged</b> ([in] ICall *pCall,[in] TCallStatus Status) This event occurs when a call transfer status changes.
HRESULT	<b>ChatMembersChanged</b> ([in] IChat *pChat,[in] IUserCollection *pMembers) This event occurs when a chat members change.
HRESULT	<b>ChatMemberRoleChanged</b> ([in] IChatMember *pMember,[in] TChatMemberRole Role) This event occurs when a chat member role changes.
HRESULT	<b>CallVideoStatusChanged</b> ([in] ICall *pCall,[in] TCallVideoStatus Status) This event occurs when a call video status changes.
HRESULT	<b>CallVideoSendStatusChanged</b> ([in] ICall *pCall,[in] TCallVideoSendStatus Status) This event occurs when a call video send status changes.
HRESULT	<b>CallVideoReceiveStatusChanged</b> ([in] ICall *pCall,[in] TCallVideoSendStatus Status) This event occurs when a call video receive status changes.
HRESULT	<b>SilentModeStatusChanged</b> ([in] VARIANT_BOOL Silent) This event occurs when a silent mode is switched off.
HRESULT	<b>UILanguageChanged</b> ([in] BSTR Code) This event occurs when user changes Skype client language.
HRESULT	<b>UserAuthorizationRequestReceived</b> ([in] IUser *pUser) This event occurs when user sends you authorization request.

## Detailed Description

**Skype** event source interface.

**Examples:**

[Call.php](#).

---

## Member Function Documentation

**HRESULT ApplicationConnecting** ( [in] **IApplication** \* **pApp**,  
[in] **IUserCollection** \* **pUsers**  
)

This event is caused by users connecting to an application.

**Parameters:**

- [in] **pApp** contains the interface pointer for the **Application** object.
- [in] **pUsers** contains the interface pointer for the **UserCollection** object.

**HRESULT ApplicationDatagram** ( [in] **IApplication** \* **pApp**,  
[in] **IApplicationStream** \* **pStream**  
[in] **BSTR** **Text**  
)

This event is caused by the arrival of an application datagram.

**Parameters:**

- [in] **pApp** contains the interface pointer for the **Application** object.
- [in] **pStream** contains the interface pointer for the **ApplicationStream** object.
- [in] **Text** contains the datagram text.

**HRESULT ApplicationReceiving** ( [in] **IApplication** \*  
[in] **IApplicationStreamCollection**  
)

This event is caused by a change of application receiving streams.

**Parameters:**

- [in] **pApp** contains the interface pointer for the **Application** object.
- [in] **pStreams** contains the interface pointer for the **ApplicationStreamCollection** object.

**HRESULT ApplicationSending ( [in] IApplication \*  
[in] IApplicationStreamCollection \*  
)**

This event is caused by a change of application sending streams.

**Parameters:**

- [in] **pApp** contains the interface pointer for the **Application** object.
- [in] **pStreams** contains the interface pointer for the **ApplicationStreamCollection** object.

**HRESULT ApplicationStreams ( [in] IApplication \*  
[in] IApplicationStreamCollection \*  
)**

This event is caused by a change in application streams.

**Parameters:**

- [in] **pApp** contains the interface pointer for the **Application** object.
- [in] **pStreams** contains the interface pointer for the **ApplicationStreamCollection** object.

**HRESULT AsyncSearchUsersFinished ( [in] LONG  
[in] IUserCollection \* pUse  
)**



This event occurs when a search is completed.

**Parameters:**

[in] **Cookie** contains the search identifier.

[in] **pUsers** contains the interface pointer for the **UserCollection** object.

**See also:**

**ISkype::AsyncSearchUsers**

**HRESULT AttachmentStatus ( [in] TAttachmentStatus Status )**

This event is caused by a change in the status of an attachment to the Skype API.

**Parameters:**

[in] **Status** contains the new attachment status.

**Examples:**

**Call.php.**

**HRESULT AutoAway ( [in] VARIANT\_BOOL Automatic )**

This event is caused by a change of auto away status.

**Parameters:**

[in] **Automatic** contains the new auto away status.

**HRESULT CallDtmfReceived ( [in] ICall \* pCall,  
[in] BSTR Code  
)**

This event is caused by a call DTMF event.

**Parameters:**

[in] **pCall** contains the interface pointer for the **Call** object.

[in] **Code** contains the received DTMF code.

**Note:**

This event can be received with P2P calls.

```
HRESULT CallInputStatusChanged ( [in] ICall *          pCall,  
                                 [in] VARIANT_BOOL Status  
                                 )
```

This event is caused by a change in the Call voice input status change.

**Parameters:**

[in] **pCall** contains the interface pointer for the **Call** object.

[in] **Status** contains the voice input status (VARIANT\_TRUE: Active, VARIANT\_FALSE: Inactive).

**See also:**

[ICall::InputDevice](#)

```
HRESULT CallSeenStatusChanged ( [in] ICall *          pCall,  
                                VARIANT_BOOL Status  
                                )
```

This event occurs when the seen status of a call changes.

**Parameters:**

[in] **pCall** contains the interface pointer for the **Call** object.

[in] **Status** contains the call seen status (VARIANT\_TRUE: Seen, VARIANT\_FALSE: Unseen).

**See also:**

## ICall::Seen

```
HRESULT CallStatus ( [in] ICall *      pCall,  
                    [in] TCallStatus Status  
                    )
```

This event is caused by a change in call status.

### Parameters:

- [in] **pCall** receives the interface pointer for the **Call** object.
- [in] **Status** contains the new call status.

### Examples:

[Call.php](#).

```
HRESULT CallTransferStatusChanged ( [in] ICall *      pCall,  
                                    [in] TCallStatus Status  
                                    )
```

This event occurs when a call transfer status changes.

### Parameters:

- [in] **pCall** contains the interface pointer for the **Call** object.
- [in] **Status** contains the new call transfer status.

```
HRESULT CallVideoReceiveStatusChanged ( [in] ICall *  
                                         [in] TCallVideoSendS  
                                         )
```

This event occurs when a call video receive status changes.

### Parameters:

- [in] **pCall** contains the interface pointer for the **Call** object.

[in] **Status** contains new video receive status.

**HRESULT CallVideoSendStatusChanged** ( [in] **ICall** \*  
[in] **TCallVideoSendStatu**  
)

This event occurs when a call video send status changes.

**Parameters:**

[in] **pCall** contains the interface pointer for the Call object.

[in] **Status** contains new video send status.

**HRESULT CallVideoStatusChanged** ( [in] **ICall** \* **pCall**,  
[in] **TCallVideoStatus** **Status**  
)

This event occurs when a call video status changes.

**Parameters:**

[in] **pCall** contains the interface pointer for the Call object.

[in] **Status** contains new video status.

**HRESULT ChatMemberRoleChanged** ( [in] **IChatMember** \* **pMe**  
[in] **TChatMemberRole** **Role**  
)

This event occurs when a chat member role changes.

**Parameters:**

[in] **pMember** contains the interface pointer for the  
ChatMember object.

[in] **Role** contains the new role code.

**HRESULT ChatMembersChanged ( [in] IChat \* pChat,  
[in] IUserCollection \* pMembers  
)**

This event occurs when a chat members change.

**Parameters:**

- [in] **pChat** contains the interface pointer for the Chat object.
- [in] **pMembers** contains the interface pointer for the UserCollection object.

**HRESULT Command ( [in] ICommand \* pCommand )**

This event indicates the command object to use with the Skype API.

**Parameters:**

- [in] **pCommand** contains the **Command** object.

**HRESULT ConnectionStatus ( [in] TConnectionStatus Status )**

This event is caused by a connection status change.

**Parameters:**

- [in] **Status** contains the new connection status.

**HRESULT ContactsFocused ( [in] BSTR Username )**

This event is caused by the contacts tab gaining or losing focus.

**Parameters:**

- [in] **Username** is an optional field to enter the Skype name of the user.

```
HRESULT Error ( [in] ICommand * pCommand,  
                [in] LONG      Number,  
                [in] BSTR      Description  
                )
```

This event indicates the error number and description associated with a bad command object.

**Parameters:**

- [in] **pCommand** contains the **Command** object.
- [in] **Number** contains the error number to be returned by the Skype API.
- [in] **Description** contains a description of the error.

```
HRESULT FileTransferStatusChanged ( [in] IFileTransfer * pT  
                                    [in] TFileTransferStatus St  
                                    )
```

This event occurs when a file transfer status changes.

**Parameters:**

- [in] **pTransfer** contains the interface pointer for the File Transfer object.
- [in] **Status** contains the new file transfer status.

**See also:**

[ISkype::FileTransfers](#), [ISkype::ActiveFileTransfers](#)

```
HRESULT GroupDeleted ( [in] LONG GroupId )
```

This event is caused by a user deleting a custom contact group.

**Parameters:**

[in] **GroupId** contains the ID of the deleted group.

```
HRESULT GroupExpanded ( [in] IGroup * pGroup,  
                        [in] VARIANT_BOOL Expanded  
                      )
```

This event is caused by a user expanding or collapsing a group in the contacts tab.

**Parameters:**

[in] **pGroup** contains the interface pointer for the **Group** object.

[in] **Expanded** is a Boolean indicator. When the value is set to True, the group is expanded.

```
HRESULT GroupUsers ( [in] IGroup * pGroup,  
                    [in] IUserCollection * pUsers  
                  )
```

This event is caused by a change in a contact group.

**Parameters:**

[in] **pGroup** contains the interface pointer for the **Group** object.

[in] **pUsers** contains the interface pointer for the **UserCollection** object.

```
HRESULT GroupVisible ( [in] IGroup * pGroup,  
                      [in] VARIANT_BOOL Visible  
                    )
```

This event is caused by a user hiding/showing a group in the contacts tab.

**Parameters:**

[in] **pGroup** contains the interface pointer for the **Group** object.

[in] **Visible** is a Boolean indicator. It is set to True when the group is visible in the contacts tab.

**HRESULT MessageHistory ( [in] BSTR Username )**

This event is caused by a change in message history.

**Parameters:**

[in] **Username** contains the name of the user whose message history changed.

**HRESULT MessageStatus ( [in] IChatMessage \* pMessage,  
[in] TChatMessageStatus Status  
)**

This event is caused by a change in chat message status.

**Parameters:**

[in] **pMessage** receives the interface pointer for the **ChatMessage** object.

[in] **Status** contains the new message status.

**HRESULT Mute ( [in] VARIANT\_BOOL Mute )**

This event is caused by a change in mute status.

**Parameters:**

[in] **Mute** contains the new mute status value.

**HRESULT OnlineStatus ( [in] IUser \* pUser,**



[in] **TOnlineStatus Status**  
)

This event is caused by a change in the online status of a user.

**Parameters:**

[in] **pUser** receives the interface pointer for the **User** object.

[in] **Status** contains the new online status of the user.

**HRESULT PluginEventClicked ( [in] IPluginEvent \* pEvent )**

This event occurs when a user clicks on a plug-in event.

**Parameters:**

[in] **pEvent** contains the interface pointer for the **PluginEvent** object.

**See also:**

**IPluginEvent**

**HRESULT PluginMenuItemClicked ( [in] IPluginMenuItem \* pMenuItem,  
[in] IUserCollection \* pUsers,  
[in] TPluginContext PluginContext,  
[in] BSTR Context )**

This event occurs when a user clicks on a menu item.

**Parameters:**

[in] **pMenuItem** contains the interface pointer for the **PluginMenuItem** object.

[in] **pUsers** contains the interface pointer for the **UserCollection** object.

[in] **PluginContext** contains the plug-in context.  
[in] **ContextId** contains the context id.

**See also:**  
[IPluginMenuItem](#)

### **HRESULT Reply ( [in] ICommand \* pCommand )**

This event indicates the API response to the command object.

**Parameters:**

[in] **pCommand** contains the response to the **Command** object.

### **HRESULT SilentModeStatusChanged ( [in] VARIANT\_BOOL Silent )**

This event occurs when a silent mode is switched off.

**Parameters:**

[in] **Silent** contains the Skype client silent status.

### **HRESULT SmsMessageStatusChanged ( [in] ISmsMessage \* [in] TSmsMessageStatus )**

This event is caused by a change in the SMS message status.

**Parameters:**

[in] **pMessage** contains the interface pointer for the **SmsMessage** object.

[in] **Status** contains the new message status.

### **HRESULT SmsTargetStatusChanged ( [in] ISmsTarget \* pTarget )**

[in] TSmsTargetStatus Status  
)

This event is caused by a change in the SMS target status.

**Parameters:**

[in] **pTarget** contains the interface pointer for the **SmsTarget** object.

[in] **Status** contains the new target status.

**HRESULT UILanguageChanged ( [in] BSTR Code )**

This event occurs when user changes Skype client language.

**Parameters:**

[in] **Code** contains the new language code.

**HRESULT UserAuthorizationRequestReceived ( [in] IUser \* pUser )**

This event occurs when user sends you authorization request.

**Parameters:**

[in] **pUser** contains the the interface pointer for the User object.

**HRESULT UserMood ( [in] IUser \* pUser,  
[in] BSTR MoodText  
)**

This event is caused by a change in the mood text of the user.

**Parameters:**

[in] **pUser** contains the interface pointer for the **User** object.

[in] **MoodText** contains the new mood text.

### **HRESULT UserStatus ( [in] TUserStatus Status )**

This event is caused by a user status change.

**Parameters:**

[in] **Status** contains the new user status.

### **HRESULT VoicemailStatus ( [in] IVoicemail \* pMail, [in] TVoicemailStatus Status )**

This event is caused by a change in voicemail status.

**Parameters:**

[in] **pMail** contains the interface pointer for the **Voicemail** object.

[in] **Status** contains the new voicemail status.

### **HRESULT WallpaperChanged ( [in] BSTR Path )**

This event occurs when a wallpaper changes.

**Parameters:**

[in] **Path** to new wallpaper bitmap.

**See also:**

**IClient::Wallpaper**

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Application</a>		

## Application Class Reference

---

This class enables two applications sharing the same name to communicate over the **Skype** network. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class enables two applications sharing the same name to communicate over the [Skype](#) network.

**See also:**

[IApplication](#), [IApplicationStream](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 [Skype Limited](#). All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ApplicationStream</a>		

## ApplicationStream Class Reference

---

This class manages application streams. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages application streams.

The [ApplicationStream](#) class provides a reliable connection between two application endpoints.

**See also:**

[IApplicationStream](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ApplicationStreamCollection</a>		

# ApplicationStreamCollection Class Reference

---

This class manages application stream collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages application stream collections.

**See also:**

[IApplicationStreamCollection](#), [IApplicationStream](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Call</a>		

## Call Class Reference

---

This class manages calls and call properties. [More...](#)

```
import "Skype4COM.idl";
```

---



## Detailed Description

This class manages calls and call properties.

Use the **Call** class to manage call properties, answer incoming calls, change call status, and send DTMF tones.

### Example 1: Placing a call

```
'// Create a Skype object:  
Set oSkype = WScript.CreateObject("Skype4COM  
.Skype", "Skype_")  
  
'// Place a call using this Skype object:  
Set oCall = oSkype.PlaceCall("echo123")  
  
'// To place a call on hold:  
oCall.Hold  
  
'// To resume a call that was on hold:  
oCall.Resume  
  
'// To end a call:  
oCall.Finish  
  
'// Call status events:  
Public Sub Skype_CallStatus(ByRef oCall, ByVal Status)  
End Sub
```

### Example 2: Accessing call properties

```
'// To query a call object with a specified  
ID:  
Set oCall = oSkype.Call(oId)  
  
'// To query the Skype name of a remote caller:  
r:
```

```
Set oHandle = oCall.PartnerHandle

'// To query the display name of a remote caller:
Set oName = oCall.PartnerDisplayName
```

**See also:**

**[ICall](#), [ICallCollection](#), [\\_ISkypeEvents](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright © 2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">CallChannel</a>		

## CallChannel Class Reference

---

This class forms a logical connection between the [ApplicationStream](#) and [Call](#) objects. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class forms a logical connection between the [ApplicationStream](#) and [Call](#) objects.

The [CallChannel](#) is an abstract layer above the [Call](#) and [ApplicationStream](#) objects. The lifetime of the [CallChannel](#) (and the corresponding [ApplicationStream](#)) is determined by the lifetime of the active [Call](#).

**See also:**

[ICallChannel](#), [ICallChannelCollection](#), [ICallChannelManager](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)
- 

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">CallChannelCollection</a>		

# CallChannelCollection Class Reference

---

Container class for **CallChannel** objects. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Container class for [CallChannel](#) objects.

**See also:**

[ICallChannel](#), [ICallChannelCollection](#), [ICallChannelManager](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">CallChannelManager</a>		

# CallChannelManager Class Reference

---

This class manages **Call** data channels. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages **Call** data channels.

The **CallChannelManager** class listens for incoming **Skype Call** and attempts to create a data channel between the callers. The **ApplicationStream** is created when the **ICall::Status** is **TCallStatus::clsRinging**. The **ApplicationStream** is deleted when the call status changes to **TCallStatus::clsCancelled**, **TCallStatus::clsFailed**, **TCallStatus::clsFinished** or **TCallStatus::clsRefused**.

### Note:

To create a data channel it is necessary to have an instance of the **CallChannelManager** with the same name running on both endpoints of a call. It is possible to have multiple data channels running at the same time.

### See also:

**ICallChannelManager**, **\_ICallChannelManagerEvents**,  
**ICallChannel**

### Example:

```
'// To create a channel manager object:  
Set oMgr = WScript.CreateObject("Skype4COM.CallChannelManager", "ChannelManager_")  
  
'// Create the application context:  
oMgr.Create("TestApplicationName")  
  
'// Wait for the application context to be created:  
Do While Not oMgr.Created  
    WScript.Sleep(1000)  
Loop  
  
'// This event occurs when a new channel is c
```

```
reated or an existing channel is deleted:
Public Sub ChannelManager_Channels(ByRef aManager, ByRef aChannels)
End Sub

'// This event occurs when a message arrives
from a remote user
Public Sub ChannelManager_Message(ByRef aManager, ByRef aChannel, ByRef aMessage)
End Sub

'// This event occurs when an application context is created
Public Sub ChannelManager_Created()
End Sub
```

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">CallChannelMessage</a>		

## CallChannelMessage Class Reference

---

This class manages **CallChannel** messages. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages **CallChannel** messages.

**See also:**

**[ICallChannelMessage](#), [ICallChannelManager](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">CallCollection</a>		



# CallCollection Class Reference

---

This class manages **Call** collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages **Call** collections.

The **CallCollection** class is used to manage call collections. To query items in a collection, use **For Each .. Next** statements.

### Example:

```
'// To query call history:  
For Each oCall In oSkype.Calls  
Next  
  
'// To query the call history for a user:  
For Each oCall In oSkype.Calls("echo123")  
Next  
  
'// To query active calls:  
For Each oCall In oSkype.ActiveCalls  
Next  
  
'// To query missed calls:  
For Each oCall In oSkype.MissedCalls  
Next
```

### See also:

**ICall**, **ICallCollection**

---

The documentation for this class was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Chat</a>		

## Chat Class Reference

---

This class manages chats. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages chats.

The **Chat** class groups messages with the same chat name.

### Example:

```
'// To query a chat object with a specified
name:
Set oChat = oSkype.Chat(oName)

'// To query a chat topic:
Set oTopic = oChat.Topic

'// To query a chat timestamp:
Set oTime = oChat.Timestamp

'// To iterate through messages in a chat:
For Each oMsg In oChat.Messages
Next
```

### See also:

**[IChat](#)**, **[IChatCollection](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ChatCollection</a>		

# ChatCollection Class Reference

---

This class manages **Chat** collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages **Chat** collections.

This class manages collections of **Chat** objects. To query a chat collection, use **For Each .. Next** statements.

### Example:

```
'// To query chat history:  
For Each oChat In oSkype.Chats  
Next
```

### See also:

**[IChat](#)**, **[IChatCollection](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright  2006 [Skype Limited](#). All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ChatMember</a>		

## ChatMember Class Reference

---

This class manages chat member. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages chat member.

**See also:**

[IChatMember](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ChatMemberCollection</a>		

# ChatMemberCollection Class Reference

---

This class manages chat member collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages chat member collections.

**See also:**

[IChatMemberCollection](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ChatMessage</a>		

## ChatMessage Class Reference

---

This class manages chat messages. [More...](#)

```
import "Skype4COM.idl";
```

---



## Detailed Description

This class manages chat messages.

**See also:**

[IChatMessage](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ChatMessageCollection</a>		

## ChatMessageCollection Class Reference

---

This class manages chat message collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages chat message collections.

**See also:**

[IChatMessageCollection](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Client</a>		

## Client Class Reference

---

This class controls features related to the **Skype** user interface. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class controls features related to the [Skype](#) user interface.

**See also:**  
[IClient](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Command</a>		



## Command Class Reference

---

This class manages [Skype API](#) command objects. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages [Skype API](#) command objects.

**See also:**

[ICommand](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Conference</a>		

## Conference Class Reference

---

This class manages conference calls. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages conference calls.

The **Conference** class manages a call connected to multiple participants.

**See also:**

**[IConference](#), [IConferenceCollection](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ConferenceCollection</a>		

## ConferenceCollection Class Reference

---

This class manages conference collections. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages conference collections.

Use the [ConferenceCollection](#) class to query conference collections in call history. To query a collection class, use **For Each .. Next** statements.

### See also:

[IConference](#), [IConferenceCollection](#), [ISkype::Conferences](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Conversion</a>		

## Conversion Class Reference

---

This class converts text to constant values and converts constant values to text. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class converts text to constant values and converts constant values to text.

### Example:

```
'// To create an object:  
Set oSkype = CreateObject("Skype4COM.Skype")  
  
'// To convert an attachment status to text:  
oAttachStatus = oSkype.Convert.AttachmentSta  
tusToText(oSkype.AttachmentStatus)  
  
'// To change the online status of a user:  
oSkype.CurrentUserStatus = oSkype.Convert.Te  
xtToUserStatus("ONLINE")
```

### See also:

[IConversion](#), [ISkype::Convert](#)

### Deprecated:

The [SKYPE4COMLib::Conversion](#) class is deprecated, support for multilanguage status codes support will be dropped in the future.

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Group</a>		

## Group Class Reference

---

This class manages contact groups. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages contact groups.

**See also:**

[IGroup](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">GroupCollection</a>		

# GroupCollection Class Reference

---

This class manages collections of **Group** objects. [More...](#)

```
import "Skype4COM.idl";
```

---



## Detailed Description

This class manages collections of **Group** objects.

**See also:**

**[IGroupCollection](#), [IGroup](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IApplication Interface Reference

---

**Application** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Create** ()

This command creates the application context.

HRESULT **Delete** ()

This command deletes the application context.

HRESULT **Connect** ([in] BSTR Username,[in, defaultvalue(0)]  
VARIANT\_BOOL WaitConnected)

This command connects a user to the application context.

HRESULT **SendDatagram** ([in] BSTR Text,[in, defaultvalue(0)]  
**IApplicationStreamCollection** \*pStreams)

This command broadcasts a datagram to connected streams.

## Properties

	<b>BSTR</b>	<b>Name</b> [get] This command queries an application name.
<b>IApplicationStreamCollection</b>		<b>Streams</b> [get] This command queries application streams.
	<b>IUserCollection</b>	<b>ConnectableUsers</b> [get] This command queries users that can connect to the application.
	<b>IUserCollection</b>	<b>ConnectingUsers</b> [get] This command queries the users who are currently connecting to the application.
<b>IApplicationStreamCollection</b>		<b>SendingStreams</b> [get] This command queries application sending streams.
<b>IApplicationStreamCollection</b>		<b>ReceivedStreams</b> [get] This command queries application receiving streams.

---

## Detailed Description

**Application** class default interface.

**See also:**

**[IApplicationStream](#)**

---

## Member Function Documentation

**HRESULT Connect** ( [in] BSTR **Username**  
[in, defaultvalue(0)] VARIANT\_BOOL **WaitCon**  
)

This command connects a user to the application context.

**Parameters:**

- [in] **Username** contains the Skype name of the user to connect.
- [in] **WaitConnected** is an optional Boolean value to indicate whether to wait for the user to connect or not.

**HRESULT SendDatagram** ( [in] BSTR  
[in, defaultvalue(0)] IApplicationStream  
)

This command broadcasts a datagram to connected streams.

**Parameters:**

- [in] **Text** contains the text of the datagram.
- [in] **pStreams** is an optional parameter you can use to specify the collection of streams that will receive the datagram.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# IApplicationStream Interface Reference

---

**ApplicationStream** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Read</b> ([out, retval] BSTR *pText)	This command reads a text string from an application stream.
HRESULT	<b>Write</b> ([in] BSTR Text)	This command writes a text string to an application stream.
HRESULT	<b>SendDatagram</b> ([in] BSTR Text)	This command sends a datagram message. Delivery of the message is not guaranteed.
HRESULT	<b>Disconnect</b> ()	This command disconnects an application stream from an application context.

## Properties

---

BSTR **ApplicationName** [get]

This command queries an application name.

---

BSTR **Handle** [get]

This command queries the name of an application stream.

---

LONG **DataLength** [get]

This command queries the data length of an application stream.

---

BSTR **PartnerHandle** [get]

This command queries the Skype name of the sender of an application stream.

---

## Detailed Description

[ApplicationStream](#) class default interface.

**See also:**

[IApplication](#)

---

## Member Function Documentation

### HRESULT Read ( [out, retval] BSTR \* **pText** )

This command reads a text string from an application stream.

**Parameters:**

[out] **pText** receives the text string from the application stream.

### HRESULT SendDatagram ( [in] BSTR **Text** )

This command sends a datagram message. Delivery of the message is not guaranteed.

**Parameters:**

[in] **Text** contains the message to send.

**Note:**

To transmit binary data use the base64 encoding to convert your data to strings before passing those strings to **IApplication.SendDatagram** or **IApplicationStream.Write** methods.

**Example:**

```
//Declare the variables
var Skype, App;

//Create the Skype application object.
Skype = new ActiveXObject("Skype4COM.Skype")
;

// Connect events:
WScript.ConnectObject(Skype, "Skype_");
```

```

//Create the application object
var App = Skype.Application("TestBase64DataTransfer");

//Create the application context
App.Create();
WScript.Echo ("App created.");

//Connect remote user
if (WScript.Arguments.Count() > 0)
{
    App.Connect(WScript.Arguments.Item(0), true);
    WScript.Echo (WScript.Arguments.Item(0) + " connected.");
}
//Byte array
var byteArray = new Array(17, 34, 51, 68, 85, 102, 119, 136, 153, 170);

// Send datagram
if (App.Streams.Count)
{
    //Encoded byte array
    var encoded = encode64(byteArray);

    //Send encoded array to streams
    for (var i=0; i < App.Streams.Count; i++)
    {
        var stm = App.Streams.Item(i+1);
        for (var i=0; i < App.Streams.Count; i++)
    ){
        stm.SendDatagram(encoded);
    }
    }
    WScript.Echo(">Datagram [" + byteArray + " ] sent");
}

```

```
}  
//Sleep 10 minutes  
WScript.Sleep(600000);  
  
//Application datagram events  
function Skype_ApplicationDatagram(aApp, aStream, aText)  
{  
    WScript.Echo(">Datagram [" + decode64(aText) + "] received");  
}  
  
//Heres the encode function  
function encode64(inp)  
{  
    // Read more about base64 at http://en.wikipedia.org/wiki/Base64  
    ...  
}  
  
//Heres the decode function  
function decode64(inp)  
{  
    ...  
}
```

### **HRESULT Write ([in] BSTR Text)**

This command writes a text string to an application stream.

#### **Parameters:**

[in] **Text** contains the text string to write.

The documentation for this interface was generated from the following

file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IApplicationStreamCollection Interface Reference

---

**ApplicationStreamCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IApplicationStream** \*pItem)

This command adds a new item to an **ApplicationStreamCollection**.

HRESULT **Remove** ([in] long Index)

This command removes a stream from an **ApplicationStreamCollection**.

HRESULT **RemoveAll** (void)

This command removes all streams from an **ApplicationStreamCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of streams in an <b>ApplicationStreamCollection</b> .
<b>IApplicationStream</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a stream with a specified index in an <b>ApplicationStreamCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command queries an <b>ApplicationStreamCollection</b> enumerator object.

## Detailed Description

[ApplicationStreamCollection](#) class default interface.

**See also:**

[IApplicationStream](#)

---

## Member Function Documentation

### **HRESULT Add ( [in] IApplicationStream \* pStream )**

This command adds a new item to an **ApplicationStreamCollection**.

**Parameters:**

[in] **pStream** contains the **ApplicationStream** to add.

### **HRESULT Remove ( [in] long Index )**

This command removes a stream from an **ApplicationStreamCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries a stream with a specified index in an [ApplicationStreamCollection](#).

**Parameters:**

[in] **Index** contains the stream to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# ICall Interface Reference

---

**Call** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Hold</b> ()	This command places a call on hold.
HRESULT	<b>Resume</b> ()	This command resumes a call that was on hold.
HRESULT	<b>Finish</b> ()	This command ends a call.
HRESULT	<b>Answer</b> ()	This command answers a call.
HRESULT	<b>Join</b> ([in] LONG Id)	This command joins a call with another call to form a call conference.
HRESULT	<b>StartVideoSend</b> ()	This command starts video send.
HRESULT	<b>StopVideoSend</b> ()	This command stops video send.
HRESULT	<b>StartVideoReceive</b> ()	This command starts video receive.
HRESULT	<b>StopVideoReceive</b> ()	This command stops video receive.
HRESULT	<b>RedirectToVoicemail</b> ()	This command redirects an incoming call to voicemail.
HRESULT	<b>Forward</b> ()	This command forwards an incoming call.
HRESULT	<b>Transfer</b> ([in] BSTR Target)	Transfers call to contact or phone number.

## Properties

LONG	<b>Id</b> [get]	This command queries the call ID.
DATE	<b>Timestamp</b> [get]	This command queries the call timestamp.
BSTR	<b>PartnerHandle</b> [get]	This command queries the Skype name of a remote caller.
BSTR	<b>PartnerDisplayName</b> [get]	This command queries the name of the remote caller.
LONG	<b>ConferenceId</b> [get]	This command queries a call conference ID.
<b>TCallType</b>	<b>Type</b> [get]	This command queries a call type.
<b>TCallStatus</b>	<b>Status</b> [get, set]	This command queries/sets a call status.
<b>TCallFailureReason</b>	<b>FailureReason</b> [get]	This command queries a call failure reason.
BSTR	<b>Subject</b> [get]	This command queries the subject of a call.
BSTR	<b>PstnNumber</b> [get]	This command queries the PSTN number of a call.
LONG	<b>Duration</b> [get]	This command queries the duration of a call.
BSTR	<b>PstnStatus</b> [get]	This command queries the call PSTN status.
VARIANT_BOOL	<b>Seen</b> [get, set]	

		This command queries or sets the call to a status of seen.
BSTR	<b>DTMF</b> [set]	This command sends a DTMF tone.
<b>IParticipantCollection</b>	<b>Participants</b> [get]	This command queries the number of participants in a conference call not hosted by the user.
LONG	<b>VmDuration</b> [get]	This command returns the duration of a voicemail.
LONG	<b>VmAllowedDuration</b> [get]	This command returns the permitted duration of a voicemail.
<b>TCallVideoStatus</b>	<b>VideoStatus</b> [get]	This command queries call video status.
<b>TCallVideoSendStatus</b>	<b>VideoSendStatus</b> [get]	This command queries call video send status.
<b>TCallVideoSendStatus</b>	<b>VideoReceiveStatus</b> [get]	This command queries call video receive status.
LONG	<b>Rate</b> [get]	This command queries the call rate, expressed in cents.
BSTR	<b>RateCurrency</b> [get]	This command queries currency code for the call rate.
LONG	<b>RatePrecision</b> [get]	This command queries the call rate precision.
BSTR	<b>InputDevice</b> ([in] <b>TCallIoDeviceType</b> DeviceType) [get, set]	This command queries or sets the input sound device.

BSTR	<b>OutputDevice</b> ([in] TCalloDeviceType DeviceType) [get, set] This command queries or sets the output sound device.
BSTR	<b>CaptureMicDevice</b> ([in] TCalloDeviceType DeviceType) [get, set] This command queries or sets the mic capture device.
VARIANT_BOOL	<b>InputStatus</b> [get] This command queries the call voice input status.
BSTR	<b>ForwardedBy</b> [get] This command queries the the identity of a user who forwards a call.
VARIANT_BOOL	<b>CanTransfer</b> ([in] BSTR Target) [get] This command queries if a call can be transferred to contact or number.
TCallStatus	<b>TransferStatus</b> [get] Returns call transferring status.
VARIANT_BOOL	<b>TransferActive</b> [get] Returns true if the call has been transferred.
BSTR	<b>TransferredBy</b> [get] Returns the SkypeName of the user who transferred the call.
BSTR	<b>TransferredTo</b> [get] Returns the SkypeName of the user or phone number the call has been transferred to.
BSTR	<b>TargetIdentity</b> [get] Returns the target number for incoming SkypeIN calls.

## Detailed Description

**Call** class default interface.

---

## Member Function Documentation

### HRESULT Join ( [in] LONG **Id** )

This command joins a call with another call to form a call conference.

**Parameters:**

[in] **Id** contains the ID of the other call to join to the conference.

### HRESULT Transfer ( [in] BSTR **Target** )

Transfers call to contact or phone number.

**Parameters:**

[in] **Target** phone number(s) or Skype name(s) the call is being transferred to.

**Note:**

You can transfer an incoming call to a group by specifying a comma separated list of targets, first one of the group to answer will get the call.

**See also:**

[ICall::CanTransfer](#)

## Property Documentation

### HRESULT CanTransfer [get]

This command queries if a call can be transferred to contact or number.

**Parameters:**

[in] **Target** Skype name or phone number the call to be transferred to.

**See also:**

[ICall::Transfer](#)

### HRESULT CaptureMicDevice [get, set]

This command queries or sets the mic capture device.

**Parameters:**

[in] **DeviceType** contains the device type, TCallIoDeviceType::callIoDeviceTypeSoundcard cannot be used with this property.

**Note:**

This command functions for active calls only.

### HRESULT DTMF [set]

This command sends a DTMF tone.

**Parameters:**

[in] **newVal** contains the DTMF value to send. Permitted symbols are: {0..9,#,\*}.

**Note:**



This command functions for active calls only.

### **HRESULT InputDevice** [get, set]

This command queries or sets the input sound device.

#### **Parameters:**

[in] **DeviceType** contains the device type.

#### **Note:**

This command functions for active calls only.

### **HRESULT OutputDevice** [get, set]

This command queries or sets the output sound device.

#### **Parameters:**

[in] **DeviceType** contains the device type.

#### **Note:**

This command functions for active calls only.

### **HRESULT Seen** [get, set]

This command queries or sets the call to a status of seen.

#### **Parameters:**

[in] **newVal** contains a Boolean value to indicate if a call is seen or not.

#### **Note:**

You cannot alter the call seen status from seen to unseen.

### **HRESULT Status** [get, set]

This command queries/sets a call status.

**Deprecated:**

Set **ICall::Status** is deprecated, use the **ICall::Hold**, **ICall::Resume**, **ICall::Finish** or **ICall::Answer** instead.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ICallChannel Interface Reference

---

**CallChannel** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

---

HRESULT **SendMessage** (BSTR Text)

This command sends a **CallChannel** message.

## Properties

<b>TCallChannelType</b>	<b>Type</b> [get, set] This command queries/sets the <b>CallChannel</b> type.
<b>IApplicationStream</b>	<b>Stream</b> [get] This command returns the channel <b>ApplicationStream</b> object.
<b>ICallChannelManager</b>	<b>Manager</b> [get] This command returns the related <b>CallChannelManager</b> object.
<b>ICall</b>	<b>Call</b> [get] This command returns the related <b>Call</b> object.

---

## Detailed Description

**CallChannel** class default interface.

---

## Member Function Documentation

### HRESULT SendMessage ( BSTR **Text** )

This command sends a **CallChannel** message.

**Parameters:**

[in] **Text** contains the text of the message to send.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ICallChannelCollection Interface Reference

---

**CallChannelCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **ICallChannel** \*pChannel)  
This command adds a new call channel to a **CallChannelCollection**.

HRESULT **Remove** ([in] long Index)  
This command removes an item from a **CallChannelCollection**.

HRESULT **RemoveAll** (void)  
This command removes all items from a **CallChannelCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of call channels in a <b>CallChannelCollection</b> .
<b>ICallChannel</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a <b>CallChannelCollection</b> item with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries a <b>CallChannelCollection</b> enumerator object.

---

## Detailed Description

**CallChannelCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] ICallChannel \* pItem )**

This command adds a new call channel to a **CallChannelCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer for the **CallChannel** object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a **CallChannelCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries a **CallChannelCollection** item with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# ICallChannelManager Interface Reference

---

**CallChannelManager** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Connect</b> ([in] <b>ISkype</b> *pSkype) This command binds the <b>CallChannelManager</b> object to the <b>Skype</b> object.
HRESULT	<b>Disconnect</b> () This command disconnects an existing binding between the <b>CallChannelManager</b> and the <b>Skype</b> object.
HRESULT	<b>CreateApplication</b> ([in, defaultValue("")] BSTR ApplicationName) This command creates the <b>Application</b> context.

## Properties

<b>ICallChannelCollection</b>	<b>Channels</b> [get] This command queries call data channels.
<b>TCallChannelType</b>	<b>ChannelType</b> [get, set] This command queries/sets the default call channel type.
<b>BSTR</b>	<b>Name</b> [get, set] This command queries/sets the <b>Application</b> context name.
<b>VARIANT_BOOL</b>	<b>Created</b> [get] This command returns the application context creation status.

---

## Detailed Description

**CallChannelManager** class default interface.

---

## Member Function Documentation

### HRESULT Connect ( [in] ISkype \* pSkype )

This command binds the **CallChannelManager** object to the **Skype** object.

**Parameters:**

[in] **pSkype** contains the interface pointer for the **Skype** object.

### HRESULT CreateApplication ( [in, defaultvalue("")] BSTR ApplicationName )

This command creates the **Application** context.

**Parameters:**

[in] **ApplicationName** contains an optional application context name.

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# ICallChannelMessage Interface Reference

---

**CallChannelMessage** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

---

BSTR **Text** [get, set]

This command queries/sets message text.

---



## Detailed Description

**CallChannelMessage** class default interface.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ICallCollection Interface Reference

---

**CallCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **ICall** \*pCall)

This command adds a new call to a **CallCollection**.

HRESULT **Remove** ([in] long Index)

This command removes a call from a **CallCollection**.

HRESULT **RemoveAll** (void)

This command removes all calls from a **CallCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of calls in a <b>CallCollection</b> .
ICall	<b>Item</b> ([in] LONG Index) [get]	This command queries a call with a specified index in a <b>CallCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command queries the enumerator object of a <b>CallCollection</b> .

---

## Detailed Description

**CallCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] ICall \* pCall )**

This command adds a new call to a **CallCollection**.

**Parameters:**

[in] **pCall** receives the interface pointer for the **Call** object.

### **HRESULT Remove ( [in] long Index )**

This command removes a call from a **CallCollection**.

**Parameters:**

[in] **Index** contains the call to be removed.

## Property Documentation

### HRESULT Item [get]

This command queries a call with a specified index in a **CallCollection**.

**Parameters:**

[in] **Index** contains the call to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IChat Interface Reference

---

**Chat** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>OpenWindow</b> ()	This command opens a chat window.
HRESULT	<b>SendMessage</b> ([in] BSTR MessageText,[out, retval] IChatMessage **pVal)	This command sends a chat message.
HRESULT	<b>Leave</b> ()	This command leaves a chat.
HRESULT	<b>AddMembers</b> ([in] IUserCollection *pMembers)	This command adds new members to a chat.
HRESULT	<b>Bookmark</b> ()	This command bookmarks a chat.
HRESULT	<b>Unbookmark</b> ()	This command removes the bookmark for a chat.
HRESULT	<b>SetPassword</b> ([in] BSTR Password,[in, defaultvalue("")] BSTR Hint)	Sets chat password.
HRESULT	<b>Join</b> (void)	Joins the chat.
HRESULT	<b>Kick</b> ([in] BSTR Handle)	Kicks member from chat.
HRESULT	<b>KickBan</b> ([in] BSTR Handle)	Kicks and ban member from chat.
HRESULT	<b>Disband</b> (void)	Ends chat.
HRESULT	<b>EnterPassword</b> ([in] BSTR Password)	Enters chat password.
HRESULT	<b>ClearRecentMessages</b> (void)	Clears recent chat messages.
HRESULT	<b>AcceptAdd</b> (void)	Accepts shared group add.

## Properties

BSTR	<b>Name</b> [get]	This command queries the chat name.
<b>IChatMessageCollection</b>	<b>Messages</b> [get]	This command queries chat messages.
DATE	<b>Timestamp</b> [get]	This command queries the chat timestamp.
<b>IUser</b>	<b>Adder</b> [get]	This command queries who added a user to a chat.
<b>TChatStatus</b>	<b>Status</b> [get]	This command queries a chat status.
<b>IUserCollection</b>	<b>Posters</b> [get]	This command queries users who have posted messages to a chat.
<b>IUserCollection</b>	<b>Members</b> [get]	This command queries the members of a chat.
BSTR	<b>Topic</b> [get, set]	This command queries/sets a chat topic.
<b>IUserCollection</b>	<b>ActiveMembers</b> [get]	This command queries the active members of a chat.
BSTR	<b>FriendlyName</b> [get]	This command queries the "friendly" name of a chat.
<b>IChatMessageCollection</b>	<b>RecentMessages</b> [get]	This command queries the most recent chat messages.
VARIANT_BOOL	<b>Bookmarked</b> [get]	This command queries if a chat is bookmarked.

	BSTR	<b>TopicXML</b> [get, set] Returns/sets chat topic in XML format.
<b>IChatMemberCollection</b>		<b>MemberObjects</b> [get] Returns chat member objects collection.
	BSTR	<b>Blob</b> [get] Returns chat blob.
	LONG	<b>Options</b> [get, set] Returns/sets chat options.
	BSTR	<b>PasswordHint</b> [get] Returns chat password hint.
	BSTR	<b>GuideLines</b> [get, set] Returns/sets chat guidelines.
	BSTR	<b>Description</b> [get, set] Returns/sets chat description.
	BSTR	<b>DialogPartner</b> [get] Returns chat dialog partner.
	DATE	<b>ActivityTimestamp</b> [get] Returns chat activity timestamp.
<b>TChatMemberRole</b>		<b>MyRole</b> [get] Returns my chat role.
<b>IUserCollection</b>		<b>Applicants</b> [get] Returns chat applicants collection.
	BSTR	<b>AlertString</b> [set] Sets chat alert string.
<b>TChatType</b>		<b>Type</b> [get] Returns chat type.
<b>TChatMyStatus</b>		<b>MyStatus</b> [get] Returns my chat status.

---

## Detailed Description

**Chat** class default interface.

---

## Member Function Documentation

**HRESULT AddMembers ( [in] IUserCollection \* pMembers )**

This command adds new members to a chat.

**Parameters:**

[in] **pMembers** receives the interface pointer for the **IUserCollection** object.

**HRESULT EnterPassword ( [in] BSTR Password )**

Enters chat password.

**Parameters:**

[in] **Password** contains password string.

**HRESULT SendMessage ( [in] BSTR MessageText, [out, retval] IChatMessage \*\* pVal )**

This command sends a chat message.

**Parameters:**

[in] **MessageText** contains the message text.

[out] **pVal** receives the interface pointer for the **ChatMessage** object.

**HRESULT SetPassword ( [in] BSTR Password, [in, defaultvalue("")] BSTR Hint )**

Sets chat password.

**Parameters:**

[in] **Password** contains password.

[in] **Hint** contains password hint.

**See also:**

[IChat::PasswordHint](#)

---



## Property Documentation

### HRESULT Options [get, set]

Returns/sets chat options.

**See also:**

[TChatOptions](#)

### HRESULT Status [get]

This command queries a chat status.

**Deprecated:**

[ICall::Status](#) is deprecated, use the [IChat::Type](#) and [IChat::MyStatus](#) instead.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IChatCollection Interface Reference

---

**ChatCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IChat** \*pItem)

This command adds a new chat to a **ChatCollection**.

HRESULT **Remove** ([in] long Index)

This command removes an item from a **ChatCollection**.

HRESULT **RemoveAll** (void)

This command removes all items from a **ChatCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of chats in a <b>ChatCollection</b> .
<b>IChat</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a chat collection item with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries a <b>ChatCollection</b> enumerator object.

---

## Detailed Description

**ChatCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] IChat \* pItem )**

This command adds a new chat to a **ChatCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer for the **Chat** object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a **ChatCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

## Property Documentation

### HRESULT Item [get]

This command queries a chat collection item with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# IChatMember Interface Reference

---

**ChatMember** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

BSTR	<b>Id</b> [get] Returns member id.
BSTR	<b>Handle</b> [get] Returns member Skype name.
<b>TChatMemberRole</b>	<b>Role</b> [get, set] Sets/returns member role.
VARIANT_BOOL	<b>CanSetRoleTo</b> ([in] <b>TChatMemberRole</b> Role) [get] Checks if the new role can be applied to member.
VARIANT_BOOL	<b>IsActive</b> [get] Returns member activity status.
<b>IChat</b>	<b>Chat</b> [get] Returns related chat object.

## Detailed Description

**ChatMember** class default interface.

---

## Property Documentation

### HRESULT CanSetRoleTo [get]

Checks if the new role can be applied to member.

**Parameters:**

[in] **Role** contains the new role.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Properties](#)

# IChatMemberCollection Interface Reference

---

**ChatMemberCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of chat members in a <b>ChatMemberCollection</b> .
<b>IChatMember</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item in a <b>ChatMemberCollection</b> with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries an enumerator object for a <b>ChatMemberCollection</b> .



## Detailed Description

**ChatMemberCollection** class default interface.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item in a **ChatMemberCollection** with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# IChatMessage Interface Reference

---

**ChatMessage** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

	LONG	<b>Id</b> [get]	This command queries a message ID.
	DATE	<b>Timestamp</b> [get]	This command queries a message timestamp.
	BSTR	<b>FromHandle</b> [get]	This command queries the Skype name of a message sender.
	BSTR	<b>FromDisplayName</b> [get]	This command queries the display name of a sender.
<b>TChatMessageType</b>		<b>Type</b> [get]	This command queries a message type.
<b>TChatMessageStatus</b>		<b>Status</b> [get]	This command queries a message status.
<b>TChatLeaveReason</b>		<b>LeaveReason</b> [get]	This command queries the reason for leaving a chat.
	BSTR	<b>Body</b> [get, set]	Returns/sets chat message body.
	BSTR	<b>ChatName</b> [get]	This command queries the name of a chat that includes the message.
<b>IUserCollection</b>		<b>Users</b> [get]	This command queries which users are added to chat.
VARIANT_BOOL		<b>Seen</b> [set]	This command marks missed a chat message as seen.
<b>IChat</b>		<b>Chat</b> [get]	This command queries a message chat.
<b>IUser</b>		<b>Sender</b> [get]	

This command queries the sender of a chat message.

BSTR **EditedBy** [get]  
Returns chat message editor Skype name.

DATE **EditedTimestamp** [get]  
Returns chat message editing timestamp.

**TChatMemberRole** **Role** [get]  
Returns chat member role changed by this message.

LONG **Options** [get]  
Returns chat options changed by this message.

VARIANT\_BOOL **IsEditable** [get]  
Returns chat message editable status.

---

## Detailed Description

**ChatMessage** class default interface.

---

## Property Documentation

### HRESULT LeaveReason [get]

This command queries the reason for leaving a chat.

**Note:**

Applies to messages with TChatMessageType::cmeLeft type.

### HRESULT Options [get]

Returns chat options changed by this message.

**Note:**

Applies to messages with TChatMessageType::cmeSetOptions type.

### HRESULT Role [get]

Returns chat member role changed by this message.

**Note:**

Applies to messages with TChatMessageType::cmeSetRole type.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IChatMessageCollection Interface Reference

---

**ChatMessageCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IChatMessage** \*pItem)

This command adds a new item to a **ChatMessageCollection**.

HRESULT **Remove** ([in] long Index)

This command removes an item from a **ChatMessageCollection**.

HRESULT **RemoveAll** (void)

This command removes all items from a **ChatMessageCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of chat messages in a <b>ChatMessageCollection</b> .
<b>IChatMessage</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item in a <b>ChatMessageCollection</b> with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries an enumerator object for a <b>ChatMessageCollection</b> .

## Detailed Description

**ChatMessageCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] IChatMessage \* pItem )**

This command adds a new item to a **ChatMessageCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer for the **ChatMessage** object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a **ChatMessageCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item in a [ChatMessageCollection](#) with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# IClient Interface Reference

---

**Client** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

[List of all members.](#)

## Public Member Functions

HRESULT	<b>Start</b> ([in, defaultvalue(0)] VARIANT_BOOL Minimized,[in, defaultvalue(0)] VARIANT_BOOL Nosplash) This command starts the Skype client.
HRESULT	<b>Minimize</b> () This command minimizes (hides) the Skype client.
HRESULT	<b>Shutdown</b> () This command closes the Skype client.
HRESULT	<b>OpenProfileDialog</b> () This command opens the profile window for the current user.
HRESULT	<b>OpenUserInfoDialog</b> ([in] BSTR Username) This command opens the information window for the current user.
HRESULT	<b>OpenConferenceDialog</b> () This command opens the create conference window.
HRESULT	<b>OpenSearchDialog</b> () This command opens the search contacts window.
HRESULT	<b>OpenOptionsDialog</b> ([in] BSTR Page) This command opens the options window.
HRESULT	<b>OpenCallHistoryTab</b> () This command brings the call history tab into focus.
HRESULT	<b>OpenContactsTab</b> () This command brings the contacts tab into focus.
HRESULT	<b>OpenDialpadTab</b> () This command brings the dialpad tab into focus.
HRESULT	<b>OpenSendContactsDialog</b> ([in, defaultvalue("")] BSTR Username) This command opens the send contacts window.
HRESULT	<b>OpenBlockedUsersDialog</b> () This command opens the blocked users window.
HRESULT	<b>OpenImportContactsWizard</b> () This command opens the import contacts window.

HRESULT	<b>OpenGettingStartedWizard</b> () This command opens the getting started wizard.
HRESULT	<b>OpenAuthorizationDialog</b> ([in] BSTR Username) This command opens the user authorization window.
HRESULT	<b>OpenDialog</b> ([in] BSTR Name,[in, defaultvalue("")] BSTR Param1,[in, defaultvalue("")] BSTR Param2) This command opens the user interface window.
HRESULT	<b>OpenVideoTestDialog</b> () This command opens the video test window.
HRESULT	<b>OpenAddContactDialog</b> ([in, defaultvalue("")] BSTR Username) This command opens the "Add a Contact" window.
HRESULT	<b>OpenMessageDialog</b> ([in] BSTR Username,[in, defaultvalue("")] BSTR Text) This command opens the "Send an IM Message" window.
HRESULT	<b>OpenFileTransferDialog</b> ([in] BSTR User,[in] BSTR Folder) This command opens the file transfer window.
HRESULT	<b>Focus</b> () This command brings the client window into focus.
HRESULT	<b>ButtonPressed</b> ([in] BSTR Key) This command sends a button pressed notification event.
HRESULT	<b>ButtonReleased</b> ([in] BSTR Key) This command sends a button released notification event.
HRESULT	<b>OpenSmsDialog</b> ([in] BSTR SmsId) This command opens SMS window.
HRESULT	<b>CreateEvent</b> ([in] BSTR EventId,[in] BSTR Caption,[in] BSTR Hint,[out, retval] <b>IPluginEvent</b> **pEvent) This command creates a plug-in event.
HRESULT	<b>CreateMenuItem</b> ([in] BSTR MenuItemId,[in] <b>TPluginContext</b> PluginContext,[in] BSTR CaptionText,[in, defaultvalue("")] BSTR HintText,[in, defaultvalue("")] BSTR IconPath,[in, defaultvalue(-1)] VARIANT_BOOL Enabled,[in, defaultvalue(pluginContactTypeAll)]

**TPluginContactType** ContactType,[in, defaultvalue(0)]  
VARIANT\_BOOL MultipleContacts,[out, retval]  
**IPluginMenuItem** \*\*pMenuItem)

This command creates a menu item for a plug-in.

---

HRESULT **OpenLiveTab** ()

This command brings the Live tab into focus.

## Properties

---

VARIANT\_BOOL **IsRunning** [get]  
This command queries the running status of the Skype client.

BSTR **Wallpaper** [get, set]  
This command queries/sets path to wallpaper bitmap.

---

# Detailed Description

**Client** class default interface.

---

## Member Function Documentation

### HRESULT ButtonPressed ( [in] BSTR **Key** )

This command sends a button pressed notification event.

**Parameters:**

[in] **Key** is a button key (0-9, A-Z, #, \*, UP, DOWN, YES, NO, SKYPE, PAGEUP, PAGEDOWN).

### HRESULT ButtonReleased ( [in] BSTR **Key** )

This command sends a button released notification event.

**Parameters:**

[in] **Key** is button key (0-9, A-Z, #, \*, UP, DOWN, YES, NO, SKYPE, PAGEUP, PAGEDOWN).

```
HRESULT CreateEvent ( [in] BSTR                               EventId,  
                    [in] BSTR                               Caption,  
                    [in] BSTR                               Hint,  
                    [out,retval] IPluginEvent ** pEvent  
                    )
```

This command creates a plug-in event.

**Parameters:**

[in] **EventId** is a unique identifier for the plug-in event.

[in] **Caption** event contains caption text.

[in] **Hint** event contains hint text.

[out] **pEvent** receives the interface pointer for the **PluginEvent** object.

```

HRESULT CreateMenuItem ( [in] BSTR
                        [in] TPluginContext
                        [in] BSTR
                        [in,defaultvalue("")] BSTR
                        [in, defaultvalue("")] BSTR
                        [in, defaultvalue(-1)] VARIANT_BOOL
                        [in, defaultvalue(pluginContactTypeAll)
                        [in, defaultvalue(0)] VARIANT_BOOL
                        [out,retval] IPluginMenuItem **
                        )

```

This command creates a menu item for a plug-in.

**Parameters:**

[in] <b>MenuItemId</b>	is a unique identifier for the menu item
[in] <b>PluginContext</b>	contains the plug-in context.
[in] <b>CaptionText</b>	contains the caption text for the menu item.
[in] <b>HintText</b>	contains the hint text for the menu item (optional).
[in] <b>IconPath</b>	is the path to an icon file (optional).
[in] <b>Enabled</b>	defines the initial state of the menu item (optional - default enabled).
[in] <b>ContactType</b>	is the contact type and is relevant only the TPluginContext::pluginContextContact context (optional - default all contacts).
[in] <b>MultipleContacts</b>	defines whether multiple contacts are allowed for this menu item (optional - default does not allow multiple contact).
[out] <b>pMenuItem</b>	receives the interface pointer for the <b>PluginMenuItem</b> object.

**See also:**

**TPluginContext**



## HRESULT OpenAddContactDialog ( [in, defaultvalue("")] BSTR Username )

This command opens the "Add a Contact" window.

### Parameters:

[in] **Username** is an optional field for the Skype name of the contact.

## HRESULT OpenAuthorizationDialog ( [in] BSTR Username )

This command opens the user authorization window.

### Parameters:

[in] **Username** contains the Skype name of the user.

## HRESULT OpenDialog ( [in] BSTR Name, [in, defaultvalue("")] BSTR Param1, [in, defaultvalue("")] BSTR Param2 )

This command opens the user interface window.

### Parameters:

[in] **Name** The Name field contains the name of the window.

[in] **Param1** is an optional parameter.

[in] **Param2** is an optional parameter.

## HRESULT OpenFileTransferDialog ( [in] BSTR Username, [in] BSTR Folder )

This command opens the file transfer window.

**Parameters:**

- [in] **Username** contains the Skype name of the user
- [in] **Folder** contains the folder name to transfer.

**HRESULT OpenMessageDialog ( [in] BSTR Username [in, defaultvalue("")] BSTR Text )**

This command opens the "Send an IM Message" window.

**Parameters:**

- [in] **Username** contains the message target.
- [in] **Text** is an optional field which can contain message text.

**HRESULT OpenOptionsDialog ( [in] BSTR Page )**

This command opens the options window.

**Parameters:**

- [in] **Page** contains the page to open.

**HRESULT OpenSendContactsDialog ( [in, defaultvalue("")] BSTR Username )**

This command opens the send contacts window.

**Parameters:**

- [in] **Username** is an optional field for the Skype name of the user.

**HRESULT OpenSmsDialog ( [in] BSTR SmsId )**

This command opens SMS window.

**Parameters:**

[in] **SmsId** is SMS message id.

**HRESULT OpenUserInfoDialog ( [in] BSTR **Username** )**

This command opens the information window for the current user.

**Parameters:**

[in] **Username** contains the Skype name of the user

**HRESULT Start ( [in, defaultvalue(0)] VARIANT\_BOOL **Minimized**,  
[in, defaultvalue(0)] VARIANT\_BOOL **Nosplash**  
)**

This command starts the Skype client.

**Parameters:**

[in] **Minimized** if True, Skype is minimized in the system tray.

[in] **Nosplash** if True, Skype does not display a splash screen on start up.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			
<a href="#">Class List</a>	<a href="#">Class Members</a>			

Properties

# ICommand Interface Reference

---

**Command** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

LONG	<b>Id</b> [get, set]	This command queries/sets a command ID.
LONG	<b>Timeout</b> [get, set]	This command queries/sets the wait timeout in milliseconds.
VARIANT_BOOL	<b>Blocking</b> [get, set]	This command queries/sets if blocking is set.
BSTR	<b>Command</b> [get, set]	This command queries/sets the command text.
BSTR	<b>Reply</b> [get, set]	This command queries/sets the reply text.
BSTR	<b>Expected</b> [get, set]	This command queries/sets the expected reply text.

## Detailed Description

**Command** class default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# IConference Interface Reference

---

**Conference** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Hold** ()

This command places all calls in a conference on hold and all active calls have the status TCallStatus::clsLocalHold.

HRESULT **Resume** ()

This command resumes a conference that was placed on hold and all active calls have the status TCallStatus::clsInProgress.

HRESULT **Finish** ()

This command finishes a conference and all active calls have the status TCallStatus::clsFinished.

## Properties

LONG **Id** [get]

This command queries the ID of a conference.

**ICallCollection** **Calls** [get]

This command queries calls with the same conference ID.

**ICallCollection** **ActiveCalls** [get]

This command queries active calls with the same conference ID.

## Detailed Description

**Conference** class default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IConferenceCollection Interface Reference

---

**ConferenceCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IConference** \*pItem)  
This command adds a new conference to a **ConferenceCollection**.

HRESULT **Remove** ([in] long Index)  
This command removes an item from a **ConferenceCollection**.

HRESULT **RemoveAll** (void)  
This command removes all items from a **ConferenceCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of conferences in a <b>ConferenceCollection</b> .
<b>IConference</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a <b>ConferenceCollection</b> item with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries the enumerator object of a <b>ConferenceCollection</b> .



## Detailed Description

**ConferenceCollection** class default interface.

---

## Member Function Documentation

### HRESULT Add ( [in] IConference \* pConference )

This command adds a new conference to a **ConferenceCollection**.

**Parameters:**

[in] **pConference** receives the interface pointer for the **Conference** object.

### HRESULT Remove ( [in] long Index )

This command removes an item from a **ConferenceCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries a **ConferenceCollection** item with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IConversion Interface Reference

---

**Conversion** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>OnlineStatusToText</b> ([in] <b>TOnlineStatus</b> status,[out, retval] BSTR *pVal) This command converts a user's online status from status code to text.
HRESULT	<b>TextToOnlineStatus</b> ([in] BSTR Text,[out, retval] <b>TOnlineStatus</b> *pVal) This command converts text to online status code.
HRESULT	<b>BuddyStatusToText</b> ([in] <b>TBuddyStatus</b> status,[out, retval] BSTR *pVal) This command converts contact status to text.
HRESULT	<b>TextToBuddyStatus</b> ([in] BSTR Text,[out, retval] <b>TBuddyStatus</b> *pVal) This command converts contact status text to contact status code.
HRESULT	<b>CallStatusToText</b> ([in] <b>TCallStatus</b> Status,[out, retval] BSTR *pVal) This command converts call status to text.
HRESULT	<b>TextToCallStatus</b> ([in] BSTR Text,[out, retval] <b>TCallStatus</b> *pVal) This command converts call status text to call status code.
HRESULT	<b>CallTypeToText</b> ([in] <b>TCallType</b> CallType,[out, retval] BSTR *pVal) This command converts the call type to text.
HRESULT	<b>TextToCallType</b> ([in] BSTR Text,[out, retval] <b>TCallType</b> *pVal) This command converts call type text to call type code.
HRESULT	<b>UserSexToText</b> ([in] <b>TUserSex</b> Sex,[out, retval] BSTR *pVal) This command converts user sex value to text.
HRESULT	<b>TextToUserSex</b> ([in] BSTR Text,[out, retval] <b>TUserSex</b> *pVal)

This command converts the user sex text string to code.

HRESULT	<b>ConnectionStatusToText</b> ([in] <b>TConnectionStatus</b> Status,[out, retval] BSTR *pVal) This command converts the connection status to a text string.
HRESULT	<b>TextToConnectionStatus</b> ([in] BSTR Text,[out, retval] <b>TConnectionStatus</b> *pVal) This command converts a text string of connection status to a code value.
HRESULT	<b>UserStatusToText</b> ([in] <b>TUserStatus</b> Status,[out, retval] BSTR *pVal) This command converts the user status value to a text string.
HRESULT	<b>TextToUserStatus</b> ([in] BSTR Text,[out, retval] <b>TUserStatus</b> *pVal) This command converts a string of user status to a code value.
HRESULT	<b>CallFailureReasonToText</b> ([in] <b>TCallFailureReason</b> reason,[out, retval] BSTR *pVal) This command converts a failure reason to a text string.
HRESULT	<b>AttachmentStatusToText</b> ([in] <b>TAttachmentStatus</b> status,[out, retval] BSTR *pVal) This command converts an attachment status code to a text string.
HRESULT	<b>ChatLeaveReasonToText</b> ([in] <b>TChatLeaveReason</b> reason,[out, retval] BSTR *pVal) This command converts a leave reason to a text string.
HRESULT	<b>ChatStatusToText</b> ([in] <b>TChatStatus</b> status,[out, retval] BSTR *pVal) This command converts a chat status code to a text string.
HRESULT	<b>VoicemailTypeToText</b> ([in] <b>TVoicemailType</b> type,[out, retval] BSTR *pVal) This command converts a voicemail type to a text string.
HRESULT	<b>VoicemailStatusToText</b> ([in] <b>TVoicemailStatus</b> status,[out, retval] BSTR *pVal)

This command converts a voicemail status to a text string.

HRESULT **TextToVoicemailStatus** ([in] BSTR Text,[out, retval] **TVoicemailStatus** \*pVal)  
This command converts a voicemail status text string to code.

HRESULT **VoicemailFailureReasonToText** ([in] **TVoicemailFailureReason** code,[out, retval] BSTR \*pVal)  
This command converts a voicemail failure reason to a text string.

HRESULT **ChatMessageStatusToText** ([in] **TChatMessageStatus** Status,[out, retval] BSTR \*pVal)  
This command converts message status code to text.

HRESULT **TextToChatMessageStatus** ([in] BSTR Text,[out, retval] **TChatMessageStatus** \*pVal)  
This command converts a message status from text to code.

HRESULT **ChatMessageTypeToText** ([in] **TChatMessageType** Type,[out, retval] BSTR \*pVal)  
This command converts the message type code to text.

HRESULT **TextToChatMessageType** ([in] BSTR Text,[out, retval] **TChatMessageType** \*pVal)  
This command converts a message type from text string to code.

HRESULT **TextToAttachmentStatus** ([in] BSTR Text,[out, retval] **TAttachmentStatus** \*pVal)  
This command converts an attachment status text string to code.

HRESULT **GroupTypeToText** ([in] **TGroupType** Type,[out, retval] BSTR \*pVal)  
This command converts a group type to a text string.

HRESULT **TextToGroupType** ([in] BSTR Text,[out, retval] **TGroupType** \*pVal)  
This command converts a text group type to a code value.



HRESULT **CallVideoStatusToText** ([in] **TCallVideoStatus** Status, [out, retval] BSTR \*pVal)  
This command converts call video status to a text string.

HRESULT **CallVideoSendStatusToText** ([in] **TCallVideoSendStatus** Status,[out, retval] BSTR \*pVal)  
This command converts call video send status to a text string.

HRESULT **SmsMessageStatusToText** ([in] **TSmsMessageStatus** status,[out, retval] BSTR \*pVal)  
This command converts SMS message status to a text string.

HRESULT **SmsMessageTypeToText** ([in] **TSmsMessageType** type, [out, retval] BSTR \*pVal)  
This command converts SMS message type to a text string.

HRESULT **SmsTargetStatusToText** ([in] **TSmsTargetStatus** status, [out, retval] BSTR \*pVal)  
This command converts SMS target status to a text string.

## Properties

---

BSTR **Language** [get, set]

This command queries/sets the language to use for a text conversion.

---

## Detailed Description

**Conversion** class default interface.

This interface converts Skype4COM constant values to text and converts text back to constant values.

**See also:**

Refer to the Skype4COM Dev Notes for more information about conversion.

---

## Member Function Documentation

```
HRESULT AttachmentStatusToText ( [in] TAttachmentStatus status  
                                [out, retval] BSTR * pVal  
                                )
```

This command converts an attachment status code to a text string.

**Parameters:**

- [in] **status** contains the attachment status code.
- [out] **pVal** receives the converted attachment status as a text string.

```
HRESULT BuddyStatusToText ( [in] TBuddyStatus status,  
                             [out, retval] BSTR * pVal  
                             )
```

This command converts contact status to text.

**Parameters:**

- [in] **status** is the contact status value.
- [out] **pVal** receives the contact status value as a text string.

```
HRESULT CallFailureReasonToText ( [in] TCallFailureReason reason  
                                  [out, retval] BSTR * pVal  
                                  )
```

This command converts a failure reason to a text string.

**Parameters:**

- [in] **reason** contains a failure reason code.
- [out] **pVal** receives the failure reason as a text string.

```
HRESULT CallStatusToText ( [in] TCallStatus    status,  
                           [out, retval] BSTR * pVal  
                           )
```

This command converts call status to text.

**Parameters:**

[in] **status** contains the call status value.

[out] **pVal** receives the call status as a text string.

```
HRESULT CallTypeToText ( [in] TCallType    CallType,  
                          [out, retval] BSTR * pVal  
                          )
```

This command converts the call type to text.

**Parameters:**

[in] **CallType** contains the call type value.

[out] **pVal** receives the call type value as a text string .

```
HRESULT CallVideoSendStatusToText ( [in] TCallVideoSendStatus  
                                     [out, retval] BSTR *  
                                     )
```

This command converts call video send status to a text string.

**Parameters:**

[in] **Status** contains the video send status code.

[out] **pVal** receives the video send status as a text string.

```
HRESULT CallVideoStatusToText ( [in] TCallVideoStatus Status,
```

```
[out, retval] BSTR * pVal  
)
```

This command converts call video status to a text string.

**Parameters:**

[in] **Status** contains the video status code.  
[out] **pVal** receives the converted code as a text string.

```
HRESULT ChatLeaveReasonToText ( [in] TChatLeaveReason reason  
[out, retval] BSTR * pVal  
)
```

This command converts a leave reason to a text string.

**Parameters:**

[in] **reason** contains a leave reason code.  
[out] **pVal** receives a leave reason as a text string.

```
HRESULT ChatMessageStatusToText ( [in] TChatMessageStatus Status  
[out, retval] BSTR * pVal  
)
```

This command converts message status code to text.

**Parameters:**

[in] **Status** contains the message status value.  
[out] **pVal** receives the message status as a text string.

```
HRESULT ChatMessageTypeToText ( [in] TChatMessageType Type  
[out, retval] BSTR * pVal  
)
```

This command converts the message type code to text.

**Parameters:**

[in] **Type** contains the message type code.

[out] **pVal** receives the message type as a text string.

```
HRESULT ChatStatusToText ( [in] TChatStatus    status,  
                           [out, retval] BSTR * pVal  
                           )
```

This command converts a chat status code to a text string.

**Parameters:**

[in] **status** contains the chat status code.

[out] **pVal** receives the converted chat status as a text string.

```
HRESULT ConnectionStatusToText ( [in] TConnectionStatus Status,  
                                  [out, retval] BSTR *    pVal  
                                  )
```

This command converts the connection status to a text string.

**Parameters:**

[in] **Status** contains the connection status value.

[out] **pVal** receives the connection status value as a text string.

```
HRESULT GroupTypeToText ( [in] TGroupType    Type,  
                           [out, retval] BSTR * pVal  
                           )
```

This command converts a group type to a text string.

**Parameters:**

- [in] **Type** contains the group type code.
- [out] **pVal** receives the group type as a text string.

```
HRESULT OnlineStatusToText ( [in] TOnlineStatus status,  
                             [out, retval] BSTR * pVal  
                             )
```

This command converts a user's online status from status code to text.

**Parameters:**

- [in] **status** is the online status value.
- [out] **pVal** receives a text string of the current online status.

```
HRESULT SmsMessageStatusToText ( [in] TSmsMessageStatus st  
                                 [out, retval] BSTR * p\  
                                 )
```

This command converts SMS message status to a text string.

**Parameters:**

- [in] **status** contains SMS message status.
- [out] **pVal** receives the status as text string.

```
HRESULT SmsMessageTypeToText ( [in] TSmsMessageType type,  
                                [out, retval] BSTR * pVal  
                                )
```

This command converts SMS message type to a text string.

**Parameters:**



[in] **type** contains SMS message type.  
[out] **pVal** receives the type as text string.

```
HRESULT SmsTargetStatusToText ( [in] TSmsTargetStatus status,  
                                [out, retval] BSTR * pVal  
                                )
```

This command converts SMS target status to a text string.

**Parameters:**

[in] **status** contains SMS target status.  
[out] **pVal** receives the status as text string.

```
HRESULT TextToAttachmentStatus ( [in] BSTR  
                                 [out, retval] TAttachmentStatu  
                                 )
```

This command converts an attachment status text string to code.

**Parameters:**

[in] **Text** contains a text string  
("UNKNOWN","SUCCESS","PENDING","REFUSED"  
[out] **pVal** receives the converted text attachment status as as

```
HRESULT TextToBuddyStatus ( [in] BSTR  
                             [out, retval] TBuddyStatus * pVal  
                             )
```

This command converts contact status text to contact status code.

**Parameters:**

[in] **Text** contains a text string of the contact status.  
[out] **pVal** receives the contact status code.

```
HRESULT TextToCallStatus ( [in] BSTR Text,  
                          [out, retval] TCallStatus * pVal  
                          )
```

This command converts call status text to call status code.

**Parameters:**

- [in] **Text** contains a string ("UNPLACED", "ROUTING", "EARLYMEDIA", "FAILED", "RINGING", "INPROGRESS", "ONHOLD", "FINISHED", "MISSED", "REFUSED", "BUSY", "LOCALHOLD", "REMOTEHOLD").
- [out] **pVal** receives the call status text as a call status code.

```
HRESULT TextToCallType ( [in] BSTR Text,  
                        [out, retval] TCallType * pVal  
                        )
```

This command converts call type text to call type code.

**Parameters:**

- [in] **Text** contains a text string ("INCOMING\_PSTN", "OUTGOING\_PSTN", "INCOMING\_P2P", "OUTGOING\_P2P").
- [out] **pVal** receives the call type as call type code.

```
HRESULT TextToChatMessageStatus ( [in] BSTR  
                                  [out, retval] TChatMessageS  
                                  )
```

This command converts a message status from text to code.

**Parameters:**

[in] **Text** contains a text string of the message status.  
[out] **pVal** receives the text string as message status code.

```
HRESULT TextToChatMessageType ( [in] BSTR  
                                [out, retval] TChatMessageTy  
                                )
```

This command converts a message type from text string to code.

**Parameters:**

[in] **Text** contains a text string for the message type.  
[out] **pVal** receives the message type in code.

```
HRESULT TextToConnectionStatus ( [in] BSTR  
                                  [out, retval] TConnectionStatu  
                                  )
```

This command converts a text string of connection status to a code value.

**Parameters:**

[in] **Text** contains a text string ("OFFLINE",  
"CONNECTING", "PAUSING", "ONLINE",  
"LOGGEDOUT").  
[out] **pVal** receives the converted text string as connection  
status code.

```
HRESULT TextToGroupType ( [in] BSTR  
                           [out, retval] TGroupType * pVal  
                           )
```

This command converts a text group type to a code value.

**Parameters:**

[in] **Text** contains a text string ("CUSTOM\_GROUP", "ALL\_USERS", "ALL\_FRIENDS", "SKYPE\_FRIENDS", "SKYPEOUT\_FRIENDS", "ONLINE\_FRIENDS", "UNKNOWN\_OR\_PENDINGAUTH\_FRIENDS", "RECENTLY\_CONTACTED\_USERS", "USERS\_WAITING\_MY\_AUTHORIZATION", "USERS\_AUTHORIZED\_BY\_ME", "USERS\_BLOCKED\_BY\_ME", "UNGROUPED\_FRIENDS", "SHARED\_GROUP", "PROPOSED\_SHARED\_GROUP")

[out] **pVal** receives the converted text group type as code.

**HRESULT TextToOnlineStatus** ( [in] BSTR **Text**,  
[out, retval] TOnlineStatus \* **pVal**  
)

This command converts text to online status code.

**Parameters:**

[in] **Text** contains a text string ("OFFLINE", "ONLINE", "AWAY", "NA", "DND", "SKYPEME", "SKYPEOUT").

[out] **pVal** receives the online status code.

**HRESULT TextToUserSex** ( [in] BSTR **Text**,  
[out, retval] TUserSex \* **pVal**  
)

This command converts the user sex text string to code.

**Parameters:**

[in] **Text** contains a text string ("UNKNOWN", "MALE",

"FEMALE").  
[out] **pVal** receives the converted text string as user sex code.

**HRESULT TextToUserStatus ( [in] BSTR Text,  
[out, retval] TUserStatus \* pVal  
)**

This command converts a string of user status to a code value.

**Parameters:**

[in] **Text** contains a text string ("OFFLINE", "ONLINE", "AWAY", "NA", "DND", "INVISIBLE", "LOGGEDOUT", "SKYPEME").  
[out] **pVal** receives the converted text string as a user status code.

**HRESULT TextToVoicemailStatus ( [in] BSTR  
[out, retval] TVoicemailStatus \*  
)**

This command converts a voicemail status text string to code.

**Parameters:**

[in] **Text** contains a text string ("NOTDOWNLOADED", "DOWNLOADING", "UNPLAYED", "BUFFERING", "PLAYING", "PLAYED", "BLANK", "RECORDING", "RECORDED", "UPLOADING", "UPLOADED", "DELETING", "FAILED", "UNKNOWN").  
[out] **pVal** receives the converted voicemail status text as code.

**HRESULT UserSexToText ( [in] TUserSex Sex,**

```
[out, retval] BSTR * pVal  
)
```

This command converts user sex value to text.

**Parameters:**

[in] **Sex** contains the user sex value.

[out] **pVal** receives the user sex value as a text string.

```
HRESULT UserStatusToText ( [in] TUserStatus Status,  
[out, retval] BSTR * pVal  
)
```

This command converts the user status value to a text string.

**Parameters:**

[in] **Status** contains the user status value.

[out] **pVal** receives the user status value as a text string.

```
HRESULT VoicemailFailureReasonToText ( [in] TVoicemailFailureReason code,  
[out, retval] BSTR * pVal  
)
```

This command converts a voicemail failure reason to a text string.

**Parameters:**

[in] **code** contains the voicemail failure reason code.

[out] **pVal** receives the converted voicemail failure reason as a text string.

```
HRESULT VoicemailStatusToText ( [in] TVoicemailStatus status,  
[out, retval] BSTR * pVal  
)
```

This command converts a voicemail status to a text string.

**Parameters:**

[in] **status** contains the voicemail status code.

[out] **pVal** receives the converted voicemail status as a text string.

```
HRESULT VoicemailTypeToText ( [in] TVoicemailType type,  
                             [out, retval] BSTR * pVal  
                             )
```

This command converts a voicemail type to a text string.

**Parameters:**

[in] **type** contains the voicemail type code.

[out] **pVal** receives the converted voicemail type as a text string.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Properties](#)



# IFileTransfer Interface Reference

---

File transfer object default interface. [More...](#)

```
import "Skype4COM.idl";
```

[List of all members.](#)

## Properties

BSTR	<b>Id</b> [get] This property returns the file transfer unique identifier.
<b>TFileTransferType</b>	<b>Type</b> [get] This property returns the file transfer type.
<b>TFileTransferStatus</b>	<b>Status</b> [get] This property returns the file transfer status.
<b>TFileTransferFailureReason</b>	<b>FailureReason</b> [get] This property returns the file transfer failure reason code.
BSTR	<b>PartnerHandle</b> [get] This property returns the file transfer partner skype name.
BSTR	<b>PartnerDisplayName</b> [get] This property returns the file transfer partner display name.
DATE	<b>StartTime</b> [get] This property returns the file transfer start timestamp.
DATE	<b>FinishTime</b> [get] This property returns the file transfer end timestamp.
BSTR	<b>FilePath</b> [get] This property returns the full path of the file being transferred.
BSTR	<b>FileName</b> [get] This property returns the file name of transferred file.
BSTR	<b>BytesPerSecond</b> [get] This property returns the transfer speed in bytes per second.

---

BSTR **BytesTransferred** [get]

This property returns the number of bytes transferred.

---

## Detailed Description

File transfer object default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IFileTransferCollection Interface Reference

---

File transfer collection class default interface. [More...](#)

```
import "Skype4COM.idl";
```

[List of all members.](#)

## Public Member Functions

HRESULT **Add** ([in] **IFileTransfer** \*pItem)

Adds a new item to a collection.

HRESULT **Remove** ([in] long Index)

This command removes an item from a collection.

HRESULT **RemoveAll** (void)

This command removes all items from a collection.

## Properties

---

LONG	<b>Count</b> [get]	Returns number of items in a collection.
<b>IFileTransfer</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item in a collection with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries an enumerator object for a collection.

---



## Detailed Description

File transfer collection class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] IFileTransfer \* pItem )**

Adds a new item to a collection.

**Parameters:**

[in] **pItem** contains the interface pointer for the File Transfer object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a collection.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item in a collection with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IGroup Interface Reference

---

**Group** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **AddUser** ([in] BSTR Username)

This command adds a user to a group.

HRESULT **RemoveUser** ([in] BSTR Username)

This command removes a user from a group.

HRESULT **Share** ([in, defaultvalue("")] BSTR MessageText)

This command shares a contact group.

HRESULT **Accept** ()

This command accepts an invitation to join a shared contact group.

HRESULT **Decline** ()

This command declines an invitation to join a shared contact group.

## Properties

LONG	<b>Id</b> [get]	This command queries the group ID.
<b>TGroupType</b>	<b>Type</b> [get]	This command queries a group type.
BSTR	<b>CustomGroupId</b> [get]	This command queries a group persistent ID.
BSTR	<b>DisplayName</b> [get, set]	This command queries/sets the display name of a group.
<b>IUserCollection</b>	<b>Users</b> [get]	This command queries the users in a group.
<b>IUserCollection</b>	<b>OnlineUsers</b> [get]	This command queries the online users in a group.
VARIANT_BOOL	<b>IsVisible</b> [get]	This command queries the visibility status of a group.
VARIANT_BOOL	<b>IsExpanded</b> [get]	This command queries the expanded status of a group.

## Detailed Description

**Group** class default interface.

---



## Member Function Documentation

### **HRESULT AddUser ( [in] BSTR Username )**

This command adds a user to a group.

**Parameters:**

[in] **Username** contains the Skype name of the contact to add.

### **HRESULT RemoveUser ( [in] BSTR Username )**

This command removes a user from a group.

**Parameters:**

[in] **Username** contains the Skype name of the contact to remove.

### **HRESULT Share ( [in, defaultvalue("")] BSTR MessageText )**

This command shares a contact group.

**Parameters:**

[in] **MessageText** contains the message text for group members.

---

## Property Documentation

### **HRESULT CustomGroupId** [get]

This command queries a group persistent ID.

The custom group ID is a persistent value that does not change.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IGroupCollection Interface Reference

---

**GroupCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IGroup** \*pGroup)

This command adds a new item to a **GroupCollection**.

HRESULT **Remove** ([in] long Index)

This command removes an item from a **GroupCollection**.

HRESULT **RemoveAll** (void)

This command removes all items from a **GroupCollection**.

## Properties

---

LONG	<b>Count</b> [get]	This command queries the number of groups in a <b>GroupCollection</b> .
<b>IGroup</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a <b>GroupCollection</b> item with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries a <b>GroupCollection</b> enumerator object.

---

## Detailed Description

**GroupCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] IGroup \* pGroup )**

This command adds a new item to a **GroupCollection**.

**Parameters:**

[in] **pGroup** contains the interface pointer for the **Group** object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a **GroupCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---



## Property Documentation

### HRESULT Item [get]

This command queries a **GroupCollection** item with a specified index.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# IParticipant Interface Reference

---

**Participant** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

BSTR	<b>Handle</b> [get]	This command queries the Skype name of a participant in a conference call.
BSTR	<b>DisplayName</b> [get]	This command queries the display name of a participant in a conference call.
<b>TCallType</b>	<b>CallType</b> [get]	This command queries a call type in a conference call.
<b>TCallStatus</b>	<b>CallStatus</b> [get]	This command queries the call status of a participant in a conference call.

## Detailed Description

**Participant** class default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IParticipantCollection Interface Reference

---

**ParticipantCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IParticipant** \*pParticipant)  
This command adds a new participant to a **ParticipantCollection**.

HRESULT **Remove** ([in] long Index)  
This command removes an item from a **ParticipantCollection**.

HRESULT **RemoveAll** (void)  
This command removes all items from a **ParticipantCollection**.



## Properties

---

LONG	<b>Count</b> [get]	This command queries the number of participants in a <b>ParticipantCollection</b> .
<b>IParticipant</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item with a specified index in a <b>ParticipantCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command queries the enumerator object of a <b>ParticipantCollection</b> .

---

## Detailed Description

**ParticipantCollection** class default interface.

---

## Member Function Documentation

### HRESULT Add ( [in] IParticipant \* pParticipant )

This command adds a new participant to a **ParticipantCollection**.

**Parameters:**

[in] **pParticipant** contains the interface pointer for the **Participant** object.

### HRESULT Remove ( [in] long Index )

This command removes an item from a **ParticipantCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item with a specified index in a **ParticipantCollection**.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

## IPluginEvent Interface Reference

---

This is the default interface for the **PluginEvent** class. [More...](#)

```
import "Skype4COM.idl";
```

[List of all members.](#)

## Public Member Functions

HRESULT **Delete** (void)

This command deletes a plug-in event from the events pane in the Skype client.

## Properties

BSTR **Id** [get]

This property is a unique identifier for a plug-in event.

---



## Detailed Description

This is the default interface for the **PluginEvent** class.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IPluginMenuItem Interface Reference

---

Class **PluginMenuItem** default interface. [More...](#)

```
import "Skype4COM.idl";
```

[List of all members.](#)

## Public Member Functions

---

HRESULT **Delete** (void)

This command removes a menu item from "Do More" menus.

## Properties

BSTR	<b>Id</b> [get]	This property is a unique identifier for a plug-in menu item.
BSTR	<b>Caption</b> [set]	This property contains the name of a plug-in menu item and must be a maximum of 32 characters.
BSTR	<b>Hint</b> [set]	This property contains a second line of text for a plug-in menu item if required and must be a maximum of 32 characters.
VARIANT_BOOL	<b>Enabled</b> [set]	This property is a Boolean indicator that defines whether a menu item is enabled when a user launches the application. If no value is defined, the menu item will be enabled.

## Detailed Description

Class [PluginMenuItem](#) default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# IProfile Interface Reference

---

**Profile** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.



## Properties

BSTR	<b>FullName</b> [get, set]	This command queries/sets the full name in the current user profile.
BSTR	<b>Birthday</b> [get, set]	This command queries/sets the birthday in the current user profile.
<b>TUserSex</b>	<b>Sex</b> [get, set]	This command queries/sets the sex in the current user profile.
BSTR	<b>Languages</b> [get, set]	This command queries/sets the language code in the current user profile.
BSTR	<b>Country</b> [get, set]	This command queries/sets the ISO country code in the current user profile.
BSTR	<b>Province</b> [get, set]	This command queries/sets the province in the current user profile.
BSTR	<b>City</b> [get, set]	This command queries/sets the city in the current user profile.
BSTR	<b>PhoneHome</b> [get, set]	This command queries/sets the home phone number in the current user profile.
BSTR	<b>PhoneOffice</b> [get, set]	This command queries/sets the business phone number in the current user profile.
BSTR	<b>PhoneMobile</b> [get, set]	This command queries/sets the mobile phone number in the current user profile.
BSTR	<b>Homepage</b> [get, set]	This command queries/sets the homepage url in the current user profile.

BSTR	<b>About</b> [get, set] This command queries/sets the "About" text in the current user profile.
BSTR	<b>MoodText</b> [get, set] This command queries/sets the mood text in the current user profile.
LONG	<b>Timezone</b> [get, set] This command queries/sets the timezone in the current user profile.
short	<b>CallNoAnswerTimeout</b> [get, set] This command queries/sets the "call no answer timeout" in the current user profile.
VARIANT_BOOL	<b>CallApplyCF</b> [get, set] This command queries/sets if call forwarding is set in the current user profile.
VARIANT_BOOL	<b>CallSendToVM</b> [get, set] This command queries/sets whether calls will be sent to voicemail in the current user profile.
BSTR	<b>CallForwardRules</b> [get, set] This command queries/sets the call forwarding rules in the current user profile.
LONG	<b>Balance</b> [get] This command queries the balance in currency cents in the current user profile.
BSTR	<b>BalanceCurrency</b> [get] This command queries the currency code in the current user profile.
BSTR	<b>BalanceToText</b> [get] This command queries the balance amount with currency symbol in the current user profile.
BSTR	<b>IPCountry</b> [get] This command queries the ISO country code by IP address.
BSTR	<b>ValidatedSmsNumbers</b> [get] This command requests a list of validated SMS numbers.

---

BSTR **RichMoodText** [get, set]

This property returns/sets rich mood text in logged in user profile.

---

## Detailed Description

**Profile** class default interface.

---

## Property Documentation

### **HRESULT CallNoAnswerTimeout** [get, set]

This command queries/sets the "call no answer timeout" in the current user profile.

The call no answer timeout is the period of time a call will ring without being answered before it stops ringing.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ISettings Interface Reference

---

**Settings** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>SaveAvatarToFile</b> ([in] BSTR Filename,[in, defaultvalue("1")] BSTR AvatarId) Saves current user avatar to file.
HRESULT	<b>LoadAvatarFromFile</b> ([in] BSTR Filename,[in, defaultvalue("1")] BSTR AvatarId) Replaces user current avatar with new avatar from file.
HRESULT	<b>ResetIdleTimer</b> (void) Reset Skype idle timer.



## Properties

BSTR	<b>AudiIn</b> [get, set] This command queries/sets the name of an audio input device.
BSTR	<b>AudioOut</b> [get, set] This command queries/sets the name of an audio output device.
VARIANT_BOOL	<b>AutoAway</b> [get, set] This command queries/sets the "Auto away" status.
BSTR	<b>Ringer</b> [get, set] This command queries/sets the name of a ringer device.
BSTR	<b>Avatar</b> ([in, defaultvalue("1")] BSTR Id) [set] This command sets the avatar picture in a user profile.
VARIANT_BOOL	<b>RingToneStatus</b> ([in, defaultvalue("1")] BSTR Id) [get, set] This command queries or sets the ringtone status.
BSTR	<b>RingTone</b> ([in, defaultvalue("1")] BSTR Id) [set] This command sets a new ringtone.
BSTR	<b>VideoIn</b> [get, set] This command queries/sets the name of a video input device.
VARIANT_BOOL	<b>PCSpeaker</b> [get, set] This command queries/sets the PC speaker status.
VARIANT_BOOL	<b>AGC</b> [get, set] This command queries/sets the status of automatic gain control.
VARIANT_BOOL	<b>AEC</b> [get, set] This command queries/sets the status of automatic echo cancellation.
BSTR	<b>Language</b> [get, set]

This command queries/sets the language of the Skype client.

---

## Detailed Description

**Settings** class default interface.

---

## Member Function Documentation

**HRESULT LoadAvatarFromFile** ( [in] BSTR **Filename**  
[in, defaultvalue("1")] BSTR **AvatarId**  
)

Replaces user current avatar with new avatar from file.

### Parameters:

- [in] **Filename** contains file name.
- [in] **AvatarId** avatar id.

### Example:

```
Set oSkype = CreateObject("Skype4COM.Skype")  
oSkype.Settings.LoadAvatarFromFile("c:\temp\  
new_avatar.jpg")
```

**HRESULT SaveAvatarToFile** ( [in] BSTR **Filename**  
[in, defaultvalue("1")] BSTR **AvatarId**  
)

Saves current user avatar to file.

### Parameters:

- [in] **Filename** contains file name.
- [in] **AvatarId** avatar id.

### Example:

```
Set oSkype = CreateObject("Skype4COM.Skype")  
oSkype.Settings.SaveAvatarToFile("c:\temp\ol  
d_avatar.jpg")
```

## Property Documentation

### HRESULT Avatar [set]

This command sets the avatar picture in a user profile.

**Parameters:**

- [in] **id** is an optional avatar ID.
- [in] **newVal** contains the new file name.

**Deprecated:**

Use the **ISettings::LoadAvatarFromFile** instead.

### HRESULT RingTone [set]

This command sets a new ringtone.

**Parameters:**

- [in] **id** contains the ringtone ID.

### HRESULT RingToneStatus [get, set]

This command queries or sets the ringtone status.

**Parameters:**

- [in] **id** contains the ringtone ID.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ISkype Interface Reference

---

**Skype** is the default interface for the class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>SearchForUsers</b> ([in] BSTR Target,[out, retval] <b>IUserCollection</b> **pVal) This command searches for users.
HRESULT	<b>Attach</b> ([in, defaultvalue(5)] LONG Protocol,[in, defaultvalue(-1)] VARIANT_BOOL Wait) This command establishes a connection to Skype.
HRESULT	<b>PlaceCall</b> ([in] BSTR Target,[in, defaultvalue("")] BSTR Target2,[in, defaultvalue("")] BSTR Target3,[in, defaultvalue("")] BSTR Target4,[out, retval] <b>ICall</b> **pCall) This command places a call to a single user or creates a conference call.
HRESULT	<b>SendMessage</b> ([in] BSTR Username,[in] BSTR Text,[out, retval] <b>IChatMessage</b> **pMessage) This command sends a chat message.
HRESULT	<b>SendCommand</b> ([in] <b>ICommand</b> *pCommand) This command sends an API command.
HRESULT	<b>ChangeUserStatus</b> ([in] <b>TUserStatus</b> newVal) This command changes the online status for the current user.
HRESULT	<b>CreateChatWith</b> ([in] BSTR Username,[out, retval] <b>IChat</b> **pVal) This command creates a chat with a single user.
HRESULT	<b>CreateChatMultiple</b> ([in] <b>IUserCollection</b> *pMembers, [out, retval] <b>IChat</b> **pChat) This command creates a chat with multiple users.
HRESULT	<b>SendVoicemail</b> ([in] BSTR Username,[out, retval] <b>IVoicemail</b> **pVoicemail) This command sends a voicemail to a specified user.
HRESULT	<b>ClearChatHistory</b> () This command clears chat history.
HRESULT	<b>ClearVoicemailHistory</b> () This command clears voicemail history.



HRESULT	<b>ClearCallHistory</b> ([in, defaultvalue("ALL")] BSTR Username,[in, defaultvalue(chsAllCalls)] <b>TCallHistory</b> type) This command clears call history.
HRESULT	<b>ResetCache</b> () This command deletes all cache entries.
HRESULT	<b>CreateGroup</b> ([in] BSTR GroupName,[out, retval] <b>IGroup</b> **pGroup) This command creates a contact group.
HRESULT	<b>DeleteGroup</b> ([in] LONG GroupId) This command deletes a contact group. Users in the contact group are moved to the All Contacts hardwired contact group.
HRESULT	<b>EnableApiSecurityContext</b> ([in] <b>TApiSecurityContext</b> Context) This command enables an API security context for Internet Explorer scripts.
HRESULT	<b>CreateSms</b> ([in] <b>TSmsMessageType</b> MessageType,[in] BSTR TargetNumbers,[out, retval] <b>ISmsMessage</b> **pMessage) This command creates an <b>SmsMessage</b> .
HRESULT	<b>SendSms</b> ([in] BSTR TargetNumbers,[in] BSTR MessageText,[in, defaultvalue("")] BSTR ReplyToNumber, [out, retval] <b>ISmsMessage</b> **pMessage) This command sends an SMS message.
HRESULT	<b>AsyncSearchUsers</b> ([in] BSTR Target,[out, retval] LONG *pCookie) This command can be used to search for Skype users.
HRESULT	<b>FindChatUsingBlob</b> ([in] BSTR Blob,[out, retval] <b>IChat</b> **pChat) Returns existing chat using given blob.
HRESULT	<b>CreateChatUsingBlob</b> ([in] BSTR Blob,[out, retval] <b>IChat</b> **pChat) Returns existing or joins a new chat using given blob.

## Properties

LONG	<b>Timeout</b> [get, set] This command queries/sets the wait timeout value in milliseconds.
BSTR	<b>Property</b> ([in] BSTR ObjectType,[in] BSTR ObjectId,[in] BSTR PropName) [get, set] This command queries/sets the properties of an object.
BSTR	<b>Variable</b> ([in] BSTR Name) [get, set] This command queries/sets Skype general parameters.
BSTR	<b>CurrentUserHandle</b> [get] This command queries the Skype name of the current user.
<b>TUserStatus</b>	<b>CurrentUserStatus</b> [get, set] This command queries/sets the online status of the current user.
<b>TConnectionStatus</b>	<b>ConnectionStatus</b> [get] This command queries the connection status of the Skype client.
VARIANT_BOOL	<b>Mute</b> [get, set] This command queries/sets the mute status of the Skype client.
BSTR	<b>Version</b> [get] This command queries the application version of the Skype client.
VARIANT_BOOL	<b>Privilege</b> ([in] BSTR Name) [get] This command queries the Skype services (privileges) enabled for the Skype client.
<b>IUser</b>	<b>CurrentUser</b> [get] This command queries the current user object.

<b>IConversion</b>	<b>Convert</b> [get] This command queries the conversion object.
<b>IUserCollection</b>	<b>Friends</b> [get] This command queries the users in a contact list.
<b>ICallCollection</b>	<b>Calls</b> ([in, defaultvalue("")] BSTR Target) [get] This command queries calls in call history.
<b>ICallCollection</b>	<b>ActiveCalls</b> [get] This command queries a list of active calls.
<b>ICallCollection</b>	<b>MissedCalls</b> [get] This command queries a list of missed calls.
<b>IChatMessageCollection</b>	<b>Messages</b> ([in, defaultvalue("")] BSTR Target) [get] This command queries chat messages which were sent/received by the user.
<b>IChatMessageCollection</b>	<b>MissedMessages</b> [get] This command queries a list of missed chat messages.
<b>TAttachmentStatus</b>	<b>AttachmentStatus</b> [get] This command queries the attachment status of the Skype client.
<b>LONG</b>	<b>Protocol</b> [get, set] This command queries/sets the protocol version used by the Skype client.
<b>IUser</b>	<b>User</b> ([in, defaultvalue("")] BSTR Username) [get] This command queries a user object.
<b>IChatMessage</b>	<b>Message</b> ([in, defaultvalue(0)] LONG Id) [get] This command queries a chat message object.

<b>ICall</b>	<b>Call</b> ([in, defaultvalue(0)] LONG Id) [get] This command queries a call object.
<b>IChatCollection</b>	<b>Chats</b> [get] This command queries a list of chats.
<b>IChat</b>	<b>Chat</b> ([in, defaultvalue("")] BSTR Name) [get] This command queries a chat object.
<b>IConference</b>	<b>Conference</b> ([in, defaultvalue(0)] LONG Id) [get] This command queries a call conference object.
<b>IConferenceCollection</b>	<b>Conferences</b> [get] This command queries a list of call conferences.
<b>BSTR</b>	<b>Profile</b> ([in] BSTR Property) [get, set] This command queries/sets user profile properties.
<b>IChatCollection</b>	<b>ActiveChats</b> [get] This command queries a list of active chats.
<b>IChatCollection</b>	<b>MissedChats</b> [get] This command queries a list of missed chats.
<b>IChatCollection</b>	<b>RecentChats</b> [get] This command queries a list of recent chats.
<b>IChatCollection</b>	<b>BookmarkedChats</b> [get] This command queries a list of bookmarked chats.
<b>IVoicemailCollection</b>	<b>Voicemails</b> [get] This command queries a list of voicemails.
<b>IUserCollection</b>	<b>UsersWaitingAuthorization</b> [get] This command queries the list of users waiting for authorization.

VARIANT_BOOL	<b>CommandId</b> [get, set] This command queries/sets the status of automatic command identifiers.
<b>IApplication</b>	<b>Application</b> ([in] BSTR Name) [get] This command queries an application object.
<b>IVoicemail</b>	<b>Greeting</b> ([in, defaultvalue("")] BSTR Username) [get] This command queries the greeting used as voicemail.
VARIANT_BOOL	<b>Cache</b> [get, set] This command queries/sets the status of internal cache.
<b>IProfile</b>	<b>CurrentUserProfile</b> [get] This command queries the user profile object.
<b>IGroupCollection</b>	<b>Groups</b> [get] This command queries the list of all contact groups.
<b>IGroupCollection</b>	<b>CustomGroups</b> [get] This command queries the list of custom contact groups.
<b>IGroupCollection</b>	<b>HardwiredGroups</b> [get] This command queries the list of hardwired contact groups.
<b>ISettings</b>	<b>Settings</b> [get] This command queries the settings for Skype general parameters.
<b>IClient</b>	<b>Client</b> [get] This command queries the user interface control object.
BSTR	<b>FriendlyName</b> [set] This command sets a "friendly" name for an application.
	<b>Command</b> ([in] LONG Id,[in] BSTR

<b>ICommand</b>	<p>Command,[in, defaultvalue("")] BSTR Reply,[in, defaultvalue(0)] VARIANT_BOOL Block,[in, defaultvalue(30000)] LONG Timeout) [get]</p> <p>This command sets the properties of a command object.</p>
<b>IVoicemail</b>	<p><b>Voicemail</b> ([in] LONG Id) [get]</p> <p>This command queries the voicemail object.</p>
<b>IVoicemailCollection</b>	<p><b>MissedVoicemails</b> [get]</p> <p>This command requests a list of missed voicemails.</p>
VARIANT_BOOL	<p><b>ApiSecurityContextEnabled</b> ([in] <b>TApiSecurityContext</b> Context) [get]</p> <p>This command queries if an API security context for Internet Explorer is enabled.</p>
<b>ISmsMessageCollection</b>	<p><b>Smss</b> [get]</p> <p>This command requests a list of all SMS messages.</p>
<b>ISmsMessageCollection</b>	<p><b>MissedSmss</b> [get]</p> <p>This command requests a list of all missed SMS messages.</p>
BSTR	<p><b>ApiWrapperVersion</b> [get]</p> <p>This command returns API wrapper DLL version.</p>
VARIANT_BOOL	<p><b>SilentMode</b> [get, set]</p> <p>This command returns/sets Skype silent mode status.</p>
<b>IFileTransferCollection</b>	<p><b>FileTransfers</b> [get]</p> <p>Returns all file transfers.</p>
<b>IFileTransferCollection</b>	<p><b>ActiveFileTransfers</b> [get]</p> <p>Returns currently active file transfers.</p>
<b>IUserCollection</b>	<p><b>FocusedContacts</b> [get]</p> <p>Returns collection of selected contacts.</p>
BSTR	<p><b>PredictiveDialerCountry</b> [get]</p>

Returns predictive dialer country code.

---

## Detailed Description

**Skype** is the default interface for the class.

---



## Member Function Documentation

**HRESULT AsyncSearchUsers ( [in] BSTR Target, [out,retval] LONG \* pCookie )**

This command can be used to search for Skype users.

**Parameters:**

[in] **Target** contains the person name or email address to be found.

[out] **pCookie** receives search identifier.

**See also:**

[\\_ISkypeEvents::AsyncSearchUsersFinished](#)

**HRESULT Attach ( [in, defaultvalue(5)] LONG Protocol, [in, defaultvalue(-1)] VARIANT\_BOOL Wait )**

This command establishes a connection to Skype.

**Parameters:**

[in] **Protocol** is an optional field where you can enter the Skype protocol version, with a default value of 5.

[in] **Wait** is an optional Boolean value which indicates whether or not to wait for a response before establishing a connection.

**HRESULT ChangeUserStatus ( [in] TUserStatus newVal )**

This command changes the online status for the current user.

**Parameters:**

[in] **newVal** contains the new online status for the user.

**Note:**

This function waits until the online status changes. Alternatively, use the **ISkype::CurrentUserStatus** property to perform an immediate change of status.

**HRESULT ClearCallHistory ( [in, defaultvalue("ALL")] BSTR  
[in, defaultvalue(chsAllCalls)] TCallHis  
)**

This command clears call history.

**Parameters:**

[in] **Username** is the Skype name of the user.  
[in] **type** call type.

**HRESULT CreateChatMultiple ( [in] IUserCollection \* pMembers,  
[out, retval] IChat \*\* pVal  
)**

This command creates a chat with multiple users.

**Parameters:**

[in] **pMembers** contains an interface pointer to the **UserCollection** object.  
[out] **pVal** receives the interface pointer for the **Chat** object.

**See also:**

**ISkype::CreateChatWith, IChat::AddMembers**

**HRESULT CreateChatUsingBlob ( [in] BSTR Blob,**

```
[out,retval] IChat ** pChat  
)
```

Returns existing or joins a new chat using given blob.

**Parameters:**

[in] **Blob** contains unique string indentifying the chat.  
[out] **pChat** receives the interface pointer for the **Chat** object.

```
HRESULT CreateChatWith ([in] BSTR Username,  
[out, retval] IChat ** pVal  
)
```

This command creates a chat with a single user.

**Parameters:**

[in] **Username** contains the Skype name of the user.  
[out] **pVal** receives the interface pointer for the **Chat** object.

**See also:**

[ISkype::CreateChatMultiple](#), [IChat::AddMembers](#)

```
HRESULT CreateGroup ([in] BSTR GroupName,  
[out, retval] IGroup ** pGroup  
)
```

This command creates a contact group.

**Parameters:**

[in] **GroupName** group name.  
[out] **pGroup** contains the interface pointer for the **Group** object.

```
HRESULT CreateSms ( [in] TSmsMessageType      MessageType,
                   [in] BSTR                    TargetNumber
                   [out, retval] ISmsMessage ** pMessage
                   )
```

This command creates an **SmsMessage**.

**Parameters:**

[in] **MessageType** contains the message type.  
[in] **TargetNumbers** contains a comma-separated list of target SMS numbers.  
[out] **pMessage** receives the interface pointer for the **SmsMessage** object.

```
HRESULT DeleteGroup ( [in] LONG GroupId )
```

This command deletes a contact group. Users in the contact group are moved to the All Contacts hardwired contact group.

**Parameters:**

[in] **GroupId** group identifier.

```
HRESULT EnableApiSecurityContext ( [in] TApiSecurityContext Context )
```

This command enables an API security context for Internet Explorer scripts.

**Parameters:**

[in] **Context** combination of API security context values.

```
HRESULT FindChatUsingBlob ( [in] BSTR          Blob,
                           [out,retval] IChat ** pChat
```

)

Returns existing chat using given blob.

**Parameters:**

- [in] **Blob** contains unique string indentifying the chat.
- [out] **pChat** receives the interface pointer for the **Chat** object.

```
HRESULT PlaceCall ( [in] BSTR Target,  
                   [in, defaultvalue("")] BSTR Target2,  
                   [in, defaultvalue("")] BSTR Target3,  
                   [in, defaultvalue("")] BSTR Target4,  
                   [out, retval] ICall ** pCall  
                   )
```

This command places a call to a single user or creates a conference call.

**Parameters:**

- [in] **Target** call target to place a single user call.
- [in] **Target2** is an optional field which can contain an additional participant for a conference call.
- [in] **Target3** is an optional field which can contain an additional participant for a conference call.
- [in] **Target4** is an optional field which can contain an additional participant for a conference call.
- [out] **pCall** receives the interface pointer for the **Call** object.

**Note:**

If multiple targets are specified, a conference call is created. The call target can be username, phone number, or speed dial code.

## HRESULT ResetCache ( )

This command deletes all cache entries.

### See also:

Refer to the Skype4COM Dev Notes for information about the API cache.

## HRESULT SearchForUsers ( [in] BSTR Target, [out, retval] IUserCollection \*\* pVal )

This command searches for users.

### Parameters:

[in] **Target** search target.

[out] **pVal** receives the interface pointer for the **UserCollection** object.

## HRESULT SendCommand ( [in] ICommand \* pCommand )

This command sends an API command.

### Parameters:

[in] **pCommand** contains the interface pointer for the **Command** object.

## HRESULT SendMessage ( [in] BSTR Username, [in] BSTR Text, [out, retval] IChatMessage \*\* pMessage )

This command sends a chat message.

**Parameters:**

- [in] **Username** user handle is the Skype name of the user.
- [in] **Text** message text.
- [out] **pMessage** receives the interface pointer for the **ChatMessage** object.

```
HRESULT SendSms ( [in] BSTR TargetNumbers  
                 [in] BSTR MessageText,  
                 [in, defaultvalue("")] BSTR ReplyToNumber  
                 [out, retval] ISmsMessage ** pMessage  
                 )
```

This command sends an SMS message.

**Parameters:**

- [out] **TargetNumbers** contains a comma-separated list of target numbers.
- [out] **MessageText** contains text of the message to send.
- [out] **ReplyToNumber** contains the reply-to number.
- [out] **pMessage** receives the interface pointer for the **SmsMessage** object.

```
HRESULT SendVoicemail ( [in] BSTR Username,  
                       [out, retval] IVoicemail ** pVoicemail  
                       )
```

This command sends a voicemail to a specified user.

**Parameters:**

- [in] **Username** contains the Skype name of the user
- [out] **pVoicemail** receives the interface pointer for the **Voicemail** object.





## Property Documentation

### **HRESULT ApiSecurityContextEnabled** [get]

This command queries if an API security context for Internet Explorer is enabled.

**Parameters:**

[in] **Context** is the TApiSecurityContext value.

### **HRESULT Application** [get]

This command queries an application object.

**Parameters:**

[in] **Name** is the application name.

### **HRESULT Cache** [get, set]

This command queries/sets the status of internal cache.

The internal API cache is used to cache Skype object properties and global parameters.

### **HRESULT Call** [get]

This command queries a call object.

**Parameters:**

[in] **Id** is an optional field for the call ID.

### **HRESULT Calls** [get]

This command queries calls in call history.

**Parameters:**

[in] **Target** optional call target.

### HRESULT Chat [get]

This command queries a chat object.

**Parameters:**

[in] **Name** is an optional field for the chat name.

### HRESULT Command [get]

This command sets the properties of a command object.

**Parameters:**

[in] **Id** is the command ID.  
[in] **Command** is the command string.  
[in] **Reply** is the expected reply string.  
[in] **Block** is a blocking indicator. If set to True, **ISkype::SendCommand** method waits for a response from **Skype API**.  
[in] **Timeout** is the blocking timeout in milliseconds.

**See also:**

**ISkype::SendCommand**

### HRESULT Conference [get]

This command queries a call conference object.

**Parameters:**

[in] **Id** is an optional field for the conference ID.

### **HRESULT CustomGroups [get]**

This command queries the list of custom contact groups.

Custom groups are contact groups defined by the user.

### **HRESULT Greeting [get]**

This command queries the greeting used as voicemail.

**Parameters:**

[in] **Username** is an optional field containing the Skype name of the user.

### **HRESULT HardwiredGroups [get]**

This command queries the list of hardwired contact groups.

Hardwired groups are "smart" contact groups, defined by Skype, that cannot be removed.

### **HRESULT Message [get]**

This command queries a chat message object.

**Parameters:**

[in] **Id** is an optional field for the message ID.

### **HRESULT Messages [get]**

This command queries chat messages which were sent/received by the user.

**Parameters:**

[in] **Target** message sender.

**HRESULT Mute [get, set]**

This command queries/sets the mute status of the Skype client.

**Note:**

This value can be set only when there is an active call.

**HRESULT PredictiveDialerCountry [get]**

Returns predictive dialer country code.

**Parameters:**

[out] **pVal** receives the country code.

**HRESULT Privilege [get]**

This command queries the Skype services (privileges) enabled for the Skype client.

**Parameters:**

[in] **Name** privilege name  
("SKYPEOUT","SKYPEIN","VOICEMAIL").

**HRESULT Profile [get, set]**

This command queries/sets user profile properties.

**Parameters:**

[in] **Property** property name (for example  
"PSTN\_BALANCE",

```
"PSTN_BALANCE_CURRENCY",  
"FULLNAME", "BIRTHDAY", "SEX",  
"LANGUAGES", "COUNTRY", "PROVINCE",  
"CITY", "PHONE_HOME",  
"PHONE_OFFICE", "PHONE_MOBILE",  
"HOMEPAGE", "ABOUT")
```

### **HRESULT Property** [get, set]

This command queries/sets the properties of an object.

#### **Parameters:**

[in] **ObjectType** object type  
("USER","CALL","CHAT","CHATMESSAGE","  
[in] **ObjectId** object id.  
[in] **PropName** property name.

#### **Example:**

```
Set oSkype = CreateObject("Skype4COM.Skype"  
)  
oSkype.Attach  
WScript.Echo oSkype.Property("USER", "echo12  
3", "FULLNAME")
```

### **HRESULT Timeout** [get, set]

This command queries/sets the wait timeout value in milliseconds.

This timeout value applies to every command sent to the [Skype API](#). If a response is not received during the timeout period, an error is returned.

### **HRESULT User** [get]

This command queries a user object.

**Parameters:**

[in] **Username** is an optional field for the Skype name of the user.

### **HRESULT Variable** [get, set]

This command queries/sets Skype general parameters.

**Parameters:**

[in] **Name** variable name.

**Example:**

```
Set oSkype = CreateObject("Skype4COM.Skype")
oSkype.Attach
WScript.Echo oSkype.Variable("MUTE")
```

### **HRESULT Voicemail** [get]

This command queries the voicemail object.

**Parameters:**

[in] **Id** id the voicemail id.

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

## ISmsChunk Interface Reference

---

**SmsChunk** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.



## Properties

LONG	<b>Id</b> [get]	This command requests the ID of an SMS message chunk.
LONG	<b>CharactersLeft</b> [get]	This command requests the number of characters left in an SMS message chunk.
BSTR	<b>Text</b> [get]	This command queries the text in an SMS message chunk.
<b>ISmsMessage</b>	<b>Message</b> [get]	This command queries the <b>SmsMessage</b> associated with a chunk.

## Detailed Description

**SmsChunk** is the default interface for this class.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

## ISmsChunkCollection Interface Reference

---

**SmsChunkCollection** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **ISmsChunk** \*pItem)

This command adds a chunk to an **SmsChunkCollection**.

HRESULT **Remove** ([in] long Index)

This command removes a chunk from an **SmsChunkCollection**.

HRESULT **RemoveAll** (void)

This command removes all chunks from an **SmsChunkCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of chunks in a <b>SmsChunkCollection</b> .
<b>ISmsChunk</b>	<b>Item</b> ([in] LONG Index) [get]	This command requests a chunk with a specified index in an <b>SmsChunkCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command requests the enumerator object of an <b>SmsChunkCollection</b> .

## Detailed Description

**SmsChunkCollection** is the default interface for this class.

---

## Member Function Documentation

### **HRESULT Add ( [in] ISmsChunk \* pItem )**

This command adds a chunk to an **SmsChunkCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer to the **SmsChunk** object.

### **HRESULT Remove ( [in] long Index )**

This command removes a chunk from an **SmsChunkCollection**.

**Parameters:**

[in] **Index** contains the index of the chunk to be remove from the SMS chunk collection.

---



## Property Documentation

### HRESULT Item [get]

This command requests a chunk with a specified index in an **SmsChunkCollection**.

**Parameters:**

[in] **Index** contains the index of the chunk to return.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

## ISmsMessage Interface Reference

---

**SmsMessage** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

---

HRESULT **Send** ()

This command sends an SMS message.

---

HRESULT **Delete** ()

This command deletes an SMS message.

## Properties

LONG	<b>Id</b> [get] This command requests an SMS message ID.
<b>TSmsMessageType</b>	<b>Type</b> [get] This command queries an SMS message type.
<b>TSmsMessageStatus</b>	<b>Status</b> [get] This command queries the status of an SMS message.
<b>TSmsFailureReason</b>	<b>FailureReason</b> [get] This command queries the reason an SMS message failed.
VARIANT_BOOL	<b>IsFailedUnseen</b> [get] This command queries if a failed SMS message was unseen.
VARIANT_BOOL	<b>Seen</b> [set] This command sets an SMS message status read.
LONG	<b>Price</b> [get] This command queries the price of an SMS message.
LONG	<b>PricePrecision</b> [get] This command queries the price precision associated with an SMS message.
BSTR	<b>PriceCurrency</b> [get] This command queries price currency code for an SMS message.
BSTR	<b>ReplyToNumber</b> [get, set] This command queries/sets the reply-to number for an SMS message.
<b>ISmsTargetCollection</b>	<b>Targets</b> [get] This command queries the targets of an SMS message.

**BSTR** **Body** [get, set]  
This command queries/sets the text of an SMS message.

**ISmsChunkCollection** **Chunks** [get]  
This command queries the chunks in an SMS message.

**DATE** **Timestamp** [get]  
This command queries the timestamp of an SMS message.

**BSTR** **TargetNumbers** [get, set]  
This command queries/sets the targets numbers of an SMS message.

---

## Detailed Description

**SmsMessage** is the default interface for this class.

---

## Property Documentation

### **HRESULT Price** [get]

This command queries the price of an SMS message.

**See also:**

[ISmsMessage::PricePrecision](#),  
[ISmsMessage::PriceCurrency](#)

### **HRESULT PriceCurrency** [get]

This command queries price currency code for an SMS message.

**See also:**

[ISmsMessage::Price](#), [ISmsMessage::PricePrecision](#)

### **HRESULT PricePrecision** [get]

This command queries the price precision associated with an SMS message.

**See also:**

[ISmsMessage::Price](#), [ISmsMessage::PriceCurrency](#)

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ISmsMessageCollection Interface Reference

---

**SmsMessageCollection** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Add</b> ([in] <b>ISmsMessage</b> *pItem) This command adds a new message to an <b>SmsMessageCollection</b> .
HRESULT	<b>Remove</b> ([in] long Index) This command removes a message from an <b>SmsMessageCollection</b> .
HRESULT	<b>RemoveAll</b> (void) This command removes all messages from an <b>SmsMessageCollection</b> .

## Properties

LONG	<b>Count</b> [get]	This command queries the number of messages in an <b>SmsMessageCollection</b> .
<b>ISmsMessage</b>	<b>Item</b> ([in] LONG Index) [get]	This command requests a message with a specified index in an <b>SmsMessageCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command requests the enumerator object of an <b>SmsMessageCollection</b> .

## Detailed Description

**SmsMessageCollection** is the default interface for this class.

**See also:**

**ISkype::Smss, ISkype::MissedSmss**

---

## Member Function Documentation

### **HRESULT Add ( [in] ISmsMessage \* pItem )**

This command adds a new message to an **SmsMessageCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer to an **SmsMessage** object.

### **HRESULT Remove ( [in] long Index )**

This command removes a message from an **SmsMessageCollection**.

**Parameters:**

[in] **Index** contains the index of the SMS message to be removed.

## Property Documentation

### HRESULT Item [get]

This command requests a message with a specified index in an **SmsMessageCollection**.

**Parameters:**

[in] **Index** contains the index of the message to be returned.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties



# ISmsTarget Interface Reference

---

**SmsTarget** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

<b>TSmsTargetStatus</b>	<b>Status</b> [get] This command queries the status of an SMS message target.
<b>BSTR</b>	<b>Number</b> [get] This command queries the target number of an SMS message.
<b>ISmsMessage</b>	<b>Message</b> [get] This command queries the <b>SmsMessage</b> related to the target.

---

## Detailed Description

**SmsTarget** is the default interface for this class.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# ISmsTargetCollection Interface Reference

---

**SmsTargetCollection** is the default interface for this class. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **ISmsTarget** \*pItem)

This command adds a target to an **SmsTargetCollection**.

HRESULT **Remove** ([in] long Index)

This command removes a target from an **SmsTargetCollection**.

HRESULT **RemoveAll** (void)

This command removes all targets from an **SmsTargetCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of targets in an <b>SmsTargetCollection</b> .
<b>ISmsTarget</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries a target with a specified index in an <b>SmsTargetCollection</b> .
IUnknown	<b>_NewEnum</b> [get]	This command queries the enumerator object of an <b>SmsTargetCollection</b> .

## Detailed Description

**SmsTargetCollection** is the default interface for this class.

---



## Member Function Documentation

### **HRESULT Add ( [in] ISmsTarget \* pItem )**

This command adds a target to an **SmsTargetCollection**.

**Parameters:**

[in] **pItem** contains the interface pointer to the new **SmsTarget** object.

### **HRESULT Remove ( [in] long Index )**

This command removes a target from an **SmsTargetCollection**.

**Parameters:**

[in] **Index** contains the index of the target to remove.

---

## Property Documentation

### HRESULT Item [get]

This command queries a target with a specified index in an **SmsTargetCollection**.

**Parameters:**

[in] **Index** contains the index of the target to return.

---

The documentation for this interface was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

Properties

# IUser Interface Reference

---

**User** class default interface. This interface enables you to read user properties. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Properties

BSTR	<b>Handle</b> [get, set]	This command queries/sets the Skype name of the user.
BSTR	<b>FullName</b> [get]	This command queries the full name of the user.
DATE	<b>Birthday</b> [get]	This command queries the birthday of the user.
<b>TUserSex</b>	<b>Sex</b> [get]	This command queries the sex of the user.
BSTR	<b>Country</b> [get]	This command queries the country of the user.
BSTR	<b>Province</b> [get]	This command queries the province of the user.
BSTR	<b>City</b> [get]	This command queries the city of the user.
BSTR	<b>PhoneHome</b> [get]	This command queries the home telephone number of the user.
BSTR	<b>PhoneOffice</b> [get]	This command queries the office telephone number of the user.
BSTR	<b>PhoneMobile</b> [get]	This command queries the mobile phone number of the user.
BSTR	<b>Homepage</b> [get]	This command queries the homepage url of the user.
BSTR	<b>About</b> [get]	This command queries the about text for the user.
VARIANT_BOOL	<b>HasCallEquipment</b> [get]	This command queries if a user has call equipment. In the current implementation, this

command always returns a value of true.

**TBuddyStatus** **BuddyStatus** [get, set]  
This command queries/sets user buddy status.

VARIANT\_BOOL **IsAuthorized** [get, set]  
This command queries/sets user authorization status.

VARIANT\_BOOL **IsBlocked** [get, set]  
This command queries/sets whether a user is blocked or not.

BSTR **DisplayName** [get, set]  
This command queries/sets the display name for a user.

**TOnlineStatus** **OnlineStatus** [get]  
This command queries the online status of a user.

DATE **LastOnline** [get]  
This command queries the time when a user was last online.

BSTR **CountryCode** [get]  
This command queries the ISO country code of a user.

BSTR **ReceivedAuthRequest** [get]  
This command queries the authorization request text of a user.

BSTR **SpeedDial** [get, set]  
This command queries/sets the speed dial code of a user.

VARIANT\_BOOL **CanLeaveVoicemail** [get]  
This command queries if it is possible to send voicemail to a user.

BSTR **MoodText** [get]  
This command queries the mood text of a user.

BSTR **Aliases** [get]  
This command queries the alias text of a user.

LONG **Timezone** [get]  
This command queries the timezone of a user.

VARIANT_BOOL	<b>IsCallForwardActive</b> [get]	This command queries the call forwarding status of a user.
BSTR	<b>Language</b> [get]	This command queries the language of the user.
BSTR	<b>LanguageCode</b> [get]	This command queries the ISO language code of a user.
VARIANT_BOOL	<b>IsVideoCapable</b> [get]	This command queries if a user has video capability.
VARIANT_BOOL	<b>IsSkypeOutContact</b> [get]	This command queries whether a user is a SkypeOut contact.
LONG	<b>NumberOfAuthBuddies</b> [get]	This command queries the number of authenticated buddies of the contact.
BSTR	<b>RichMoodText</b> [get]	This property returns user rich mood text.
VARIANT_BOOL	<b>IsVoicemailCapable</b> [get]	This property returns user voicemail capability status.

---

## Detailed Description

**User** class default interface. This interface enables you to read user properties.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IUserCollection Interface Reference

---

**UserCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Add** ([in] **IUser** \*pUser)

This command adds a new item to a **UserCollection**.

HRESULT **Remove** ([in] long Index)

This command removes an item from a **UserCollection**.

HRESULT **RemoveAll** (void)

This command removes all items from a **UserCollection**.

## Properties

LONG	<b>Count</b> [get]	This command queries the number of users in a <b>UserCollection</b> .
<b>IUser</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item in a <b>UserCollection</b> with a specified index value.
IUnknown	<b>_NewEnum</b> [get]	This command queries the enumerator object for a <b>UserCollection</b> .

---

## Detailed Description

**UserCollection** class default interface.

---

## Member Function Documentation

### **HRESULT Add ( [in] IUser \* pUser )**

This command adds a new item to a **UserCollection**.

**Parameters:**

[in] **pUser** receives the interface pointer for the **User** object.

### **HRESULT Remove ( [in] long Index )**

This command removes an item from a **UserCollection**.

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item in a **UserCollection** with a specified index value.

**Parameters:**

[in] **Index** contains the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)



# IVoicemail Interface Reference

---

**Voicemail** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT **Open** ()

This command opens and plays voicemail.

HRESULT **StartPlayback** ()

This command starts playing downloaded voicemail.

HRESULT **StopPlayback** ()

This command stops playing downloaded voicemail.

HRESULT **Upload** ()

This command uploads recorded voicemail from a local computer to the voicemail server.

HRESULT **Download** ()

This command downloads a voicemail object from the voicemail server to a local computer.

HRESULT **StartRecording** ()

This command stops playing a voicemail greeting and starts recording a voicemail message.

HRESULT **StopRecording** ()

This command ends the recording of a voicemail message.

HRESULT **Delete** ()

This command deletes a voicemail object.

HRESULT **StartPlaybackInCall** ()

This command starts playing downloaded voicemail during a call.

HRESULT **SetUnplayed** ()

This command changes the status of a voicemail from played to unplayed.

## Properties

<b>TVoicemailType</b>	<b>Type</b> [get] This command queries the voicemail type.
BSTR	<b>PartnerHandle</b> [get] This command queries the SkypeName of the user receiving a voicemail.
BSTR	<b>PartnerDisplayName</b> [get] This command queries the display name of the user receiving the voicemail.
<b>TVoicemailStatus</b>	<b>Status</b> [get] This command queries the voicemail status.
<b>TVoicemailFailureReason</b>	<b>FailureReason</b> [get] This command queries the reason a voicemail failed.
DATE	<b>Timestamp</b> [get] This command queries the timestamp of a voicemail.
LONG	<b>Duration</b> [get] This command queries the duration of a voicemail.
LONG	<b>AllowedDuration</b> [get] This command queries the permitted duration of a voicemail.
LONG	<b>Id</b> [get] This command queries a voicemail ID.

## Detailed Description

**Voicemail** class default interface.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

[Public Member Functions](#) |  
[Properties](#)

# IVoicemailCollection Interface Reference

---

**VoicemailCollection** class default interface. [More...](#)

```
import "Skype4COM.idl";
```

List of all members.

## Public Member Functions

HRESULT	<b>Add</b> ([in] <b>IVoicemail</b> *pItem) This command adds a new item to a <b>VoicemailCollection</b> .
HRESULT	<b>Remove</b> ([in] long Index) This command removes an item from a <b>VoicemailCollection</b> .
HRESULT	<b>RemoveAll</b> (void) This command removes all items from a <b>VoicemailCollection</b> .

## Properties

LONG	<b>Count</b> [get]	This command returns the number of voicemails in a <b>VoicemailCollection</b> .
<b>IVoicemail</b>	<b>Item</b> ([in] LONG Index) [get]	This command queries an item in a <b>VoicemailCollection</b> with a specified index.
IUnknown	<b>_NewEnum</b> [get]	This command queries a <b>VoicemailCollection</b> enumerator object.

---



## Detailed Description

**VoicemailCollection** class default interface.

Use the **VoicemailCollection** to manage **Voicemail** objects.

---

## Member Function Documentation

### HRESULT Add ( [in] **IVoicemail** \* **pItem** )

This command adds a new item to a **VoicemailCollection** .

**Parameters:**

[in] **pItem** receives the interface pointer for the **Voicemail** object.

### HRESULT Remove ( [in] long **Index** )

This command removes an item from a **VoicemailCollection** .

**Parameters:**

[in] **Index** contains the item to be removed.

---

## Property Documentation

### HRESULT Item [get]

This command queries an item in a **VoicemailCollection** with a specified index.

**Parameters:**

[in] **Index** contains the index of the item to be returned.

---

The documentation for this interface was generated from the following file:

- [Skype4COM.idl](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Participant</a>		

## Participant Class Reference

---

This class manages participants in a conference call. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages participants in a conference call.

**See also:**

[IParticipant](#), [IParticipantCollection](#), [ICall::Participants](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">ParticipantCollection</a>		

## ParticipantCollection Class Reference

---

This class manages collections of participants in a conference call.

[More...](#)

```
import "Skype4COM.idl";
```

---



## Detailed Description

This class manages collections of participants in a conference call.

Use the [ParticipantCollection](#) class to manage collections of participants in conference calls. To query a collection class use **For Each .. Next** statements.

**See also:**

[IParticipant](#), [IParticipantCollection](#), [ICall::Participants](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">PluginEvent</a>		

## PluginEvent Class Reference

---

This class manages plug-in event. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages plug-in event.

**See also:**

[IPluginEvent](#), [\\_ISkypeEvents::PluginEventClicked](#),  
[IPluginMenuItem](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">PluginMenuItem</a>		

## PluginMenuItem Class Reference

---

Plugin menu item. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Plugin menu item.

**See also:**

[IPluginMenuItem](#), [\\_ISkypeEvents::PluginMenuItemClicked](#),  
[IPluginEvent](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Profile</a>		



## Profile Class Reference

---

This class manages the user profile. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages the user profile.

**See also:**  
[IProfile](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Settings</a>		

## Settings Class Reference

---

This class manages the **Skype** API general parameters. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages the **Skype** API general parameters.

**See also:**  
[ISettings](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Skype</a>		

# Skype Class Reference

---

Skype API COM wrapper. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Skype API COM wrapper.

**See also:**

[ISkype](#), [\\_ISkypeEvents](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsChunk</a>		

## SmsChunk Class Reference

---

SMS message chunk. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

SMS message chunk.

**See also:**

[ISmsMessage](#), [ISmsChunk](#), [ISmsChunkCollection](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsChunkCollection</a>		

## SmsChunkCollection Class Reference

---

Collection class for **SmsChunk** objects. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Collection class for **SmsChunk** objects.

**See also:**

**[ISmsMessage](#), [ISmsChunk](#), [ISmsChunkCollection](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsMessage</a>		

## SmsMessage Class Reference

---

SMS message. [More...](#)

```
import "Skype4COM.idl";
```

---



## Detailed Description

SMS message.

**See also:**

[ISmsMessage](#), [ISmsMessageCollection](#), [ISmsTarget](#),  
[ISmsChunk](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsMessageCollection</a>		

## SmsMessageCollection Class Reference

---

Collection class for SMS messages. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Collection class for SMS messages.

**See also:**

[ISmsMessage](#), [ISmsMessageCollection](#), [ISmsTarget](#),  
[ISmsChunk](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsTarget</a>		

## SmsTarget Class Reference

---

SMS message target. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

SMS message target.

**See also:**

[ISmsMessage](#), [ISmsTarget](#), [ISmsTargetCollection](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">SmsTargetCollection</a>		



# SmsTargetCollection Class Reference

---

Collection class for SMS targets. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

Collection class for SMS targets.

**See also:**

[ISmsMessage](#), [ISmsTarget](#), [ISmsTargetCollection](#)

---

The documentation for this class was generated from the following file:

- [Skype4COM.idl](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">User</a>		

## User Class Reference

---

This class manages user properties. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages user properties.

The **User** class provides access to user properties.

### Example:

```
'// Get current user object
oUser = oSkype.CurrentUser

'// Get full name
oHandle = oUser.Fullname

'// Get about text
oAbout = oUser.About
```

### See also:

**IUser**, **IUserCollection**, **ISkype::Friends**,  
**ISkype::SearchForUsers**, **IChat::ActiveMembers**,  
**IChat::Adder**, **IChat::Members**, **IChat::Posters**

---

The documentation for this class was generated from the following file:

- **Skype4COM.idl**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">UserCollection</a>		

# UserCollection Class Reference

---

**User** collection. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

**User** collection.

The **UserCollection** class returns **User** objects. To query items in a collection, use **For Each .. Next** statements.

### Example:

```
'// To query contacts:  
For Each oUser In oSkype.Friends  
Next  
  
'// To search for users:  
For Each oUser In oSkype.Users("peeter")  
Next
```

### See also:

**IUser**, **IUserCollection**, **ISkype::Friends**,  
**ISkype::SearchForUsers**, **IChat::ActiveMembers**,  
**IChat::Adder**, **IChat::Members**, **IChat::Posters**

---

The documentation for this class was generated from the following file:

- **Skype4COM.idl**



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">Voicemail</a>		

## Voicemail Class Reference

---

This class manages **Skype** voicemail. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

This class manages **Skype** voicemail.

**See also:**

**[IVoicemail](#), [IVoicemailCollection](#)**

---

The documentation for this class was generated from the following file:

- **[Skype4COM.idl](#)**

---

Copyright  2006 **Skype Limited**. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">SKYPE4COMLib</a>	<a href="#">VoicemailCollection</a>		

# VoicemailCollection Class Reference

---

**Voicemail** collection. [More...](#)

```
import "Skype4COM.idl";
```

---

## Detailed Description

**Voicemail** collection.

Use this class to return collections of **Voicemail** objects. To access items in a **VoicemailCollection**, use **For Each .. Next** statements.

**See also:**

**IVoicemail, IVoicemailCollection**

---

The documentation for this class was generated from the following file:

- **Skype4COM.idl**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# Class Index

A | C | G | I | P | S | U | V | \_

## A

**Application**  
(SKYPE4COMLib)

**ApplicationStream**  
(SKYPE4COMLib)

**ApplicationStreamCollection**  
(SKYPE4COMLib)

## C

**Call** (SKYPE4COMLib)

**CallChannel**  
(SKYPE4COMLib)

**CallChannelCollection**  
(SKYPE4COMLib)

**CallChannelManager**  
(SKYPE4COMLib)

**CallChannelMessage**  
(SKYPE4COMLib)

**CallCollection**  
(SKYPE4COMLib)

**Chat** (SKYPE4COMLib)

**ChatCollection**  
(SKYPE4COMLib)

**ChatMember**  
(SKYPE4COMLib)

**ChatMemberCollection**  
(SKYPE4COMLib)

**ChatMessage**

**Conference**

(SKYPE4COMLib)

**ConferenceCollection**

(SKYPE4COMLib)

**Conversion**

(SKYPE4COMLib)

## G

**Group** (SKYPE4COMLib)

**GroupCollection**

(SKYPE4COMLib)

## I

**IApplication**

**IConf**

**IApplicationStream**

**IApplicationStreamCollection**

**ICall**

**IFileT**

**ICallChannel**

**ICallChannelCollection**

**IGi**

**ICallChannelManager**

**ICallChannelMessage**

**IPart**



(SKYPE4COMLib)

**ICallCollection**

I

**ChatMessageCollection**

(SKYPE4COMLib)

**IChat**

IPI

**Client** (SKYPE4COMLib)

**IChatCollection**

**Command**

(SKYPE4COMLib)

A | C | G | I | P | S | U | V | \_

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- \_ -

- [\\_NewEnum](#) : [IUserCollection](#) , [ICallChannelCollection](#) , [IFileTransferCollection](#) , [ISmsMessageCollection](#) , [ISmsTargetCollection](#) , [ISmsChunkCollection](#) , [IGroupCollection](#) , [IApplicationStreamCollection](#) , [IVoicemailCollection](#) , [IConferenceCollection](#) , [IChatCollection](#) , [IChatMemberCollection](#) , [IChatMessageCollection](#) , [ICallCollection](#) , [IParticipantCollection](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			
<a href="#">File List</a>				

## File List

---

Here is a list of all documented files with brief descriptions:

**Skype4COM.idl** [\[code\]](#)

---

Copyright  2006 [Skype Limited](#). All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Examples

---

Here is a list of all examples:

- [Answer.vbs](#)
- [Attach.vbs](#)
- [Buttons.vbs](#)
- [Call.php](#)
- [Call.vbs](#)
- [Chats.vbs](#)
- [Chats2.vbs](#)
- [Client.vbs](#)
- [Command.vbs](#)
- [Conference.vbs](#)
- [Conferences.vbs](#)
- [FileTransfers.vbs](#)
- [Groupchat.vbs](#)
- [Groups.vbs](#)
- [Join.vbs](#)
- [MessageStatus.js](#)
- [MessageStatus.vbs](#)
- [MissedMessages.js](#)
- [MissedMessages.vbs](#)
- [Plugins.vbs](#)
- [Profile.vbs](#)
- [Response.vbs](#)
- [Search.vbs](#)
- [Settings.vbs](#)
- [Sms.vbs](#)
- [Smss.vbs](#)
- [VoicemailServer.cs](#)
- [VoicemailServer.vbs](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Answer.vbs

---

This script sets up automatic answering of incoming calls.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Connect to the Skype API:
oSkype.Attach

'// Run an infinite loop to ensure the connection
remains active:
Do While True
    WScript.Sleep(60000)
Loop

'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub

'// The CallStatus event handler monitors call sta
tus and if the status is "ringing" and it is an in
```



coming call, it attempts to answer the call:

```
Public Sub Skype_CallStatus(ByRef aCall, ByVal aStatus)
    WScript.Echo ">Call " & aCall.Id & " status " &
    aStatus & " " & oSkype.Convert.CallStatusToText(a
    Status)
    If oSkype.Convert.TextToCallStatus("RINGING") =
    aStatus And _
        (oSkype.Convert.TextToCallType("INCOMING_P2P")
    = aCall.Type Or _
        oSkype.Convert.TextToCallType("INCOMING_PSTN"
    ) = aCall.Type) Then
        WScript.Echo "Answering call from " & aCall.P
    artnerHandle
        If aCall.TargetIdentity <> "" Then
            WScript.Echo "Call target identity: " & aCa
    ll.TargetIdentity
        End If
        aCall.Answer()
    End If
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Attach.vbs

This script listens for connections on the Skype API, connects when available, and holds the skype name of the user when a connection is successful.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Connect to the Skype API with protocol version
5 (the default version):
oSkype.Attach 5

'// When attached to the API, send repeated comman
ds to the API to test if they succeed:
Do While True
    WScript.Sleep(1000)
    If oSkype.AttachmentStatus = oSkype.Convert.Text
ToAttachmentStatus("SUCCESS") Then
        WScript.Stdout.Write "PING.."
        oSkype.SendCommand(oSkype.Command(0, "PING", "P
ONG", TRUE))
        'oSkype.SendCommand(oSkype.Command(0, "PING"))
        WScript.Stdout.Write "ok"
        WScript.Echo ""
    End If
Loop
```

```
'// The AttachmentStatus event handler monitors attachment status and returns the Skype name of the user when the status is "success":
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Convert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then oSkype.Attach() End If
    If aStatus = oSkype.Convert.TextToAttachmentStatus("SUCCESS") Then WScript.Echo "Current user is " & oSkype.CurrentUserHandle End If
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Buttons.vbs

---

This script sends key press events to the Skype client.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Connect to the Skype API:
oSkype.Attach

'// Bring the Skype client into focus and open the
Contacts tab:
oSkype.Client.Focus
oSkype.Client.OpenContactsTab()

'// Emulate typing "ECHO123" into the address bar
of the Skype client (appears in the address bar or
the Skype client):
oSkype.Client.ButtonPressed "E"
oSkype.Client.ButtonReleased "E"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "C"
oSkype.Client.ButtonReleased "C"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "H"
oSkype.Client.ButtonReleased "H"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "O"
```

```

oSkype.Client.ButtonReleased "0"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "1"
oSkype.Client.ButtonReleased "1"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "2"
oSkype.Client.ButtonReleased "2"
WScript.Sleep(1000)
oSkype.Client.ButtonPressed "3"
oSkype.Client.ButtonReleased "3"
WScript.Sleep(1000)

'// Emulate pressing the Green (YES) button and th
e client starts a call:
oSkype.Client.ButtonPressed "YES"
oSkype.Client.ButtonReleased "YES"
WScript.Sleep(10000)

'// Emulate pressing the Red (NO) button and the c
lient ends the call:
oSkype.Client.ButtonPressed "NO"
oSkype.Client.ButtonReleased "NO"

'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub

```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			



# Call.php

This PHP script places a call through the Skype API, and monitors status while the call is in progress.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
<?php

// Event sink:
class _ISkypeEvents {

    function AttachmentStatus($status) {
        echo ">Attachment status $status\n";
    }
    function CallStatus($call, $status) {
        echo ">Call $call->id status $status\n";
    }
}

// Create a Skype4COM object:
$skype = new COM("Skype4COM.Skype");

// Create a sink object:
$sink =& new _ISkypeEvents ();
$sink->convert = $skype->convert();

// Connect to the sink:
com_event_sink($skype, $sink, "_ISkypeEvents");

// Create a conversion object:
$convert = $skype->convert;
$convert->language = "en";

// Start the Skype client, minimized and with no s
```

```

plash screen:
if (!$skype->client()->isRunning()) {
    $skype->client()->start(true, true);
}

// If the user status is not "online", change user
status to "online":
if ($skype->currentUserStatus() == $convert->textT
oUserStatus("OFFLINE")) {
    $skype->changeUserStatus($convert->textToUserSta
tus("ONLINE"));
}

// Create a user object
$user = $skype->user("echo123");
echo "User " . $user->handle . " online status is "
    . $convert->onlineStatusToText($user->onlineStatu
s) . "\n";

// Place a call
$call = $skype->PlaceCall($user->handle);

// Wait for the call to be "in progress" and repor
t an error if:
while ($call->status <> $convert->textToCallStatus(
"INPROGRESS")) {
    if ($call->status == $convert->textToCallStatus(
"FAILED") ||
        $call->status == $convert->textToCallStatus("R
EFUSED") ||
        $call->status == $convert->textToCallStatus("C
ANCELLED") ||
        $call->status == $convert->textToCallStatus("F
INISHED") ||
        $call->status == $convert->textToCallStatus("B
USY"))
        die ("Call status " . $convert->callStatusToTe

```

```
xt($call->status));
    else
        com_message_pump (500);
}

// Send dtmf tones:
com_message_pump (10000);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "0";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "1";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "2";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "3";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "4";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "5";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "6";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "7";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "8";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("I
NPROGRESS")) $call->dtmf = "9";
com_message_pump (500);
```

```
if ($call->status == $convert->textToCallStatus("INPROGRESS")) $call->dtmf = "#";
com_message_pump (500);
if ($call->status == $convert->textToCallStatus("INPROGRESS")) $call->dtmf = "*";

// Finish the call:
if ($call->status <> $convert->textToCallStatus("FINISHED")) $call->finish();

//Sleep:
com_message_pump (1000);

?>
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Call.vbs

---

This script places a call through the Skype API, and monitors status while the call is in progress.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Declare the following Skype constants:
cUserStatus_Offline = oSkype.Convert.TextToUserSta
tus("OFFLINE")
cUserStatus_Online = oSkype.Convert.TextToUserStat
us("ONLINE")
cCallStatus_Ringing = oSkype.Convert.TextToCallSta
tus("RINGING")
cCallStatus_Inprogress = oSkype.Convert.TextToCall
Status("INPROGRESS")
cCallStatus_Failed = oSkype.Convert.TextToCallStat
us("FAILED")
cCallStatus_Refused = oSkype.Convert.TextToCallSta
tus("REFUSED")
cCallStatus_Cancelled = oSkype.Convert.TextToCallS
tatus("CANCELLED")
cCallStatus_Finished = oSkype.Convert.TextToCallSt
atus("FINISHED")
cCallStatus_Busy = oSkype.Convert.TextToCallStatus
("BUSY")
cAttachmentStatus_Available = oSkype.Convert.TextT
```

```

oAttachmentStatus("AVAILABLE")

'// The PlaceCall command will fail if the user is
  offline. To avoid failure, check user status and
  change to online if necessary:
If cUserStatus_Offline = oSkype.CurrentUserStatus
Then oSkype.ChangeUserStatus(cUserStatus_Online) E
nd If

'// Create a user object:
Set oUser = oSkype.User("echo123")
WScript.Echo "User " & oUser.Handle & " online sta
tus is " & oSkype.Convert.OnlineStatusToText(oUser
.OnlineStatus)

'// Place a call:
Set oCall = oSkype.PlaceCall(oUser.Handle)

'// Wait until the call is in progress and return
the relevant error if any other status occurs:
Do While oCall.Status <> cCallStatus_Inprogress
  If oCall.Status = cCallStatus_Failed Or _
    oCall.Status = cCallStatus_Refused Or _
    oCall.Status = cCallStatus_Cancelled Or _
    oCall.Status = cCallStatus_Finished Or _
    oCall.Status = cCallStatus_Busy Then
    Err.Raise vbObjectError + 1, "", "Call status:
" & oSkype.Convert.CallStatusToText(oCall.Status)

  End If
  WScript.Sleep(500)
Loop

'// Check sending DTMF tones for use with interact
ive voice response (IVR) applications:
WScript.Sleep(10000)
If cCallStatus_InProgress = oCall.Status Then oCal

```

```
l.DTMF = "0" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "1" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "2" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "3" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "4" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "5" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "6" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "7" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "8" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "9" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "#" End If
WScript.Sleep(500)
If cCallStatus_InProgress = oCall.Status Then oCal
l.DTMF = "*" End If
WScript.Sleep(500)

'// Finish the call:
```



```
If oCall.Status <> cCallStatus_Finished Then oCall
.Finish() End If
```

```
'// The AttachmentStatus event handler monitors at
achment status and automatically attempts to reat
tach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Con
vert.AttachmentStatusToText(aStatus)
    If aStatus = cAttachmentStatus_Available Then oS
kype.Attach() End If
End Sub
```

```
'// If the call status is "failed", the CallStatus
event handler returns the relevant failure reason
as text:
```

```
Public Sub Skype_CallStatus(ByRef aCall, ByVal aSt
atus)
    WScript.Echo ">Call " & aCall.Id & " status " &
aStatus & " " & oSkype.Convert.CallStatusToText(a
Status)
    If cCallStatus_Failed = aStatus Then
        WScript.Echo ">Failure reason:" & oSkype.Conve
rt.CallFailureReasonToText(aCall.FailureReason)
    End If
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Chats.vbs

---

This script queries chats in chat history.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Query the total number of messages in history:
WScript.Echo "Total message count: " & oSkype.Mess
ages.Count

'// Query the timestamp, name, and display name of
all chats in history:
WScript.Echo "All chats:"
For Each oChat In oSkype.Chats
    WScript.Echo oChat.Timestamp & " " & oChat.Name
    & " " & oChat.FriendlyName
Next

'// Query the timestamp, name, and display name of
all active chats:
WScript.Echo "Active chats:"
For Each oChat In oSkype.ActiveChats
    WScript.Echo oChat.Timestamp & " " & oChat.Name
    & " " & oChat.FriendlyName
Next

'// Query the timestamp, name, and display name of
```

```

    all missed chats:
WScript.Echo "Missed chats:"
For Each oChat In oSkype.MissedChats
    WScript.Echo oChat.Timestamp & " " & oChat.Name
    & " " & oChat.FriendlyName
Next

'// Query the timestamp, name, and display name of
all recent chats:
WScript.Echo "Recent chats:"
For Each oChat In oSkype.RecentChats
    WScript.Echo oChat.Timestamp & " " & oChat.Name
    & " " & oChat.FriendlyName
Next

'// Query the timestamp, name, and display name of
all bookmarked chats:
WScript.Echo "Bookmarked chats:"
For Each oChat In oSkype.BookmarkedChats
    WScript.Echo oChat.Timestamp & " " & oChat.Name
    & " " & oChat.FriendlyName
Next

'// The AttachmentStatus event handler monitors at
achment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub

```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Chats2.vbs

An improved chats example to illustrate new chat properties.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create Skype object
Set oSkype = CreateObject("Skype4COM.Skype")

'// Query properties of all recent chats:
WScript.Echo "Recent chats:"
For Each oChat In oSkype.RecentChats
    displayChatProperties(oChat)
Next

'// Query properties of all bookmarked chats:
WScript.Echo "Bookmarked chats:"
For Each oChat In oSkype.BookmarkedChats
    displayChatProperties(oChat)
Next

'// Displays chat properties
Function displayChatProperties(aChat)
    WScript.Echo vbCrLf & "Label: " & aChat.Friendly
Name
    WScript.Echo "Created at: " & aChat.Timestamp
    WScript.Echo "Last activity at: " & aChat.Activi
tyTimestamp
    WScript.Echo "Id: " & aChat.Name
    WScript.Echo "Blob: " & aChat.Blob
    WScript.Echo "Description: " & aChat.Description
    WScript.Echo "Guidelines: " & aChat.Guidelines
    WScript.Echo "My role: " & chatMemberRoleToText(
aChat.MyRole)
    WScript.Echo "My status: " & chatStatusToText(aC
```

```

hat.MyStatus)
    WScript.Echo "Type: " & chatTypeToText(aChat.Type)
)
    If aChat.Type = chatTypeDialog Or aChat.Type = chatTypeLegacyDialog Then
        WScript.Echo "Dialog partner: " & aChat.DialogPartner
    End If
    If aChat.Members.Count > 0 Then
        WScript.Stdout.Write "Members:"
        For Each oMember In aChat.Members
            WScript.Stdout.Write " " & oMember.Handle
        Next
        WScript.Echo ""
    End If
    If aChat.Applicants.Count > 0 Then
        WScript.Stdout.Write "Applicants:"
        For Each oApplicant In aChat.Applicants
            WScript.Stdout.Write " " & oApplicant.Handle
        Next
        WScript.Echo ""
    End If
End Function

```

'// Converts chat types to plaintext

```

Function chatTypeToText(aType)
    Const chatTypeUnknown = -1
    Const chatTypeDialog = 0
    Const chatTypeLegacyDialog = 1
    Const chatTypeLegacyUnsubscribed = 2
    Const chatTypeMultiChat = 3
    Const chatTypeSharedGroup = 4
    Select Case aType
        Case chatTypeUnknown
            chatTypeToText = "Unknown"
        Case chatTypeDialog
            chatTypeToText = "Dialog"

```

```

Case chatTypeLegacyDialog
    chatTypeToText = "Legacy Dialog"
Case chatTypeLegacyUnsubscribed
    chatTypeToText = "Unsubscribed Legacy Dialog"
"
Case chatTypeMultiChat
    chatTypeToText = "Multichat"
Case chatTypeSharedGroup
    chatTypeToText = "Shared Group"
End Select
End Function

'// Converts chat roles to plaintext
Function chatMemberRoleToText(aRole)
Const chatMemberRoleUnknown = -1
Const chatMemberRoleCreator = 0
Const chatMemberRoleMaster = 1
Const chatMemberRoleHelper = 2
Const chatMemberRoleUser = 3
Const chatMemberRoleListener = 4
Const chatMemberRoleApplicant = 5
Select Case aRole
Case chatMemberRoleUnknown
    chatMemberRoleToText = "Unknown"
Case chatMemberRoleCreator
    chatMemberRoleToText = "Creator"
Case chatMemberRoleMaster
    chatMemberRoleToText = "Master"
Case chatMemberRoleHelper
    chatMemberRoleToText = "Helper"
Case chatMemberRoleUser
    chatMemberRoleToText = "User"
Case chatMemberRoleListener
    chatMemberRoleToText = "Listener"
Case chatMemberRoleApplicant
    chatMemberRoleToText = "Applicant"
End Select

```



End Function

```
'// Converts chat status to plaintext
Function chatStatusToText(aStatus)
    Const chatStatusUnknown = -1
    Const chatStatusConnecting = 0
    Const chatStatusWaitingRemoteAccept = 1
    Const chatStatusAcceptRequired = 2
    Const chatStatusPasswordRequired = 3
    Const chatStatusSubscribed = 4
    Const chatStatusUnsubscribed = 5
    Const chatStatusDisbanded = 6
    Const chatStatusQueuedBecauseChatIsFull = 7
    Const chatStatusApplicationDenied = 8
    Const chatStatusKicked = 9
    Const chatStatusBanned = 10
    Const chatStatusRetryConnecting = 11
    Select Case aStatus
        Case chatStatusUnknown
            chatStatusToText = "Unknown"
        Case chatStatusConnecting
            chatStatusToText = "Connecting"
        Case chatStatusWaitingRemoteAccept
            chatStatusToText = "Waiting for Remote Accept"
        Case chatStatusAcceptRequired
            chatStatusToText = "Accept Required"
        Case chatStatusPasswordRequired
            chatStatusToText = "Password Required"
        Case chatStatusSubscribed
            chatStatusToText = "Subscribed"
        Case chatStatusUnsubscribed
            chatStatusToText = "Unsubscribed"
        Case chatStatusDisbanded
            chatStatusToText = "Disbanded"
        Case chatStatusQueuedBecauseChatIsFull
            chatStatusToText = "Queued"
```

```
Case chatStatusApplicationDenied
    chatStatusToText = "Application Denied"

Case chatStatusKicked
    chatStatusToText = "Kicked"
Case chatStatusBanned
    chatStatusToText = "Banned"
Case chatStatusRetryConnecting
    chatStatusToText = "Reconnecting"
End Select
End Function
```

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Client.vbs

---

This script opens different windows in the Skype client.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Bring the client into focus:
oSkype.Client.Focus
WScript.Sleep(1000)

'// Minimize the client to the system tray:
oSkype.Client.Minimize
WScript.Sleep(1000)

'// Open the Add Contacts window and enter "echo12
3" in the contact field:
oSkype.Client.OpenAddContactDialog "echo123"
WScript.Sleep(1000)

'// Open a Chat window with "echo123" and type "Hi
" in the message pane:
oSkype.Client.OpenMessageDialog "echo123", "Hi"
WScript.Sleep(1000)

'// Transfer a file to "echo123"
'oSkype.Client.OpenFileTransferDialog "echo123", "
c:\temp"
```

```
'WScript.Sleep(1000)

'// Open other windows as required, with the following commands:
oSkype.Client.OpenProfileDialog
WScript.Sleep(1000)
oSkype.Client.OpenUserInfoDialog "echo123"
WScript.Sleep(1000)
oSkype.Client.OpenConferenceDialog
WScript.Sleep(1000)
oSkype.Client.OpenSearchDialog
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("GENERAL")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("PRIVACY")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("NOTIFICATIONS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("SOUNDALERTS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("SOUNDDEVICES")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("HOTKEYS")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("CONNECTION")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("VOICEMAIL")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("CALLFORWARD")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("VIDEO")
WScript.Sleep(1000)
oSkype.Client.OpenOptionsDialog("ADVANCED")
WScript.Sleep(1000)
oSkype.Client.OpenCallHistoryTab
WScript.Sleep(1000)
oSkype.Client.OpenContactsTab
```

```

WScript.Sleep(1000)
oSkype.Client.OpenDialpadTab
WScript.Sleep(1000)
oSkype.Client.OpenLiveTab
WScript.Sleep(1000)
oSkype.Client.OpenSendContactsDialog "echo123"
WScript.Sleep(1000)
oSkype.Client.OpenBlockedUsersDialog
WScript.Sleep(1000)
oSkype.Client.OpenImportContactsWizard
WScript.Sleep(1000)
oSkype.Client.OpenGettingStartedWizard
WScript.Sleep(1000)
oSkype.Client.OpenAuthorizationDialog "echo123"
WScript.Sleep(1000)
oSkype.Client.OpenVideoTestDialog
WScript.Sleep(1000)

'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub

```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Command.vbs

---

This script sends blocking and non-blocking commands to the Skype API.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
","Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Connect to the Skype API:
oSkype.Attach

'// Skype4COM can send two types of command; block
ing commands and non-blocking commands. Blocking i
s defined as a Boolean variant of the command obje
ct and the default value is False, non-blocking. A
blocking command awaits a reply before continuing
. A non-blocking command does not require a reply.

'// Send the following non-blocking commands:
oSkype.SendCommand(oSkype.Command(0, "FOCUS"))
oSkype.SendCommand(oSkype.Command(1, "GET AUDIO_IN
", "AUDIO_IN"))
oSkype.SendCommand(oSkype.Command(2, "GET AUDIO_OU
T", "AUDIO_OUT"))
WScript.Sleep(1000)

'// Send the following blocking commands (note the
```



```
True (blocking) variant for each command):
oSkype.SendCommand(oSkype.Command(3, "GET CURRENTU
SERHANDLE", "CURRENTUSERHANDLE", True))
oSkype.SendCommand(oSkype.Command(4, "GET USER ech
o123 FULLNAME", "USER echo123 FULLNAME", True))
oSkype.SendCommand(oSkype.Command(5, "GET USER ech
o123 DISPLAYNAME", "USER echo123 DISPLAYNAME", Tru
e))
oSkype.SendCommand(oSkype.Command(6, "GET USER ech
o123 COUNTRY", "USER echo123 COUNTRY", True))
WScript.Sleep(1000)
```

```
'//As with command objects, search commands can be
defined as blocking or non-blocking, and the defa
ult value is False, non-blocking. A blocking searc
h requires a response before activities continue.
Following is a blocking search (note the True (blo
cking) variant:
```

```
Set oBlockingSearch = oSkype.Command(8888, "SEARCH
USERS echo123", "USERS ", True)
oSkype.SendCommand(oBlockingSearch)
WScript.Echo "Search Result:" & oBlockingSearch.Re
ply
```

```
'//A non-blocking search does not require a respon
se:
```

```
Set oSearchCommand = oSkype.Command(9999, "SEARCH
USERS john doe", "USERS ")
oSkype.SendCommand(oSearchCommand)
```

```
WScript.Echo "Sleeping ..."
WScript.Sleep(30000)
```

```
'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
```

```
WScript.Echo ">Attachment status " & oSkype.Convert.AttachmentStatusToText(aStatus)
  If aStatus = oSkype.Convert.TextToAttachmentStatus("AVAILABLE") Then oSkype.Attach() End If
End Sub
```

```
'// The reply event handler monitors the non-blocking search results:
```

```
Public Sub Skype_Reply(ByRef aCmd)
  If IsObject(oSearchCommand) Then
    If oSearchCommand.Id = aCmd.Id Then WScript.Echo "Search Finished!" End If
  End If
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Conference.vbs

---

This script creates a conference call.

Copyright  2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// The PlaceCall command will fail if the user is
offline. To avoid failure, check user status and
change to online if necessary:
If oSkype.Convert.TextToUserStatus("OFFLINE") = oS
kype.CurrentUserStatus Then
    oSkype.ChangeUserStatus(oSkype.Convert.TextToUse
rStatus("ONLINE"))
End If

'// To create a conference call, use the PlaceCall
command and add multiple users to the list of tar
gets (to the maximum number of conference users su
pported by the OS environment):
Set oFirstCall = oSkype.PlaceCall("skypeuser1", "s
kypeuser2")
WScript.Sleep(30000)

'// The AttachmentStatus monitors and returns call
status:
Public Sub Skype_CallStatus(ByRef aCall, ByVal aSt
atus)
```

```
WScript.Echo ">Call " & aCall.Id & " status " &
aStatus & " " & oSkype.Convert.CallStatusToText(aS
tatus)
End Sub
```

```
'// The AttachmentStatus event handler monitors at
achment status and automatically attempts to reat
tach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub
```

---

Copyright  2006 Skype Limited. All rights reserved.


# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Conferences.vbs

---

This script lists conference calls hosted by a user, and lists participants in conference calls not hosted by a user.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
","Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Count conference calls hosted by the current u
ser. A conference call hosted by the current user
is represented by multiple call objects which have
a common conference ID (conf_id). For each confer
ence object, return the conf_id, the number of ca
lls, and the number of active calls:
WScript.Echo "There are " & oSkype.Conferences.Cou
nt & " conference calls hosted by " & oSkype.Curre
ntUser.Handle
For Each oConf In oSkype.Conferences
    WScript.Echo "Conference " & oConf.Id & " includ
es " & oConf.Calls.Count & " calls from which " &
oConf.ActiveCalls.Count & " are active."
Next

'// To list conference calls in which the current
user is active but not the host, begin by listing
active calls for a user:
WScript.Echo "There are total " & oSkype.ActiveCal
ls.Count & " active calls."
```

```
'//Conference calls not hosted by the current user
are all calls which have one or more participants
(participants do not include the current user or
the conference host). For each active call, return
the Skype name and display name of each participan
t:
```

```
For Each oCall In oSkype.ActiveCalls
```

```
    If oCall.Participants.Count > 0 Then
        WScript.Echo "Conference call " & oCall.Id & "
with " & oCall.PartnerHandle & " has " & oCall.Pa
rticipants.Count & " participants."
```

```
        For Each oParticipant In oCall.Participants
            WScript.Echo "Participant " & oParticipant.H
andle & " display name is " & oParticipant.Display
Name
```

```
        Next
    Else
        WScript.Echo "Active call " & oCall.Id & " wit
h " & oCall.PartnerHandle & " is not a conference
call."
```

```
    End If
Next
```

```
'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub
```



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# FileTransfers.vbs

---

This script lists file transfers.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
Set oSkype = WScript.CreateObject("Skype4COM.Skype"
, "Skype_")

For Each oTransfer In oSkype.FileTransfers
    FTDetails(oTransfer)
Next

For Each oTransfer In oSkype.ActiveFileTransfers
    FTDetails(oTransfer)
Next

WScript.Sleep(30000)

Public Sub Skype_FileTransferStatusChanged(ByRef a
Transfer, ByVal aStatus)
    WScript.Echo ">File Transfer " & aTransfer.Id &
" status " & FTStatusToText(aStatus)
End Sub

Sub FTDetails(aTransfer)
    WScript.Echo "Id:" & aTransfer.Id & " start:" &
aTransfer.StartTime & " end:" & aTransfer.FinishTi
me
    WScript.Echo " Type:" & FTTypeToText(aTransfer.T
ype)
    WScript.Echo " Status:" & FTStatusToText(aTransf
er.Status)
    WScript.Echo " Partner:" & aTransfer.PartnerHand
le & "(" & aTransfer.PartnerDisplayName & ")"
```

```
WScript.Echo " File:" & aTransfer.FilePath
WScript.Echo " Bytes Transferred:" & aTransfer.BytesTransferred & " bytes"
WScript.Echo " Transfer Speed:" & aTransfer.BytesPerSecond & " bytes/sec"
WScript.Echo " Failure Reason:" & FTFailureReasonToText(aTransfer.FailureReason)
End Sub
```

```
Function FTTypeToText(aType)
```

```
    Const fileTransferTypeIncoming=0
    Const fileTransferTypeOutgoing=1
```

```
    Select Case aType
        Case fileTransferTypeIncoming
            FTTypeToText = "Incoming"
        Case fileTransferTypeOutgoing
            FTTypeToText = "Outgoing"
    End Select
```

```
End Function
```

```
Function FTStatusToText(aStatus)
```

```
    Const fileTransferStatusNew = 0
    Const fileTransferStatusConnecting = 1
    Const fileTransferStatusWaitingForAccept = 2
    Const fileTransferStatusTransferring = 3
    Const fileTransferStatusTransferringOverRelay =
4
    Const fileTransferStatusPaused = 5
    Const fileTransferStatusRemotelyPaused = 6
    Const fileTransferStatusCancelled = 7
    Const fileTransferStatusCompleted = 8
    Const fileTransferStatusFailed = 9
```

```
Select Case aStatus
  Case fileTransferStatusNew
    FTStatusToText = "New"
  Case fileTransferStatusConnecting
    FTStatusToText = "Connecting"
  Case fileTransferStatusWaitingForAccept
    FTStatusToText = "Waiting for Accept"
  Case fileTransferStatusTransferring
    FTStatusToText = "Transferring"
  Case fileTransferStatusTransferringOverRelay
    FTStatusToText = "Transferring over Relay"
  Case fileTransferStatusPaused
    FTStatusToText = "Paused"
  Case fileTransferStatusRemotelyPaused
    FTStatusToText = "Remotely Paused"
  Case fileTransferStatusCancelled
    FTStatusToText = "Cancelled"
  Case fileTransferStatusCompleted
    FTStatusToText = "Completed"
  Case fileTransferStatusFailed
    FTStatusToText = "Failed"
End Select
```

End Function

Function FTFailureReasonToText(aReason)

```
  Const fileTransferFailureReasonSenderNotAuthorized= 1
  Const fileTransferFailureReasonRemotelyCancelled = 2
  Const fileTransferFailureReasonFailedRead = 3
  Const fileTransferFailureReasonFailedRemoteRead = 4
  Const fileTransferFailureReasonFailedWrite = 5
  Const fileTransferFailureReasonFailedRemoteWrite = 6
```

```

Const fileTransferFailureReasonRemoteDoesNotSupportFT = 7
Const fileTransferFailureReasonRemoteOfflineTooLong = 8

Select Case aReason
    Case fileTransferFailureReasonSenderNotAuthorized
        FTFailureReasonToText = "Sender Not Authorized"
    Case fileTransferFailureReasonRemotelyCancelled
        FTFailureReasonToText = "Remotely Cancelled"
    Case fileTransferFailureReasonFailedRead
        FTFailureReasonToText = "Failed Read"
    Case fileTransferFailureReasonFailedRemoteRead
        FTFailureReasonToText = "Remote Read"
    Case fileTransferFailureReasonFailedWrite
        FTFailureReasonToText = "Write"
    Case fileTransferFailureReasonFailedRemoteWrite
        FTFailureReasonToText = "Remote Write"
    Case fileTransferFailureReasonRemoteDoesNotSupportFT
        FTFailureReasonToText = "Remote Does Not Support FT"
    Case fileTransferFailureReasonRemoteOfflineTooLong
        FTFailureReasonToText = "Remote Offline Too Long"

End Select

End Function

```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Groupchat.vbs

This script creates a group chat.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Create a UserCollection object and add the mul
tichat users to this collection:
Set oMembers = CreateObject("Skype4COM.UserCollect
ion")
oMembers.Add(oSkype.User("skypeuser2"))
oMembers.Add(oSkype.User("skypeuser3"))

'// Create a multichat:
Set oChat = oSkype.CreateChatMultiple(oMembers)

'// Open a chat window:
oChat.OpenWindow

'// Set a chat topic:
oChat.Topic = "Skype is cool"

'// Send a chat message:
Set oMsg = oChat.SendMessage("Greetings!")

'// Leave a chat:
oChat.Leave
```

```
'// The AttachmentStatus event handler monitors at  
tachment status and automatically attempts to reat  
tach to the API following loss of connection:  
Public Sub Skype_AttachmentStatus(ByVal aStatus)  
    WScript.Echo ">Attachment status " & oSkype.Conv  
ert.AttachmentStatusToText(aStatus)  
    If aStatus = oSkype.Convert.TextToAttachmentStat  
us("AVAILABLE") Then oSkype.Attach() End If  
End Sub
```

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Groups.vbs

---

This script manages various aspects of contact groups.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Create a custom group:
Set oMyGroup = oSkype.CreateGroup("MyGroup")

'// Add user "echo123" to the group:
oMyGroup.AddUser "echo123"

'// List the users in MyGroup and return the Skype
name of each user:
WScript.Echo "Group ("& oMyGroup.Id &) labelled [
"& oMyGroup.DisplayName &] has " & oMyGroup.Users
.Count & " users."
For Each oUser In oMyGroup.Users
    WScript.Echo oUser.Handle & " ("& oUser.FullName
&)"
Next

'// Remove a user from a group:
oMyGroup.RemoveUser "echo123"

'// Delete a custom group:
oSkype.DeleteGroup oMyGroup.Id
```

```

'// List all groups:
WScript.Echo "There are total " & oSkype.Groups.Count & " groups (" &_
oSkype.CustomGroups.Count & " custom and " & oSkype.HardwiredGroups.Count & " hardwired)" & vbCrLf

'// List all custom groups:
WScript.Echo "Custom groups are:"& vbCrLf
For Each oGroup In oSkype.CustomGroups
    WScript.Echo "Group ("& oGroup.Id &") labelled ["& oGroup.DisplayName &"] has " & oGroup.Users.Count & " users."
    For Each oUser In oGroup.Users
        WScript.Echo oUser.Handle & " ("& oUser.FullName &")"
    Next
    WScript.Echo ""
Next

'// List all hardwired groups and return the Skype name and full name of each member of each group:
WScript.Echo "Hardwired groups are:"& vbCrLf
For Each oGroup In oSkype.HardwiredGroups
    WScript.Echo "Group ("& oGroup.Id &") type of ["& oSkype.Convert.GroupTypeToText(oGroup.Type) &"] has " & oGroup.Users.Count & " users."
    For Each oUser In oGroup.Users
        WScript.Echo oUser.Handle & " ("& oUser.FullName &")"
    Next
    WScript.Echo ""
Next

'// Keep the script running for 60 seconds:
WScript.Sleep(60000)

```

```
'// The GroupVisible event handler returns informa-
tion about whether a group is visible or hidden:
Public Sub Skype_GroupVisible(ByRef aGroup, ByVal
aVisible)
    WScript.Stdout.Write "Group ("& aGroup.Id &") ty-
pe of ["& oSkype.Convert.GroupTypeToText(aGroup.Ty-
pe) &"] labelled ["& aGroup.DisplayName &"]"
    If aVisible Then
        WScript.Echo " is visible."
    Else
        WScript.Echo " is hidden."
    End If
End Sub
```

```
'// The GroupExpanded event handler returns inform-
ation about whether a group is expanded or collaps-
ed:
Public Sub Skype_GroupExpanded(ByRef aGroup, ByVal
aExpanded)
    WScript.Stdout.Write "Group ("& aGroup.Id &") ty-
pe of ["& oSkype.Convert.GroupTypeToText(aGroup.Ty-
pe) &"] labelled ["& aGroup.DisplayName &"]"
    If aExpanded Then
        WScript.Echo " is expanded."
    Else
        WScript.Echo " is collapsed."
    End If
End Sub
```

```
'// The GroupUsers event handler gets the Skypenam-
es of members of a group:
Public Sub Skype_GroupUsers(ByRef aGroup, ByRef aU-
sers)
    WScript.Stdout.Write "Group ("& aGroup.Id &") ty-
pe of ["& oSkype.Convert.GroupTypeToText(aGroup.Ty-
pe) &"] labelled ["& aGroup.DisplayName &"] users"
    Dim oUser
```

```

For Each oUser In aUsers
    WScript.Stdout.Write " " & oUser.Handle
Next
WScript.Echo ""
End Sub

'// Delete a group:
Public Sub Skype_GroupDeleted(ByRef aGroupId)
    WScript.Echo "Group " & aGroupId & " deleted."
End Sub

'// Bring a contact into focus:
Public Sub Skype_ContactsFocused(ByVal aHandle)

    If Len(aHandle) Then
        WScript.Echo "Contact " & aHandle & " focused."
    "
    Else
        WScript.Echo "No contact focused."
    End If
End Sub

'// The AttachmentStatus event handler monitors at
attachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub

```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Join.vbs

---

This script joins callers to form a conference call.

Copyright  2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Declare the following Skype constants:
cUserStatus_Offline = oSkype.Convert.TextToUserSta
tus("OFFLINE")
cUserStatus_Online = oSkype.Convert.TextToUserStat
us("ONLINE")
cCallStatus_Ringing = oSkype.Convert.TextToCallSta
tus("RINGING")
cCallStatus_Inprogress = oSkype.Convert.TextToCall
Status("INPROGRESS")
cCallStatus_Failed = oSkype.Convert.TextToCallStat
us("FAILED")
cCallStatus_Refused = oSkype.Convert.TextToCallSta
tus("REFUSED")
cCallStatus_Cancelled = oSkype.Convert.TextToCallS
tatus("CANCELLED")
cCallStatus_Finished = oSkype.Convert.TextToCallSt
atus("FINISHED")
cCallStatus_Busy = oSkype.Convert.TextToCallStatus
("BUSY")
cCallStatus_LocalHold = oSkype.Convert.TextToCallS
tatus("LOCALHOLD")
```

```

'// The PlaceCall command will fail if the user is
  offline. To avoid failure, check user status and
  change to online if necessary:
If cUserStatus_Offline = oSkype.CurrentUserStatus
Then oSkype.ChangeUserStatus(cUserStatus_Online) E
nd If

'// Place a call (the first of more than one):
Set oCall1 = oSkype.PlaceCall("skypeuser1")
WaitForCallStatus oCall1, cCallStatus_Inprogress

'// Put the first call on hold:
oCall1.Hold
WaitForCallStatus oCall1, cCallStatus_LocalHold

'// Place the second call:
Set oCall2 = oSkype.PlaceCall("skypeuser2")
WaitForCallStatus oCall2, cCallStatus_Inprogress

'// Join these two calls to a conference:
WScript.Echo "Join call " & oCall2.Id & " to call "
  & oCall1.Id
oCall2.Join oCall1.Id

WScript.Sleep(60000)

'// The AttachmentStatus event handler monitors at
  tachment status and automatically attempts to reat
  tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
  WScript.Echo ">Attachment status " & oSkype.Conv
  ert.AttachmentStatusToText(aStatus)
  If aStatus = oSkype.Convert.TextToAttachmentStat
  us("AVAILABLE") Then oSkype.Attach() End If
End Sub

```



```
'// The CallStatus event handler monitors call sta
tus and returns status changes as text:
Public Sub Skype_CallStatus(ByRef oCall, ByVal Sta
tus)
    WScript.Echo ">Call " & oCall.Id & " status " &
Status & " " & oSkype.Convert.CallStatusToText(Sta
tus)
End Sub

'// Waits for call status to change:
Public Sub WaitForCallStatus(ByRef aCall, ByVal aS
tatus)
    Do While aCall.Status <> aStatus
        If aCall.Status = cCallStatus_Failed Or _
            aCall.Status = cCallStatus_Refused Or _
            aCall.Status = cCallStatus_Cancelled Or _
            aCall.Status = cCallStatus_Finished Or _
            aCall.Status = cCallStatus_Busy Then
            Err.Raise vbObjectError + 1, "", "Call " & _
                aCall.Id & " " & oSkype.Convert.CallStatus
ToText(aCall.Status)
        End If
        WScript.Sleep(500)
    Loop
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# MessageStatus.js

---

This script logs chat text messages.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
// Create a Skype4COM object:
var oSkype = new ActiveXObject("Skype4COM.Skype");

// Connect events:
WScript.ConnectObject(oSkype, "Skype_");

// Connect to Skype API:
oSkype.Attach();

// Loop and sleep:
while (true) {
    WScript.Sleep(1000);
}

// Chat message status events:
function Skype_MessageStatus(aMsg, aStatus)
{
    if ((aStatus == oSkype.Convert.TextToChatMessage
Status("RECEIVED") ||
        aStatus == oSkype.Convert.TextToChatMessage
Status("SENT")) &&
        aMsg.Type == oSkype.Convert.TextToChatMessa
geType("SAID")) {

        WScript.Echo(">Message " + aMsg.Id + " " + aMs
g.FromHandle + " said " + aMsg.Body);
    }
}
```

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# MessageStatus.vbs

This script logs chat text messages.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = CreateObject("Skype4COM.Skype")

'// Connect events:
WScript.ConnectObject oSkype, "Skype_"

'// Connect to Skype API:
oSkype.Attach

'// Loop and sleep:
Do While True
    WScript.Sleep(60000)
Loop

'// Chat message status events:
Public Sub Skype_MessageStatus(ByRef aMsg, ByVal a
Status)

    If (aStatus = oSkype.Convert.TextToChatMessageSt
atus("RECEIVED") Or _
        aStatus = oSkype.Convert.TextToChatMessageSt
atus("SENT")) And _
        aMsg.Type = oSkype.Convert.TextToChatMessage
Type("SAID") Then

        WScript.Echo ">Message " & aMsg.Id & " " & a
Msg.FromHandle & " said " & aMsg.Body
    End If
```

End Sub

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			



## MissedMessages.js

---

This script list all missed messages and changes message status to "Read".

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
// Create a Skype4COM object:
var oSkype = new ActiveXObject("Skype4COM.Skype");

// Query the total number of missed messages:
WScript.Echo("Total missed message count: " + oSkype.MissedMessages.Count);

// Query the id, timestamp, and sender Skype name of the message:
for (var i=0; i < oSkype.MissedMessages.Count; i++)
{
    var oMsg = oSkype.MissedMessages.Item(i+1);
    WScript.Echo("Msg id: " + oMsg.Id + " at " + oMsg.Timestamp + " from " + oMsg.FromHandle);
}
// Mark the missed message as "Read"
for (var i=0; i < oSkype.MissedMessages.Count; i++)
{
    var oMsg = oSkype.MissedMessages.Item(i+1);
    oMsg.Seen = true;
}
```

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## MissedMessages.vbs

---

This script list all missed messages and changes message status to "Read".

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
","Skype_")

'// Query the total number of missed messages:
WScript.Echo "Total missed message count: " & oSkype.MissedMessages.Count

'// Query the id, timestamp, and sender Skype name
of the message:
For Each oMsg In oSkype.MissedMessages
    WScript.Echo "Msg id: " & oMsg.Id & " at " & oMsg.Timestamp & " from " & oMsg.FromHandle
Next

'// Mark the missed message as "Read"
For Each oMsg In oSkype.MissedMessages
    oMsg.Seen = True
Next
```

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Plugins.vbs

This script demos how to create plugin menu items and events.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
const pluginContextChat = 0
const pluginContextCall = 1
const pluginContextContact = 2
const pluginContextMyself = 3
const pluginContextTools = 4

const pluginContactTypeUndefined = -1
const pluginContactTypeAll = 0
const pluginContactTypeSkype = 1
const pluginContactTypeSkypeOut = 2

const pluginIconPath = "c:\temp\icon.png"

Set oSkype = WScript.CreateObject("Skype4COM.Skype"
, "Skype_")

Set oMenu1 = oSkype.Client.CreateMenuItem("menu1",
pluginContextContact, "&_
Single Contact of Any Type", "Plugin that does
cool stuff for single contact of any type", plugin
IconPath, True, pluginContactTypeAll)

Set oMenu2 = oSkype.Client.CreateMenuItem("menu2",
pluginContextContact, "&_
Single Skype Contact", "Plugin that does cool s
tuff for single Skype contact", pluginIconPath, Tr
ue, pluginContactTypeSkype, False)

Set oMenu3 = oSkype.Client.CreateMenuItem("menu3",
```

```
pluginContextContact, ""&_
    "Multiple Skype Contacts", "Plugin that does cool
stuff for multiple contacts", pluginIconPath, True,
pluginContactTypeSkype, True)
```

```
Set oMenu4 = oSkype.Client.CreateMenuItem("menu4",
pluginContextContact, ""&_
    "Single SkypeOut Contact", "Plugin that does cool
stuff for a single SkypeOut contact", pluginIcon
Path, True, pluginContactTypeSkypeOut, False)
```

```
Set oMenu5 = oSkype.Client.CreateMenuItem("menu5",
pluginContextContact, ""&_
    "Multiple SkypeOut Contacts", "Plugin that does
cool stuff for a one or more SkypeOut contacts", p
luginIconPath, True, pluginContactTypeSkypeOut, Tr
ue)
```

```
Set oMenu6 = oSkype.Client.CreateMenuItem("menu6",
pluginContextCall, ""&_
    "Single Participant Call", "Plugin that does cool
stuff in a call", pluginIconPath, True, pluginCo
ntactTypeUndefined, False)
```

```
Set oMenu7 = oSkype.Client.CreateMenuItem("menu7",
pluginContextCall, ""&_
    "Multiple Participants Call", "Plugin that does
cool stuff in a conference call", pluginIconPath,
True, pluginContactTypeUndefined, True)
```

```
Set oMenu8 = oSkype.Client.CreateMenuItem("menu8",
pluginContextChat, ""&_
    "Single Chat Member", "Plugin that does cool stu
ff for one-to-one chat", pluginIconPath, True, plu
ginContactTypeUndefined, False)
```

```
Set oMenu9 = oSkype.Client.CreateMenuItem("menu9",
```

```

pluginContextChat, ""&_
    "Multiple Chat Members", "Plugin that does cool
stuff in group chat", pluginIconPath, True, plugin
ContactTypeUndefined, True)

Set oMenu10 = oSkype.Client.CreateMenuItem("menu10"
, pluginContextTools, ""&_
    "Tools plugin", "Plugin that does cool stuff in
tools menu", pluginIconPath, True, pluginContactTy
peUndefined, False)

Set oEvent = oSkype.Client.CreateEvent("event1", "P
lugin example is running", "Click here to stop the
example")
Do While Not bEventClicked
    WScript.Sleep(1000)
Loop
oEvent.Delete
oMenu1.Delete
oMenu2.Delete
oMenu3.Delete
oMenu4.Delete
oMenu5.Delete
oMenu6.Delete
oMenu7.Delete
oMenu8.Delete
oMenu9.Delete
oMenu10.Delete

Public Sub Skype_PluginEventClicked(ByVal aEvent)
    WScript.Echo ">Event "& aEvent.Id & " clicked."
    bEventClicked = True
End Sub

Public Sub Skype_PluginMenuItemClicked(ByRef aMenu
Item, ByRef aUsers, ByVal aPluginContext, ByVal aC
ontextId)

```

```
WScript.Echo ">Menu item ["& aMenuItem.Id &"] co
ntext [" & pluginContextToText(aPluginContext) & "
] id ["& aContextId &"] clicked."
  For Each oUser In aUsers
    WScript.Echo " User: "& oUser.FullName & " (" &
oUser.Handle &)"
  Next
End Sub

Function pluginContextToText (context)
  Select Case context
  Case pluginContextChat
    pluginContextToText = "Chat"
  Case pluginContextCall
    pluginContextToText = "Call"
  Case pluginContextContact
    pluginContextToText = "Contact"
  Case pluginContextMyself
    pluginContextToText = "Myself"
  Case pluginContextTools
    pluginContextToText = "Tools"
  End Select
End Function
```



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Profile.vbs

---

This script lists the profile properties for the current user.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// List the profile properties for the user:
WScript.Echo "Full name: " & oSkype.CurrentUserPro
file.FullName
WScript.Echo "Birthday: " & oSkype.CurrentUserProf
ile.Birthday
WScript.Echo "Sex: " & oSkype.CurrentUserProfile.S
ex
WScript.Echo "Languages: " & oSkype.CurrentUserPro
file.Languages
WScript.Echo "Country: " & oSkype.CurrentUserProfi
le.Country
WScript.Echo "IP Country: " & oSkype.CurrentUserPr
ofile.IPCountry
WScript.Echo "Province: " & oSkype.CurrentUserProf
ile.Province
WScript.Echo "City: " & oSkype.CurrentUserProfile.
City
WScript.Echo "Phone home: " & oSkype.CurrentUserPr
ofile.PhoneHome
WScript.Echo "Phone office: " & oSkype.CurrentUser
Profile.PhoneOffice
```

```
WScript.Echo "Phone mobile: " & oSkype.CurrentUser
Profile.PhoneMobile
WScript.Echo "Homepage: " & oSkype.CurrentUserProf
ile.Homepage
WScript.Echo "About: " & oSkype.CurrentUserProfile
.About
WScript.Echo "Mood: " & oSkype.CurrentUserProfile.
MoodText
WScript.Echo "Timezone: " & oSkype.CurrentUserProf
ile.Timezone
WScript.Echo "Noanswer timeout: " & oSkype.Current
UserProfile.CallNoAnswerTimeout
WScript.Echo "Forward rules: " & oSkype.CurrentUse
rProfile.CallForwardRules
WScript.Echo "Call apply forward: " & oSkype.Curre
ntUserProfile.CallApplyCF
WScript.Echo "Call send to voicemail: " & oSkype.C
urrentUserProfile.CallSendToVM
WScript.Echo "Balance: " & oSkype.CurrentUserProfi
le.Balance / 100 &" "& oSkype.CurrentUserProfile.B
alanceCurrency
WScript.Echo "Balance Text: " & oSkype.CurrentUser
Profile.BalanceToText
WScript.Echo "Validated SMS numbers: " & oSkype.Cu
rrentUserProfile.ValidatedSmsNumbers
```

```
'// The AttachmentStatus event handler monitors at
achment status and automatically attempts to reat
tach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Response.vbs

---

This script provides a response to a message received in a chat.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Declare the following Skype constants:
cAttachmentStatus_Available = oSkype.Convert.TextT
oAttachmentStatus("AVAILABLE")
cMessageStatus_Sending = oSkype.Convert.TextToChat
MessageStatus("SENDING")
cMessageStatus_Received = oSkype.Convert.TextToCha
tMessageStatus("RECEIVED")
cMessageType_Said = oSkype.Convert.TextToChatMessa
geType("SAID")
cMessageType_Left = oSkype.Convert.TextToChatMessa
geType("LEFT")

'// The SendMessage command will fail if the user
is offline. To avoid failure, check user status an
d change to online if necessary:
If cUserStatus_Offline = oSkype.CurrentUserStatus
Then oSkype.ChangeUserStatus(cUserStatus_Online) E
nd If

'// Sleep
Do While True
```

```

    WScript.Sleep(60000)
Loop

'// The AttachmentStatus event handler monitors at
achment status and attempts to connect to the Sky
pe API:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Con
vert.AttachmentStatusToText(aStatus)
    If aStatus = cAttachmentStatus_Available Then oS
kype.Attach() End If
End Sub

'// The MessageStatus event handler monitors messa
ge status, decodes received messages and, for thos
e of type "Said", sends an autoresponse quoting th
e original message:
Public Sub Skype_MessageStatus(ByRef aMsg, ByVal a
Status)
    WScript.Echo ">Message " & aMsg.Id & " status "
& oSkype.Convert.ChatMessageStatusToText(aStatus)
    If aStatus = cMessageStatus_Received Then
        DecodeMsg aMsg
        If aMsg.Type = cMessageType_Said Then
            'oSkype.SendMessage aMsg.FromHandle, "This is
autoresponse."
            aMsg.Chat.SendMessage "You said [" & aMsg.Bo
dy & "]"
        End If
    End If
End Sub

'// The DecodeMsg event handler decodes messages i
n a chat and converts leave reasons to text for me
ssages of type "Left":
Public Sub DecodeMsg(ByRef oMsg)
    sText = oMsg.FromHandle & " " & oSkype.Convert.C

```

```
hatMessageTypeToText(oMsg.Type) & ":"
  If len(oMsg.Body) Then
    sText = sText & " " & oMsg.Body
  End If
  Dim oUser
  For Each oUser In oMsg.Users
    sText = sText & " " & oUser.Handle
  Next
  If oMsg.Type = cMessageType_Left Then
    sText = sText & " " & oSkype.Convert.ChatLeaveReasonToText(oMsg.LeaveReason)
  End If
  WScript.Echo ">" & sText
End Sub
```

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			



## Search.vbs

---

This script performs a simple search for a user.

Copyright © 2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Search for users John Doe and return the Skypn
ame, full name and country of each matching user:
Set oUsers = oSkype.SearchForUsers("john doe")
For Each oUser In oUsers
    WScript.Echo oUser.Handle & " "& oUser.Fullname &
" "& oUser.Country
Next

'// The AttachmentStatus event handler monitors at
tachment status and automatically attempts to reat
tach to the API following loss of connection:
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat
us("AVAILABLE") Then oSkype.Attach() End If
End Sub
```

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# Settings.vbs

This script queries the Skype general parameters.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// Query Skype general parameters:
WScript.Echo "Audio in:" & oSkype.Settings.AudioIn
WScript.Echo "Audio out:" & oSkype.Settings.AudioO
ut
WScript.Echo "Auto away:" & oSkype.Settings.AutoAw
ay
WScript.Echo "Ringer:" & oSkype.Settings.Ringer
'oSkype.Settings.Avatar = "c:\temp\test.jpg"
WScript.Echo "Ringtone status:" & oSkype.Settings.
RingToneStatus
'oSkype.Settings.RingTone = "c:\temp\test.mp3"
WScript.Echo "Video in:" & oSkype.Settings.VideoIn
WScript.Echo "PC speaker:" & oSkype.Settings.PCSpe
aker
WScript.Echo "AGC:" & oSkype.Settings.AGC
WScript.Echo "AEC:" & oSkype.Settings.AEC
'oSkype.Settings.Language = "en"
WScript.Echo "Language:" & oSkype.Settings.Languag
e

'// The language change event
```

```
Public Sub Skype_UILanguageChanged(ByVal aCode)
    WScript.Echo ">Language changed to " & aCode
End Sub
```

```
'// The AttachmentStatus event handler monitors at  
tachment status and automatically attempts to reat  
tach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
    WScript.Echo ">Attachment status " & oSkype.Conv  
ert.AttachmentStatusToText(aStatus)
    If aStatus = oSkype.Convert.TextToAttachmentStat  
us("AVAILABLE") Then oSkype.Attach() End If
End Sub
```

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

## Sms.vbs

This script sends a SMS message.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

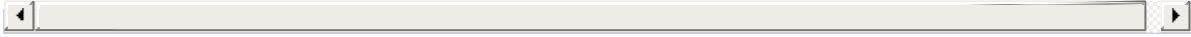
'// Send SMS:
Set oSMS = oSkype.SendSms("+1234567890", "Hello!")

WScript.Sleep(60000)

'// Message event handler:
Public Sub Skype_SmsMessageStatusChanged(ByRef aSm
s, ByVal aStatus)
    WScript.Echo ">Sms " & aSms.Id & " status " & a
Status & " " & oSkype.Convert.SmsMessageStatusToTe
xt(aStatus)
End Sub

'// Target event handler:
Public Sub Skype_SmsTargetStatusChanged(ByRef aTar
get, ByVal aStatus)
    WScript.Echo ">Sms " & aTarget.Message.Id & " t
arget " & aTarget.Number & " status " & aStatus &
" " & oSkype.Convert.SmsTargetStatusToText(aStatus)

End Sub
```



---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			



## Smss.vbs

---

This script lists all SMS messages.

Copyright © 2004-2006 Skype Limited. Licensed under BSD license.

```
'// Create a Skype4COM object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Start the Skype client:
If Not oSkype.Client.IsRunning Then oSkype.Client.
Start() End If

'// List Smss
For Each oSms In oSkype.Smss
    WScript.Echo "Sms Id: " & oSms.Id & " time " & o
Sms.Timestamp
    WScript.Echo "    type: " & oSms.Type & " " & oSky
pe.Convert.SmsMessageTypeToText(oSms.Type)
    WScript.Echo "    status: " & oSms.Status & " " &
oSkype.Convert.SmsMessageStatusToText(oSms.Status)
    WScript.Echo "    failure reason: " & oSms.Failure
Reason
    WScript.Echo "    failed unseen: " & oSms.IsFailed
Unseen
    WScript.Echo "    price: " & oSms.Price
    WScript.Echo "    price precision: " & oSms.PriceP
recision
    WScript.Echo "    price currency: " & oSms.PriceCu
rrency
    WScript.Echo "    reply to number: " & oSms.ReplyT
oNumber
    For Each oTarget In oSms.Targets
        WScript.Echo "    target: " & oTarget.Number & "
```

```
status: " & oSkype.Convert.SmsTargetStatusToText(
oTarget.Status)
Next
WScript.Echo " body: [" & oSms.Body & "]"
For Each oChunk In oSms.Chunks
    WScript.Echo " chunk: " & oChunk.Id & " [" &
oChunk.Text & "]"
Next
Next
```

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# VoicemailServer.cs

This script listens for new voicemails, sends SMS message and answers incoming calls.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
using System;
using System.Collections.Generic;
using System.Text;
using SKYPE4COMLib;

namespace ConsoleApplication1
{
    class VoicemailServer
    {
        [MTAThread]
        static void Main(string[] args)
        {
            oSkype.CallStatus += Skype_CallStatus;
            oSkype.VoicemailStatus += Skype_Voicem
            ailStatus;
            oSkype.Attach(5, true);
            while (true){
                System.Threading.Thread.Sleep(1000
            );
        }
        static void Skype_CallStatus(Call call, TC
            allStatus status)
        {
            Console.WriteLine(">Call " + call.Id +
            " status: " + oConvert.CallStatusToText(status));
            if (status == TCallStatus.clsRinging &
            & (call.Type == TCallType.cltIncomingP2P || call.T
```

```

ype == TCallType.cltIncomingPSTN)) {
    if (call.PartnerHandle == sMyMobil
eNumber) {
        call.Answer();
        foreach (Voicemail oVoicemail
in oSkype.Voicemails) {
            if (oVoicemail.Status == T
VoicemailStatus.vmsUnplayed) {

                oVoicemail.StartPlayba
ckInCall();

                while (oVoicemail.Stat
us != TVoicemailStatus.vmsPlayed) {
                    System.Threading.T
hread.Sleep(1000);
                }

                oVoicemail.SetUnplayed
());
            }
        }
    }
    if (call.Status == TCallStatus.cls
InProgress) {
        call.Finish();
    }
}
static void Skype_VoicemailStatus(Voicemai
l voicemail, TVoicemailStatus status)
{
    Console.WriteLine(">Voicemail status: "
+ oConvert.VoicemailStatusToText(status));
    if (status == TVoicemailStatus.vmsDown
loading) {
        Console.WriteLine("Sending SMS to "
+ sMyMobileNumber);
    }
}

```

```
        SmsMessage oMessage = oSkype.SendS  
ms(sMyMobileNumber, "You have new voicemail from "  
+ voicemail.PartnerDisplayName, "");  
    }  
}  
static Skype oSkype = new Skype();  
static Conversion oConvert = new Conversio  
n();  
static String sMyMobileNumber = "+12345678  
90";  
}  
}
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			

# VoicemailServer.vbs

This script listens for new voicemails, sends SMS message and answers incoming calls.

Copyright  2004-2006 Skype Limited. Licensed under [BSD license](#).

```
'// Create Skype object:
Set oSkype = WScript.CreateObject("Skype4COM.Skype
", "Skype_")

'// Connect to Skype API:
oSkype.Attach

'// The mobile number used in this example
Const cMyMobileNumber = "+1234567890"

'// Skype constants
cCallStatusRinging = oSkype.Convert.TextToCallStat
us("RINGING")
cCallStatusInProgress = oSkype.Convert.TextToCallS
tatus("INPROGRESS")
cCallTypeIncomingP2P = oSkype.Convert.TextToCallTy
pe("INCOMING_P2P")
cCallTypeIncomingPSTN = oSkype.Convert.TextToCallT
ype("INCOMING_PSTN")
cVoicemailStatusUnplayed = oSkype.Convert.TextToVo
icemailStatus("UNPLAYED")
cVoicemailStatusPlayed = oSkype.Convert.TextToVoic
emailStatus("PLAYED")
cAttachmentStatus_Available = oSkype.Convert.TextT
oAttachmentStatus("AVAILABLE")

'// This script runs forever
Do While True
```



```

WScript.Sleep(30000)
Loop

'// The CallStatus event handler monitors call sta
tus and if the status is "ringing" and it is an in
coming call, answers the call:
Public Sub Skype_CallStatus(ByRef aCall, ByVal aSt
atus)
    WScript.Echo ">Call " & aCall.Id & " status " &
aStatus & " " & oSkype.Convert.CallStatusToText(a
Status)
    If cCallStatusRinging = aStatus And _
        (cCallTypeIncomingP2P = aCall.Type Or cCallTyp
eIncomingPSTN = aCall.Type) Then

        '// Accept incoming call only from predefined
phone number or SkypeName:
        If aCall.PartnerHandle = cMyMobileNumber Then

            aCall.Answer()
            For Each oVoicemail In oSkype.Voicemails
                If oVoicemail.Status = cVoicemailStatusUn
played Then
                    oVoicemail.StartPlaybackInCall
                    Do While oVoicemail.Status <> cVoicemai
lStatusPlayed
                        WScript.Sleep(1000)
                    Loop
                    '// Change the voicemail status back to
"unplayed"
                    oVoicemail.SetUnplayed
                End If
            Next
            If cCallStatusInProgress = aCall.Status The
n aCall.Finish() End If
        End If
    End If

```

```
End Sub
```

```
'// The VoicemailStatus event handler monitors voicemail status changes:
```

```
Public Sub Skype_VoicemailStatus(aVoicemail, aStatus)
```

```
    WScript.Echo ">Voicemail " & aVoicemail.Id & " status " & oSkype.Convert.VoicemailStatusToText(aStatus)
```

```
    If cVoicemailStatusDownloading = aStatus Then
```

```
        WScript.Echo "Sending SMS to " & cMyMobileNumber
```

```
        Set oSms = oSkype.SendSms(cMyMobileNumber, "You have new voicemail from " & aVoicemail.PartnerDisplayName)
```

```
    End If
```

```
End Sub
```

```
'// The AttachmentStatus event handler monitors attachment status and automatically attempts to reattach to the API following loss of connection:
```

```
Public Sub Skype_AttachmentStatus(ByVal aStatus)
```

```
    WScript.Echo ">Attachment status " & oSkype.Convert.AttachmentStatusToText(aStatus)
```

```
    If aStatus = cAttachmentStatus_Available Then oSkype.Attach() End If
```

```
End Sub
```

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>	
<a href="#">Files</a>	<a href="#">Examples</a>			
<a href="#">File List</a>				

# Skype4COM.idl

```
00001 // Skype4COM.idl : IDL source for Skype4COM.
dll
00002 //
00003
00004 import "oaidl.idl";
00005 import "ocidl.idl";
00006
00007 #include "HelpString.h"
00008 #include "olectl.h"
00009
00014 [
00015     uuid(74A50AB8-1E6F-4246-8933-08E4FBBCF
4E5),
00016     helpstring(hsAttachStatus), helpcontex
t(17)
00017 ]
00018 typedef enum TAttachmentStatus
00019 {
00020     [helpcontext(17),helpstring(hsAttachSt
atus_Unknown)] apiAttachUnknown = -1,
00021     [helpcontext(17),helpstring(hsAttachSt
atus_Success)] apiAttachSuccess = 0,
00022     [helpcontext(17),helpstring(hsAttachSt
atus_Pending)] apiAttachPendingAuthorization = 1,
00023     [helpcontext(17),helpstring(hsAttachSt
atus_Refused)] apiAttachRefused = 2,
00024     [helpcontext(17),helpstring(hsAttachSt
atus_NotAvail)] apiAttachNotAvailable = 3,
00025     [helpcontext(17),helpstring(hsAttachSt
atus_Available)] apiAttachAvailable = 4
00026 } TAttachmentStatus;
00027
00028 [
```

```
00029         uuid(69658FFD-F587-4946-A4EC-7C0709EA9
F44),
00030         helpstring(hsConnStatus), helpcontext(
17)
00031     ]
00032     typedef enum TConnectionStatus
00033     {
00034         [helpcontext(17),helpstring(hsConnStat
us_Unknown)] conUnknown = -1,
00035         [helpcontext(17),helpstring(hsConnStat
us_Offline)] conOffline = 0,
00036         [helpcontext(17),helpstring(hsConnStat
us_Connecting)] conConnecting = 1,
00037         [helpcontext(17),helpstring(hsConnStat
us_Pausing)] conPausing = 2,
00038         [helpcontext(17),helpstring(hsConnStat
us_Online)] conOnline = 3
00039     } TConnectionStatus;
00040
00041     [
00042         uuid(FBAB58DC-9C60-4EE6-9826-26EE2335D
D0A),
00043         helpstring(hsUserStatus), helpcontext(
17)
00044     ]
00045     typedef enum TUserStatus
00046     {
00047         [helpcontext(17),helpstring(hsUserStat
us_Unknown)] cusUnknown = -1,
00048         [helpcontext(17),helpstring(hsUserStat
us_Offline)] cusOffline = 0,
00049         [helpcontext(17),helpstring(hsUserStat
us_Online)] cusOnline = 1,
00050         [helpcontext(17),helpstring(hsUserStat
us_Away)] cusAway = 2,
00051         [helpcontext(17),helpstring(hsUserStat
us_NotAvail)] cusNotAvailable = 3,
```

```
00052         [helpcontext(17),helpstring(hsUserStat
us_DND)] cusDoNotDisturb = 4,
00053         [helpcontext(17),helpstring(hsUserStat
us_Invisible)] cusInvisible = 5,
00054         [helpcontext(17),helpstring(hsUserStat
us_LoggedOut)] cusLoggedOut = 6,
00055         [helpcontext(17),helpstring(hsUserStat
us_SkypeMe)] cusSkypeMe = 7
00056     } TUserStatus;
00057
00058     [
00059         uuid(DCE03024-D231-45D4-AB69-D22C9E295
4BA),
00060         helpstring(hsFailureReason), helpconte
xt(17)
00061     ]
00062     typedef enum TCallFailureReason
00063     {
00064         [helpcontext(17),helpstring(hsFailureR
eason_Unknown)] cfrUnknown = -1,
00065         [helpcontext(17),helpstring(hsFailureR
eason_MiscError)] cfrMiscError = 0,
00066         [helpcontext(17),helpstring(hsFailureR
eason_DoesNotExist)] cfrUserDoesNotExist= 1,
00067         [helpcontext(17),helpstring(hsFailureR
eason_UserOffline)] cfrUserIsOffline = 2,
00068         [helpcontext(17),helpstring(hsFailureR
eason_NoProxy)] cfrNoProxyFound = 3,
00069         [helpcontext(17),helpstring(hsFailureR
eason_Terminated)] cfrSessionTerminated= 4,
00070         [helpcontext(17),helpstring(hsFailureR
eason_NoCodec)] cfrNoCommonCodec = 5,
00071         [helpcontext(17),helpstring(hsFailureR
eason_SoundIOError)] cfrSoundIOError = 6,
00072         [helpcontext(17),helpstring(hsFailureR
eason_RemoteDevice)] cfrRemoteDeviceError = 7,
00073         [helpcontext(17),helpstring(hsFailureR
```

```

eason_Blocked)] cfrBlockedByRecipient = 8,
00074      [helpcontext(17),helpstring(hsFailureR
eason_NotFriend)] cfrRecipientNotFriend = 9,
00075      [helpcontext(17),helpstring(hsFailureR
eason_NotAuthorized)] cfrNotAuthorizedByRecipient
= 10,
00076      [helpcontext(17),helpstring(hsFailureR
eason_RecordErr)] cfrSoundRecordingError = 11
00077  } TCallFailureReason;
00078
00079  [
00080      uuid(47E3ACA4-04BD-4A4B-AB17-7EBF9FD7F
5DE),
00081      helpstring(hsCallStatus), helpcontext(
17)
00082  ]
00083  typedef enum TCallStatus
00084  {
00085      [helpcontext(17),helpstring(hsCallStat
us_Unknown)] clsUnknown = -1,
00086      [helpcontext(17),helpstring(hsCallStat
us_Unplaced)] clsUnplaced = 0,
00087      [helpcontext(17),helpstring(hsCallStat
us_Routing)] clsRouting = 1,
00088      [helpcontext(17),helpstring(hsCallStat
us_EarlyMedia)] clsEarlyMedia = 2,
00089      [helpcontext(17),helpstring(hsCallStat
us_Failed)] clsFailed = 3,
00090      [helpcontext(17),helpstring(hsCallStat
us_Ringing)] clsRinging = 4,
00091      [helpcontext(17),helpstring(hsCallStat
us_InProgress)] clsInProgress = 5,
00092      [helpcontext(17),helpstring(hsCallStat
us_OnHold)] clsOnHold = 6,
00093      [helpcontext(17),helpstring(hsCallStat
us_Finished)] clsFinished = 7,
00094      [helpcontext(17),helpstring(hsCallStat

```

```
us_Missed)] clsMissed = 8,
00095      [helpcontext(17),helpstring(hsCallStat
us_Refused)] clsRefused = 9,
00096      [helpcontext(17),helpstring(hsCallStat
us_Busy)] clsBusy = 10,
00097      [helpcontext(17),helpstring(hsCallStat
us_Cancelled)] clsCancelled = 11,
00098      [helpcontext(17),helpstring(hsCallStat
us_LocalHold)] clsLocalHold = 12,
00099      [helpcontext(17),helpstring(hsCallStat
us_RemoteHold)] clsRemoteHold = 13,
00100      [helpcontext(17),helpstring(hsCallStat
us_VmBuffGreeting)] clsVoicemailBufferingGreeting
= 14,
00101      [helpcontext(17),helpstring(hsCallStat
us_VmPlayGreeting)] clsVoicemailPlayingGreeting =
15,
00102      [helpcontext(17),helpstring(hsCallStat
us_VmRecording)] clsVoicemailRecording = 16,
00103      [helpcontext(17),helpstring(hsCallStat
us_VmUploading)] clsVoicemailUploading = 17,
00104      [helpcontext(17),helpstring(hsCallStat
us_VmSent)] clsVoicemailSent = 18,
00105      [helpcontext(17),helpstring(hsCallStat
us_VmCancelled)] clsVoicemailCancelled = 19,
00106      [helpcontext(17),helpstring(hsCallStat
us_VmFailed)] clsVoicemailFailed = 20,
00107      [helpcontext(17),helpstring(hsCallStat
us_Transferring)] clsTransferring= 21,
00108      [helpcontext(17),helpstring(hsCallStat
us_Transferred)] clsTransferred= 22
00109      } TCallStatus;
00110
00111      [
00112          uuid(38667A7C-DF57-4C1D-AC65-F7FC87B0E
000),
00113          helpstring(hsCallType), helpcontext(17
```



```

)
00114     ]
00115     typedef enum TCallType
00116     {
00117         [helpcontext(17),helpstring(hsCallType
_Unknown)] cltUnknown = -1,
00118         [helpcontext(17),helpstring(hsCallType
_IncomingPSTN)] cltIncomingPSTN = 0,
00119         [helpcontext(17),helpstring(hsCallType
_OutgoingPSTN)] cltOutgoingPSTN = 1,
00120         [helpcontext(17),helpstring(hsCallType
_IncomingP2P)] cltIncomingP2P = 2,
00121         [helpcontext(17),helpstring(hsCallType
_OutgoingP2P)] cltOutgoingP2P = 3
00122     } TCallType;
00123
00124     [
00125         uuid(AA127663-5272-473D-B14C-1A8BC3FB6
6C6),
00126         helpstring(hsCallHistory), helpcontext
(17)
00127     ]
00128     typedef enum TCallHistory
00129     {
00130         [helpcontext(17),helpstring(hsCallHist
ory_All)] chsAllCalls = 0,
00131         [helpcontext(17),helpstring(hsCallHist
ory_Missed)] chsMissedCalls = 1,
00132         [helpcontext(17),helpstring(hsCallHist
ory_Incoming)] chsIncomingCalls = 2,
00133         [helpcontext(17),helpstring(hsCallHist
ory_Outgoing)] chsOutgoingCalls = 3
00134     } TCallHistory;
00135
00136     [
00137         uuid(169E6E65-20F4-47AD-8329-02425CD53
769),

```

```
00138         helpstring(hsCallVideoStatus), helpcon
text(17)
00139     ]
00140     typedef enum TCallVideoStatus
00141     {
00142         [helpcontext(17),helpstring(hsCallVSta
tus_Unknown)] cvsUnknown = -1,
00143         [helpcontext(17),helpstring(hsCallVSta
tus_None)] cvsNone = 0,
00144         [helpcontext(17),helpstring(hsCallVSta
tus_SendEnabled)] cvsSendEnabled = 1,
00145         [helpcontext(17),helpstring(hsCallVSta
tus_RecvEnabled)] cvsReceiveEnabled = 2,
00146         [helpcontext(17),helpstring(hsCallVSta
tus_BothEnabled)] cvsBothEnabled = 3
00147     } TCallVideoStatus;
00148
00149     [
00150         uuid(944F55CF-F7AC-4955-8368-B89027EBC
75C),
00151         helpstring(hsCallVideoSendStatus), hel
pcontext(17)
00152     ]
00153     typedef enum TCallVideoSendStatus
00154     {
00155         [helpcontext(17),helpstring(hsCallVSen
dStatus_Unknown)] vssUnknown = -1,
00156         [helpcontext(17),helpstring(hsCallVSen
dStatus_NotAvailable)] vssNotAvailable = 0,
00157         [helpcontext(17),helpstring(hsCallVSen
dStatus_Available)] vssAvailable = 1,
00158         [helpcontext(17),helpstring(hsCallVSen
dStatus_Starting)] vssStarting = 2,
00159         [helpcontext(17),helpstring(hsCallVSen
dStatus_Rejected)] vssRejected = 3,
00160         [helpcontext(17),helpstring(hsCallVSen
dStatus_Running)] vssRunning = 4,
```

```

00161      [helpcontext(17),helpstring(hsCallVSendStatus_Stopping)] vssStopping = 5,
00162      [helpcontext(17),helpstring(hsCallVSendStatus_Paused)] vssPaused = 6
00163  } TCallVideoSendStatus;
00164
00165  [
00166      uuid(8b0d63f5-57ec-4305-8f0d-517b6e85bad2),
00167      helpstring(hsCallIoDeviceType), helpcontext(17)
00168  ]
00169  typedef enum TCallIoDeviceType
00170  {
00171      [helpcontext(17),helpstring(hsCallIoDeviceTypeUnknown)] callIoDeviceTypeUnknown = -1,
00172      [helpcontext(17),helpstring(hsCallIoDeviceTypeSoundcard)] callIoDeviceTypeSoundcard= 0,
00173      [helpcontext(17),helpstring(hsCallIoDeviceTypePort)] callIoDeviceTypePort = 1,
00174      [helpcontext(17),helpstring(hsCallIoDeviceTypeFile)] callIoDeviceTypeFile = 2
00175  } TCallIoDeviceType;
00176
00177  [
00178      uuid(28B277BA-2D86-467D-BB47-A805A15585C1),
00179      helpstring(hsChatMessageType), helpcontext(17)
00180  ]
00181  typedef enum TChatMessageType
00182  {
00183      [helpcontext(17),helpstring(hsChatMsgType_Unknown)] cmeUnknown = -1,
00184      [helpcontext(17),helpstring(hsChatMsgType_With)] cmeCreatedChatWith = 0,
00185      [helpcontext(17),helpstring(hsChatMsgT

```

```
ype_Saw)] cmeSawMembers= 1,
00186      [helpcontext(17),helpstring(hsChatMsgT
ype_Added)] cmeAddedMembers = 2,
00187      [helpcontext(17),helpstring(hsChatMsgT
ype_Topic)] cmeSetTopic = 3,
00188      [helpcontext(17),helpstring(hsChatMsgT
ype_Said)] cmeSaid = 4,
00189      [helpcontext(17),helpstring(hsChatMsgT
ype_Left)] cmeLeft = 5,
00190      [helpcontext(17),helpstring(hsChatMsgT
ype_Emoted)] cmeEmoted = 6,
00191      [helpcontext(17),helpstring(hsChatMsgT
ype_PostContacts)] cmePostedContacts = 7,
00192      [helpcontext(17),helpstring(hsChatMsgT
ype_GapInChat)] cmeGapInChat = 8,
00193      [helpcontext(17),helpstring(hsChatMsgT
ype_SetRole)] cmeSetRole = 9,
00194      [helpcontext(17),helpstring(hsChatMsgT
ype_Kicked)] cmeKicked = 10,
00195      [helpcontext(17),helpstring(hsChatMsgT
ype_SetOpts)] cmeSetOptions = 11,
00196      [helpcontext(17),helpstring(hsChatMsgT
ype_Kickbanned)] cmeKickBanned = 12,
00197      [helpcontext(17),helpstring(hsChatMsgT
ype_Joined)] cmeJoinedAsApplicant = 13,
00198      [helpcontext(17),helpstring(hsChatMsgT
ype_SetPicture)] cmeSetPicture = 14,
00199      [helpcontext(17),helpstring(hsChatMsgT
ype_SetGuideline)] cmeSetGuidelines = 15,
00200    } TChatMessageType;
00201
00202    [
00203      uuid(6CF9ED72-45D0-479E-936A-EB34B7486
61E),
00204      helpstring(hsChatMessageStatus), helpc
ontext(17)
00205    ]
```

```

00206     typedef enum TChatMessageStatus
00207     {
00208         [helpcontext(17),helpstring(hsChatMsgS
tat_Unknown)] cmsUnknown = -1,
00209         [helpcontext(17),helpstring(hsChatMsgS
tat_Sending)] cmsSending = 0,
00210         [helpcontext(17),helpstring(hsChatMsgS
tat_Sent)] cmsSent = 1,
00211         [helpcontext(17),helpstring(hsChatMsgS
tat_Received)] cmsReceived = 2,
00212         [helpcontext(17),helpstring(hsChatMsgS
tat_Read)] cmsRead = 3,
00213     } TChatMessageStatus;
00214
00215     [
00216         uuid(AE9D03DE-7E88-466A-A3E8-1E840ACD9
310),
00217         helpstring(hsChatMemberRole), helpcont
ext(17)
00218     ]
00219     typedef enum TChatMemberRole
00220     {
00221         [helpcontext(17),helpstring(hsChatMemb
erRole_Unknown)] chatMemberRoleUnknown = -1,
00222         [helpcontext(17),helpstring(hsChatMemb
erRole_Creator)] chatMemberRoleCreator = 0,
00223         [helpcontext(17),helpstring(hsChatMemb
erRole_Master)] chatMemberRoleMaster = 1,
00224         [helpcontext(17),helpstring(hsChatMemb
erRole_Helper)] chatMemberRoleHelper = 2,
00225         [helpcontext(17),helpstring(hsChatMemb
erRole_User)] chatMemberRoleUser = 3,
00226         [helpcontext(17),helpstring(hsChatMemb
erRole_Listener)] chatMemberRoleListener = 4,
00227         [helpcontext(17),helpstring("")] chatM
emberRoleApplicant = 5,
00228     } TChatMemberRole;

```

```
00229
00230     [
00231         uuid(F2E9781B-286D-42C7-8FA3-7DE5C52F2
00232         86B),
00233         helpstring(hsUserSex), helpcontext(17)
00234     ]
00235     typedef enum TUserSex
00236     {
00237         [helpcontext(17),helpstring(hsUserSex_
00238         Unknown)] usexUnknown = -1,
00239         [helpcontext(17),helpstring(hsUserSex_
00240         Male)] usexMale = 0,
00241         [helpcontext(17),helpstring(hsUserSex_
00242         Female)] usexFemale = 1
00243     } TUserSex;
00244
00245     [
00246         uuid(5E758C58-F940-476C-B737-8D823737F
00247         EBA),
00248         helpstring(hsBuddyStatus), helpcontext
00249         (17)
00250     ]
00251     typedef enum TBuddyStatus
00252     {
00253         [helpcontext(17),helpstring(hsBuddySta
00254         tus_Unknown)] budUnknown = -1,
00255         [helpcontext(17),helpstring(hsBuddySta
00256         tus_Never)] budNeverBeenFriend = 0,
00257         [helpcontext(17),helpstring(hsBuddySta
00258         tus_Deleted)] budDeletedFriend = 1,
00259         [helpcontext(17),helpstring(hsBuddySta
00260         tus_Pending)] budPendingAuthorization = 2,
00261         [helpcontext(17),helpstring(hsBuddySta
00262         tus_Friend)] budFriend = 3
00263     } TBuddyStatus;
00264
00265     [
```

```
00255         uuid(93F10C8A-D3E5-4E62-9315-1A5D21C9D
BC3),
00256         helpstring(hsOnlineStatus), helpcontex
t(17)
00257     ]
00258     typedef enum TOnlineStatus
00259     {
00260         [helpcontext(17),helpstring(hsOnlineSt
atus_Unknown)] olsUnknown = -1,
00261         [helpcontext(17),helpstring(hsOnlineSt
atus_Offline)] olsOffline = 0,
00262         [helpcontext(17),helpstring(hsOnlineSt
atus_Online)] olsOnline = 1,
00263         [helpcontext(17),helpstring(hsOnlineSt
atus_Away)] olsAway = 2,
00264         [helpcontext(17),helpstring(hsOnlineSt
atus_NotAvail)] olsNotAvailable = 3,
00265         [helpcontext(17),helpstring(hsOnlineSt
atus_DND)] olsDoNotDisturb = 4,
00266         [helpcontext(17),helpstring(hsOnlineSt
atus_SkypeOut)] olsSkypeOut = 5,
00267         [helpcontext(17),helpstring(hsOnlineSt
atus_SkypeMe)] olsSkypeMe = 6
00268     } TOnlineStatus;
00269
00270     [
00271         uuid(FCDEF5B9-BAE9-4213-815F-0F68C7A8F
03E),
00272         helpstring(hsLeaveReason), helpcontext
(17)
00273     ]
00274     typedef enum TChatLeaveReason
00275     {
00276         [helpcontext(17),helpstring(hsLeaveRea
son_Unknown)] leaUnknown = -1,
00277         [helpcontext(17),helpstring(hsLeaveRea
son_NotFound)] leaUserNotFound = 0,
```

```

00278     [helpcontext(17),helpstring(hsLeaveReason_Incapable)] leaUserIncapable = 1,
00279     [helpcontext(17),helpstring(hsLeaveReason_NotFriend)] leaAdderNotFriend = 2,
00280     [helpcontext(17),helpstring(hsLeaveReason_NotAuthorized)] leaAddedNotAuthorized = 3,
00281     [helpcontext(17),helpstring(hsLeaveReason_Declined)] leaAddDeclined = 4,
00282     [helpcontext(17),helpstring(hsLeaveReason_Unsubscribe)] leaUnsubscribe = 5
00283 } TChatLeaveReason;
00284
00285 [
00286     uuid(D90B58CC-53AA-4A8C-8774-4B48BF307D9B),
00287     helpstring(hsChatStatus), helpcontext(17)
00288 ]
00289 typedef enum TChatStatus
00290 {
00291     [helpcontext(17),helpstring(hsChatStatus_Unknown)] chsUnknown = -1,
00292     [helpcontext(17),helpstring(hsChatStatus_Legacy)] chsLegacyDialog = 0,
00293     [helpcontext(17),helpstring(hsChatStatus_Dialog)] chsDialog = 1,
00294     [helpcontext(17),helpstring(hsChatStatus_Accept)] chsMultiNeedAccept = 2,
00295     [helpcontext(17),helpstring(hsChatStatus_Subscrib)] chsMultiSubscribed = 3,
00296     [helpcontext(17),helpstring(hsChatStatus_Unsubscr)] chsUnsubscribed= 4
00297 } TChatStatus;
00298
00299 [
00300     uuid(204D84E3-D7D1-450D-A038-6A3A5210E4FB),

```



```

00301         helpstring(hsChatType), helpcontext(17
)
00302     ]
00303     typedef enum TChatType
00304     {
00305         [helpcontext(17),helpstring(hsChatType
_Unknown)] chatTypeUnknown = -1,
00306         [helpcontext(17),helpstring(hsChatType
_Dialog)] chatTypeDialog = 0,
00307         [helpcontext(17),helpstring(hsChatType
_Legacy)] chatTypeLegacyDialog = 1,
00308         [helpcontext(17),helpstring(hsChatType
_LegacyUnsub)] chatTypeLegacyUnsubscribed = 2,
00309         [helpcontext(17),helpstring(hsChatType
_Multichat)] chatTypeMultiChat = 3,
00310         [helpcontext(17),helpstring(hsChatType
_SharedGrp)] chatTypeSharedGroup = 4,
00311     } TChatType;
00312
00313     [
00314         uuid(1685094C-829A-431B-880C-FE61D82A0
85E),
00315         helpstring(hsChatMyStatus), helpcontex
t(17)
00316     ]
00317     typedef enum TChatMyStatus
00318     {
00319         [helpcontext(17),helpstring(hsChatMySt
atus_Unknown)] chatStatusUnknown = -1,
00320         [helpcontext(17),helpstring(hsChatMySt
atus_Connecting)] chatStatusConnecting = 0,
00321         [helpcontext(17),helpstring(hsChatMySt
atus_WaitAccept)] chatStatusWaitingRemoteAccept =
1,
00322         [helpcontext(17),helpstring(hsChatMySt
atus_AcceptRequired)] chatStatusAcceptRequired = 2
,

```

```

00323      [helpcontext(17),helpstring(hsChatMySt
atus_PasswordRequired)] chatStatusPasswordRequired =
    3,
00324      [helpcontext(17),helpstring(hsChatMySt
atus_Subscribed)] chatStatusSubscribed = 4,
00325      [helpcontext(17),helpstring(hsChatMySt
atus_Unsubscribed)] chatStatusUnsubscribed = 5,
00326      [helpcontext(17),helpstring(hsChatMySt
atus_Disbanded)] chatStatusDisbanded = 6,
00327      [helpcontext(17),helpstring(hsChatMySt
atus_Queued)] chatStatusQueuedBecauseChatIsFull =
    7,
00328      [helpcontext(17),helpstring(hsChatMySt
atus_ApplicationDenied)] chatStatusApplicationDeni
ed = 8,
00329      [helpcontext(17),helpstring(hsChatMySt
atus_Kicked)] chatStatusKicked = 9,
00330      [helpcontext(17),helpstring(hsChatMySt
atus_Banned)] chatStatusBanned = 10,
00331      [helpcontext(17),helpstring(hsChatMySt
atus_Reconnecting)] chatStatusRetryConnecting = 11
,
00332      } TChatMyStatus;
00333
00334      [
00335          uuid(371F92C4-78BC-45BB-B0B4-E9924C2B9
EF9),
00336          helpstring(""), helpcontext(17)
00337      ]
00338      typedef enum TChatOptions
00339      {
00340          [helpcontext(17),helpstring(hsChatOpti
ons_JoinEnabled)] chatOptionJoiningEnabled = 1,
00341          [helpcontext(17),helpstring(hsChatOpti
ons_JoinerApplicant)] chatOptionJoinersBecomeAppli
cants = 2,
00342          [helpcontext(17),helpstring(hsChatOpti

```

```

ons_JoinerListener)] chatOptionJoinersBecomeListeners = 4,
00343      [helpcontext(17),helpstring(hsChatOptions_HistoryDisclosed)] chatOptionHistoryDisclosed
= 8,
00344      [helpcontext(17),helpstring(hsChatOptions_UsersAreListeners)] chatOptionUsersAreListeners
= 16,
00345      [helpcontext(17),helpstring(hsChatOptions_TopicLocked)] chatOptionTopicAndPictureLockedForUsers = 32,
00346  } TChatOptions;
00347
00348  [
00349      uuid(7771BE16-D5F0-4368-8A88-643B5B519
A43),
00350      helpstring(hsVmType), helpcontext(17)
00351  ]
00352  typedef enum TVoicemailType
00353  {
00354      [helpcontext(17),helpstring(hsVmType_U
nknown)] vmtUnknown = -1,
00355      [helpcontext(17),helpstring(hsVmType_I
ncoming)] vmtIncoming = 0,
00356      [helpcontext(17),helpstring(hsVmType_D
efaultGreeting)] vmtDefaultGreeting = 1,
00357      [helpcontext(17),helpstring(hsVmType_C
ustomGreeting)] vmtCustomGreeting = 2,
00358      [helpcontext(17),helpstring(hsVmType_0
utgoing)] vmtOutgoing = 3
00359  } TVoicemailType;
00360
00361  [
00362      uuid(F9810AFA-DBC3-4327-9876-E94A47E7C
ED5),
00363      helpstring(hsVmStatus), helpcontext(17
)

```

```
00364     ]
00365     typedef enum TVoicemailStatus
00366     {
00367         [helpcontext(17),helpstring(hsVmStatus
00368         _Unknown)] vmsUnknown = -1,
00369         [helpcontext(17),helpstring(hsVmStatus
00370         _NotDownloaded)] vmsNotDownloaded = 0,
00371         [helpcontext(17),helpstring(hsVmStatus
00372         _Downloading)] vmsDownloading = 1,
00373         [helpcontext(17),helpstring(hsVmStatus
00374         _Unplayed)] vmsUnplayed = 2,
00375         [helpcontext(17),helpstring(hsVmStatus
00376         _Buffering)] vmsBuffering = 3,
00377         [helpcontext(17),helpstring(hsVmStatus
00378         _Playing)] vmsPlaying = 4,
00379         [helpcontext(17),helpstring(hsVmStatus
00380         _Played)] vmsPlayed = 5,
00381         [helpcontext(17),helpstring(hsVmStatus
00382         _Blank)] vmsBlank = 6,
00383         [helpcontext(17),helpstring(hsVmStatus
00384         _Recording)] vmsRecording = 7,
00385         [helpcontext(17),helpstring(hsVmStatus
00386         _Recorded)] vmsRecorded = 8,
00387         [helpcontext(17),helpstring(hsVmStatus
00388         _Uploading)] vmsUploading = 9,
00389         [helpcontext(17),helpstring(hsVmStatus
00390         _Uploaded)] vmsUploaded = 10,
00391         [helpcontext(17),helpstring(hsVmStatus
00392         _Deleting)] vmsDeleting = 11,
00393         [helpcontext(17),helpstring(hsVmStatus
00394         _Failed)] vmsFailed = 12
00395     } TVoicemailStatus;
00396
00397     [
00398         uuid(977E2F50-2361-4E81-B9F3-2303B8C8B
00399         B49),
00400         helpstring(hsVmFailureReason), helpcon
```

```
text(17)
00386     ]
00387     typedef enum TVoicemailFailureReason
00388     {
00389         [helpcontext(17),helpstring(hsVmFailureReason_Unknown)] vmrUnknown = -1,
00390         [helpcontext(17),helpstring(hsVmFailureReason_NoError)] vmrNoError = 0,
00391         [helpcontext(17),helpstring(hsVmFailureReason_MiscError)] vmrMiscError = 1,
00392         [helpcontext(17),helpstring(hsVmFailureReason_ConnectError)] vmrConnectError = 2,
00393         [helpcontext(17),helpstring(hsVmFailureReason_NoPrivilege)] vmrNoPrivilege = 3,
00394         [helpcontext(17),helpstring(hsVmFailureReason_NoVoicemail)] vmrNoVoicemail = 4,
00395         [helpcontext(17),helpstring(hsVmFailureReason_FileReadError)] vmrFileReadError = 5,
00396         [helpcontext(17),helpstring(hsVmFailureReason_FileWriteError)] vmrFileWriteError = 6,
00397         [helpcontext(17),helpstring(hsVmFailureReason_RecordingError)] vmrRecordingError = 7,
00398         [helpcontext(17),helpstring(hsVmFailureReason_PlaybackError)] vmrPlaybackError = 8
00399     } TVoicemailFailureReason;
00400
00401     [
00402         uuid(4015AAE2-C9AD-40E6-A06C-A1609840C
00403         CDD),
00404         helpstring(hsGroupType), helpcontext(1
00405         7)
00406     ]
00407     typedef enum TGroupType
00408     {
00409         [helpcontext(17),helpstring(hsGroupType_Unknown)] grpUnknown = -1,
00410         [helpcontext(17),helpstring(hsGroupType
```

```

e_Custom)] grpCustomGroup = 0,
00409     [helpcontext(17),helpstring(hsGroupTyp
e_AllUsers)] grpAllUsers = 1,
00410     [helpcontext(17),helpstring(hsGroupTyp
e_AllFriends)] grpAllFriends = 2,
00411     [helpcontext(17),helpstring(hsGroupTyp
e_SkypeFriends)] grpSkypeFriends = 3,
00412     [helpcontext(17),helpstring(hsGroupTyp
e_SkypeOutFriends)] grpSkypeOutFriends = 4,
00413     [helpcontext(17),helpstring(hsGroupTyp
e_OnlineFriends)] grpOnlineFriends = 5,
00414     [helpcontext(17),helpstring(hsGroupTyp
e_PendingAuthFriends)] grpPendingAuthorizationFrie
nds = 6,
00415     [helpcontext(17),helpstring(hsGroupTyp
e_RecentContactUsers)] grpRecentlyContactedUsers =
7,
00416     [helpcontext(17),helpstring(hsGroupTyp
e_WaitMyAuthorization)] grpUsersWaitingMyAuthoriza
tion = 8,
00417     [helpcontext(17),helpstring(hsGroupTyp
e_AuthorizedByMe)] grpUsersAuthorizedByMe = 9,
00418     [helpcontext(17),helpstring(hsGroupTyp
e_BlockedByMe)] grpUsersBlockedByMe = 10,
00419     [helpcontext(17),helpstring(hsGroupTyp
e_UngroupedFriends)] grpUngroupedFriends = 11,
00420     [helpcontext(17),helpstring(hsGroupTyp
e_SharedGroup)] grpSharedGroup = 12,
00421     [helpcontext(17),helpstring(hsGroupTyp
e_ProposedSharedGroup)] grpProposedSharedGroup = 1
3
00422     } TGroupType;
00423
00424     [
00425         uuid(46FF6EA6-E4B4-4E0A-B079-2E616F633
396),
00426         helpstring(hsCallChannelType), helpcon

```

```
text(17)
00427     ]
00428     typedef enum TCallChannelType
00429     {
00430         [helpcontext(17),helpstring(hsCallChan
nelType_Unknown)] cctUnknown= -1,
00431         [helpcontext(17),helpstring(hsCallChan
nelType_Datagram)] cctDatagram= 0,
00432         [helpcontext(17),helpstring(hsCallChan
nelType_Reliable)] cctReliable = 1
00433     } TCallChannelType;
00434
00435     [
00436         uuid(775c273f-6b14-4aa5-b33c-83127f41b
a2e),
00437         helpstring(hsApiSecurityContext), help
context(17), hidden
00438     ]
00439     typedef enum TApiSecurityContext
00440     {
00441         [helpcontext(17),helpstring(hsApiSecur
ityContext_Unknown)] apiContextUnknown = 0x0000,
00442         [helpcontext(17),helpstring(hsApiSecur
ityContext_Voice)] apiContextVoice = 0x0001,
00443         [helpcontext(17),helpstring(hsApiSecur
ityContext_Messaging)] apiContextMessaging = 0x000
2,
00444         [helpcontext(17),helpstring(hsApiSecur
ityContext_Account)] apiContextAccount = 0x0004,
00445         [helpcontext(17),helpstring(hsApiSecur
ityContext_Contacts)] apiContextContacts = 0x0008

00446     } TApiSecurityContext;
00447
00448     [
00449         uuid(48a2b076-ba6a-433b-8065-314987c89
cc1),
```

```
00450         helpstring(hsSmsMessageType), helpcont
ext(17)
00451     ]
00452     typedef enum TSmsMessageType
00453     {
00454         [helpcontext(17),helpstring(hsSmsMessa
geType_Unknown)] smsMessageTypeUnknown = -1,
00455         [helpcontext(17),helpstring(hsSmsMessa
geType_Incoming)] smsMessageTypeIncoming = 0,
00456         [helpcontext(17),helpstring(hsSmsMessa
geType_Outgoing)] smsMessageTypeOutgoing = 1,
00457         [helpcontext(17),helpstring(hsSmsMessa
geType_CCRequest)] smsMessageTypeCCRequest = 2,
00458         [helpcontext(17),helpstring(hsSmsMessa
geType_CCSubmit)] smsMessageTypeCCSubmit = 3
00459     } TSmsMessageType;
00460
00461     [
00462         uuid(531f8be0-6ab6-4521-a998-6c87bb9e5
d5c),
00463         helpstring(hsSmsMessageStatus), helpco
ntext(17)
00464     ]
00465     typedef enum TSmsMessageStatus
00466     {
00467         [helpcontext(17),helpstring(hsSmsMessa
geStatus_Unknown)] smsMessageStatusUnknown = -1,
00468         [helpcontext(17),helpstring(hsSmsMessa
geStatus_Received)] smsMessageStatusReceived = 0,
00469         [helpcontext(17),helpstring(hsSmsMessa
geStatus_Read)] smsMessageStatusRead = 1,
00470         [helpcontext(17),helpstring(hsSmsMessa
geStatus_Composing)] smsMessageStatusComposing = 2
,
00471         [helpcontext(17),helpstring(hsSmsMessa
geStatus_SendToServ)] smsMessageStatusSendingToSer
ver = 3,
```



```
00472      [helpcontext(17),helpstring(hsSmsMessageStatus_SentToServ)] smsMessageStatusSentToServer
    = 4,
00473      [helpcontext(17),helpstring(hsSmsMessageStatus_Delivered)] smsMessageStatusDelivered = 5
,
00474      [helpcontext(17),helpstring(hsSmsMessageStatus_TargetsFailed)] smsMessageStatusSomeTargetsFailed = 6,
00475      [helpcontext(17),helpstring(hsSmsMessageStatus_Failed)] smsMessageStatusFailed = 7
00476  } TSmsMessageStatus;
00477
00478  [
00479      uuid(fe7e480a-5b32-432e-9711-e8e9c10c6bf9),
00480      helpstring(hsSmsFailureReason), helpcontext(17)
00481  ]
00482  typedef enum TSmsFailureReason
00483  {
00484      [helpcontext(17),helpstring(hsSmsFailureReason_Unknown)] smsFailureReasonUnknown = -1,
00485      [helpcontext(17),helpstring(hsSmsFailureReason_MiscErr)] smsFailureReasonMiscError = 0,
00486      [helpcontext(17),helpstring(hsSmsFailureReason_ServConnFailed)] smsFailureReasonServerConnectFailed = 1,
00487      [helpcontext(17),helpstring(hsSmsFailureReason_NoSmsCapability)] smsFailureReasonNoSmsCapability = 2,
00488      [helpcontext(17),helpstring(hsSmsFailureReason_InsufficientFunds)] smsFailureReasonInsufficientFunds = 3,
00489      [helpcontext(17),helpstring(hsSmsFailureReason_InvalidCCode)] smsFailureReasonInvalidConfirmationCode = 4,
```

```
00490         [helpcontext(17),helpstring(hsSmsFailureReason_UserBlocked)] smsFailureReasonUserBlocked
= 5,
00491         [helpcontext(17),helpstring(hsSmsFailureReason_IPBlocked)] smsFailureReasonIPBlocked = 6
,
00492         [helpcontext(17),helpstring(hsSmsFailureReason_NodeBlocked)] smsFailureReasonNodeBlocked
= 7
00493     } TSmsFailureReason;
00494
00495     [
00496         uuid(34f2a71c-d453-4199-ab29-5b6e17b3fb5e),
00497         helpstring(hsSmsTargetStatus), helpcontext(17)
00498     ]
00499     typedef enum TSmsTargetStatus
00500     {
00501         [helpcontext(17),helpstring(hsSmsTargetStatus_Unknown)] smsTargetStatusUnknown = -1,
00502         [helpcontext(17),helpstring(hsSmsTargetStatus_Undefined)] smsTargetStatusUndefined = 0,
00503         [helpcontext(17),helpstring(hsSmsTargetStatus_Analyzing)] smsTargetStatusAnalyzing = 1,
00504         [helpcontext(17),helpstring(hsSmsTargetStatus_Acceptable)] smsTargetStatusAcceptable = 2
,
00505         [helpcontext(17),helpstring(hsSmsTargetStatus_NotRoutable)] smsTargetStatusNotRoutable =
3,
00506         [helpcontext(17),helpstring(hsSmsTargetStatus_DeliveryPending)] smsTargetStatusDeliveryP
ending = 4,
00507         [helpcontext(17),helpstring(hsSmsTargetStatus_DeliverySuccess)] smsTargetStatusDeliveryS
uccessful = 5,
```

```

00508         [helpcontext(17),helpstring(hsSmsTargetStatus_DeliveryFailed)] smsTargetStatusDeliveryFailed = 6
00509     } TSmsTargetStatus;
00510
00511     [
00512         uuid(ED06559A-B942-49f5-AE48-DBD6BEF807B9),
00513         helpstring(hsPluginContext), helpcontext(17)
00514     ]
00515     typedef enum TPluginContext
00516     {
00517         [helpcontext(17),helpstring(hsPluginContext_Unknown)] pluginContextUnknown = -1,
00518         [helpcontext(17),helpstring(hsPluginContext_Chat)] pluginContextChat = 0,
00519         [helpcontext(17),helpstring(hsPluginContext_Call)] pluginContextCall = 1,
00520         [helpcontext(17),helpstring(hsPluginContext_Contact)] pluginContextContact = 2,
00521         [helpcontext(17),helpstring(hsPluginContext_MySelf)] pluginContextMySelf = 3,
00522         [helpcontext(17),helpstring(hsPluginContext_Tools)] pluginContextTools = 4,
00523     } TPluginContext;
00524
00525     [
00526         uuid(000D0FD6-DC80-4cb1-A5AF-1A748DF8159E),
00527         helpstring(hsPluginContactType), helpcontext(17)
00528     ]
00529     typedef enum TPluginContactType
00530     {
00531         [helpcontext(17),helpstring(hsPluginContactType_Unknown)] pluginContactTypeUnknown = -1,

```

```

00532     [helpcontext(17),helpstring(hsPluginCo
nctType_All)] pluginContactTypeAll = 0,
00533     [helpcontext(17),helpstring(hsPluginCo
nctType_Skype)] pluginContactTypeSkype = 1,
00534     [helpcontext(17),helpstring(hsPluginCo
nctType_SkypeOut)] pluginContactTypeSkypeOut= 2,

00535     } TPluginContactType;
00536
00537     [
00538         uuid(BD808125-6CB8-46ad-88E1-139614347
81F),
00539         helpstring(hsFileTransferType), helpco
nctext(17)
00540     ]
00541     typedef enum TFileTransferType
00542     {
00543         [helpcontext(17),helpstring(hsFileTran
sferType_Incoming)] fileTransferTypeIncoming = 0,

00544         [helpcontext(17),helpstring(hsFileTran
sferType_Outgoing)] fileTransferTypeOutgoing = 1,

00545     } TFileTransferType;
00546
00547     [
00548         uuid(C4CE4074-11E0-4026-A809-5C6D259F1
B7C),
00549         helpstring(hsFileTransferStatus), help
context(17)
00550     ]
00551     typedef enum TFileTransferStatus
00552     {
00553         [helpcontext(17),helpstring(hsFileTran
sferStatus_New)] fileTransferStatusNew = 0,
00554         [helpcontext(17),helpstring(hsFileTran

```

```

sferStatus_Connecting)] fileTransferStatusConnecti
ng = 1,
00555      [helpcontext(17),helpstring(hsFileTran
sferStatus_WaitingForAccept)] fileTransferStatusWa
itingForAccept = 2,
00556      [helpcontext(17),helpstring(hsFileTran
sferStatus_Transferring)] fileTransferStatusTransf
erring = 3,
00557      [helpcontext(17),helpstring(hsFileTran
sferStatus_TransferOverRelay)] fileTransferStatusT
ransferringOverRelay = 4,
00558      [helpcontext(17),helpstring(hsFileTran
sferStatus_Paused)] fileTransferStatusPaused = 5,
00559      [helpcontext(17),helpstring(hsFileTran
sferStatus_RemotelyPaused)] fileTransferStatusRemo
telyPaused = 6,
00560      [helpcontext(17),helpstring(hsFileTran
sferStatus_Cancelled)] fileTransferStatusCancelled
= 7,
00561      [helpcontext(17),helpstring(hsFileTran
sferStatus_Completed)] fileTransferStatusCompleted
= 8,
00562      [helpcontext(17),helpstring(hsFileTran
sferStatus_Failed)] fileTransferStatusFailed = 9,
00563      } TFileTransferStatus;
00564
00565      [
00566          uuid(A59E0AAE-86BA-4d10-959B-4E75C2C8D
AD4),
00567          helpstring(hsFTFailureReason), helpcon
text(17)
00568      ]
00569      typedef enum TFileTransferFailureReason
00570      {
00571          [helpcontext(17),helpstring(hsFTFailur
eReason_SenderNotAuthorized)] fileTransferFailureR
easonSenderNotAuthorized= 1,

```

```

00572         [helpcontext(17),helpstring(hsFTFailureReason_RemotelyCancelled)] fileTransferFailureReasonRemotelyCancelled= 2,
00573         [helpcontext(17),helpstring(hsFTFailureReason_FailedRead)] fileTransferFailureReasonFailedRead = 3,
00574         [helpcontext(17),helpstring(hsFTFailureReason_FailedRemoteRead)] fileTransferFailureReasonFailedRemoteRead = 4,
00575         [helpcontext(17),helpstring(hsFTFailureReason_FailedWrite)] fileTransferFailureReasonFailedWrite = 5,
00576         [helpcontext(17),helpstring(hsFTFailureReason_FailedRemoteWrite)] fileTransferFailureReasonFailedRemoteWrite = 6,
00577         [helpcontext(17),helpstring(hsFTFailureReason_RemoteDoesNotSupportFT)] fileTransferFailureReasonRemoteDoesNotSupportFT = 7,
00578         [helpcontext(17),helpstring(hsFTFailureReason_RemoteOfflineTooLong)] fileTransferFailureReasonRemoteOfflineTooLong = 8,
00579     } TFileTransferFailureReason;
00582     [
00583         object,
00584         uuid(48E046A8-31D7-4E5F-A611-47BF32B86405),
00585         dual,
00586         nonextensible,
00587         pointer_default(unique),
00588         oleautomation,
00589         helpstring(hsICommand),
00590         helpcontext(7)
00591     ]
00592     interface ICommand : IDispatch
00593     {
00594         [propget, id(1), helpstring(hsICommand_Id),helpcontext(7)] HRESULT Id([out, retval] LONG

```

```

    *pVal);
00595     [propput, id(1), helpstring(hsICommand
_Id),helpcontext(7)] HRESULT Id([in] LONG newVal);
00596     [propget, id(2), helpstring(hsICommand
_Timeout),helpcontext(7)] HRESULT Timeout([out, re
tval] LONG *pVal);
00597     [propput, id(2), helpstring(hsICommand
_Timeout),helpcontext(7)] HRESULT Timeout([in] LON
G newVal);
00598     [propget, id(3), helpstring(hsICommand
_Block),helpcontext(7)] HRESULT Blocking([out, ret
val] VARIANT_BOOL *pVal);
00599     [propput, id(3), helpstring(hsICommand
_Block),helpcontext(7)] HRESULT Blocking([in] VARI
ANT_BOOL newVal);
00600     [propget, id(4), helpstring(hsICommand
_Command),helpcontext(7)] HRESULT Command([out, re
tval] BSTR *pVal);
00601     [propput, id(4), helpstring(hsICommand
_Command),helpcontext(7)] HRESULT Command([in] BSTR
R newVal);
00602     [propget, id(5), helpstring(hsICommand
_Reply),helpcontext(7)] HRESULT Reply([out, retval
] BSTR *pVal);
00603     [propput, id(5), helpstring(hsICommand
_Reply),helpcontext(7)] HRESULT Reply([in] BSTR ne
wVal);
00604     [propget, id(6), helpstring(hsICommand
_Expect),helpcontext(7)] HRESULT Expected([out, re
tval] BSTR *pVal);
00605     [propput, id(6), helpstring(hsICommand
_Expect),helpcontext(7)] HRESULT Expected([in] BSTR
R newVal);
00606     };
00607     [
00608     object,
00609     uuid(D0BC5168-A518-4192-9D20-0B907B23C

```

```

4D1),
00610         dual,
00611         nonextensible,
00612         pointer_default(unique),
00613         oleautomation,
00614         helpstring(hsIUser),
00615         helpcontext(15)
00616     ]
00617     interface IUser : IDispatch
00618     {
00619         [propget, id(1), helpstring(hsIUser_Handle), helpcontext(15)] HRESULT Handle([out, retval] BSTR *pVal);
00620         [propput, id(1), helpstring(hsIUser_Handle), helpcontext(15)] HRESULT Handle([in] BSTR newVal);
00621         [propget, id(2), helpstring(hsIUser_FullName), helpcontext(15)] HRESULT FullName([out, retval] BSTR *pVal);
00622         [propget, id(3), helpstring(hsIUser_Birthday), helpcontext(15)] HRESULT Birthday([out, retval] DATE *pVal);
00623         [propget, id(4), helpstring(hsIUser_Sex), helpcontext(15)] HRESULT Sex([out, retval] TUserSex *pVal);
00624         [propget, id(5), helpstring(hsIUser_Country), helpcontext(15)] HRESULT Country([out, retval] BSTR *pVal);
00625         [propget, id(6), helpstring(hsIUser_Province), helpcontext(15)] HRESULT Province([out, retval] BSTR *pVal);
00626         [propget, id(7), helpstring(hsIUser_City), helpcontext(15)] HRESULT City([out, retval] BSTR *pVal);
00627         [propget, id(8), helpstring(hsIUser_PhoneHome), helpcontext(15)] HRESULT PhoneHome([out, retval] BSTR *pVal);

```



```
00628      [propget, id(9), helpstring(hsIUser_Ph
oneOffice), helpcontext(15)] HRESULT PhoneOffice([
out, retval] BSTR *pVal);
00629      [propget, id(10), helpstring(hsIUser_P
honeMobile), helpcontext(15)] HRESULT PhoneMobile(
[out, retval] BSTR *pVal);
00630      [propget, id(11), helpstring(hsIUser_H
omepage), helpcontext(15)] HRESULT Homepage([out,
retval] BSTR *pVal);
00631      [propget, id(12), helpstring(hsIUser_A
bout), helpcontext(15)] HRESULT About([out, retval
] BSTR *pVal);
00632      [propget, id(13), helpstring(hsIUser_H
asCallEquipment), helpcontext(15)] HRESULT HasCall
Equipment([out, retval] VARIANT_BOOL *pVal);
00633      [propget, id(14), helpstring(hsIUser_B
uddyStatus), helpcontext(15)] HRESULT BuddyStatus(
[out, retval] TBuddyStatus *pVal);
00634      [propput, id(14), helpstring(hsIUser_B
uddyStatus), helpcontext(15)] HRESULT BuddyStatus(
[in] TBuddyStatus newVal);
00635      [propget, id(15), helpstring(hsIUser_A
uthorized), helpcontext(15)] HRESULT IsAuthorized(
[out, retval] VARIANT_BOOL *pVal);
00636      [propput, id(15), helpstring(hsIUser_A
uthorized), helpcontext(15)] HRESULT IsAuthorized(
[in] VARIANT_BOOL newVal);
00637      [propget, id(16), helpstring(hsIUser_B
locked), helpcontext(15)] HRESULT IsBlocked([out,
retval] VARIANT_BOOL *pVal);
00638      [propput, id(16), helpstring(hsIUser_B
locked), helpcontext(15)] HRESULT IsBlocked([in] V
ARIANT_BOOL newVal);
00639      [propget, id(17), helpstring(hsIUser_D
isplayName), helpcontext(15)] HRESULT DisplayName(
[out, retval] BSTR *pVal);
00640      [propget, id(18), helpstring(hsIUser_0
```

```
nlineStatus), helpcontext(15)] HRESULT OnlineStatus([out, retval] TOnlineStatus *pVal);
00641 [propget, id(19), helpstring(hsIUser_LastOnline), helpcontext(15)] HRESULT LastOnline([out, retval] DATE *pVal);
00642 [propget, id(20), helpstring(hsIUser_CountryCode), helpcontext(15)] HRESULT CountryCode([out, retval] BSTR *pVal);
00643 [propget, id(21), helpstring(hsIUser_RecvAuthReq), helpcontext(15)] HRESULT ReceivedAuthRequest([out, retval] BSTR *pVal);
00644 [propget, id(22), helpstring(hsIUser_SpeedDial), helpcontext(15)] HRESULT SpeedDial([out, retval] BSTR *pVal);
00645 [propput, id(22), helpstring(hsIUser_SpeedDial), helpcontext(15)] HRESULT SpeedDial([in] BSTR newVal);
00646 [propget, id(23), helpstring(hsIUser_CanLeaveVm), helpcontext(15)] HRESULT CanLeaveVoiceMail([out, retval] VARIANT_BOOL *pVal);
00647 [propget, id(24), helpstring(hsIUser_MoodText), helpcontext(15)] HRESULT MoodText([out, retval] BSTR *pVal);
00648 [propget, id(25), helpstring(hsIUser_Aliases), helpcontext(15)] HRESULT Aliases([out, retval] BSTR *pVal);
00649 [propget, id(26), helpstring(hsIUser_Timezone), helpcontext(15)] HRESULT Timezone([out, retval] LONG *pVal);
00650 [propget, id(27), helpstring(hsIUser_IsCFActive), helpcontext(15)] HRESULT IsCallForwardActive([out, retval] VARIANT_BOOL *pVal);
00651 [propget, id(28), helpstring(hsIUser_Language), helpcontext(15)] HRESULT Language([out, retval] BSTR *pVal);
00652 [propget, id(29), helpstring(hsIUser_LanguageCode), helpcontext(15)] HRESULT LanguageCod
```

```

e([out, retval] BSTR *pVal);
00653     [propget, id(30), helpstring(hsIUser_V
ideoCapable), helpcontext(15)] HRESULT IsVideoCapa
ble([out, retval] VARIANT_BOOL *pVal);
00654     [propget, id(31), helpstring(hsIUser_S
kypeOut), helpcontext(15)] HRESULT IsSkypeOutConta
ct([out, retval] VARIANT_BOOL *pVal);
00655     [propget, id(32), helpstring(hsIUser_N
umberOfAuthBuddies), helpcontext(15)] HRESULT Numb
erOfAuthBuddies([out, retval] LONG* pVal);
00656     [propput, id(17), helpstring(hsIUser_D
isplayName), helpcontext(15)] HRESULT DisplayName(
[in] BSTR newVal);
00657     [propget, id(35), helpstring(hsIUser_R
ichMoodText), helpcontext(15)] HRESULT RichMoodTex
t([out, retval] BSTR *pVal);
00658     [propget, id(36), helpstring(hsIUser_I
sVmCapable), helpcontext(15)] HRESULT IsVoicemailC
apable([out, retval] VARIANT_BOOL *pVal);
00659     };
00660     [
00661         object,
00662         uuid(EC163200-44EB-483B-907F-A8C1CF56B
8EE),
00663         dual,
00664         nonextensible,
00665         pointer_default(unique),
00666         oleautomation,
00667         helpstring(hsIUserColl),
00668         helpcontext(16)
00669     ]
00670     interface IUserCollection : IDispatch
00671     {
00672         [propget, id(1), helpstring(hsIUserCol
l_Count), helpcontext(16)] HRESULT Count([out, ret
val] LONG* pVal);
00673         [id(2), helpstring(hsIUserColl_Add), h

```

```

elpcontext(16)] HRESULT Add([in] IUser* pUser);
00674     [id(3), helpstring(hsIUserColl_Remove)
, helpcontext(16)] HRESULT Remove([in] long Index)
;
00675     [id(4), helpstring(hsIUserColl_RemoveA
ll), helpcontext(16)] HRESULT RemoveAll(void);
00676     [propget, id(DISPID_VALUE), helpstring
(hsIUserColl_Item), helpcontext(16)] HRESULT Item(
[in] LONG Index, [out, retval] IUser ** pVal);
00677     [propget, id(DISPID_NEWENUM), helpstri
ng(hsIUserColl_NewEnum), helpcontext(16)] HRESULT
_NewEnum([out, retval] IUnknown** pVal);
00678     };
00679     [
00680         object,
00681         uuid(F1B2AD09-1961-4336-A6BF-93010B73D
E72),
00682         dual,
00683         nonextensible,
00684         pointer_default(unique),
00685         oleautomation,
00686         helpstring(hsIParticipant),
00687         helpcontext(37)
00688     ]
00689     interface IParticipant : IDispatch
00690     {
00691         [propget, id(1), helpstring(hsIPartici
pant_Handle), helpcontext(37)] HRESULT Handle([out
, retval] BSTR *pVal);
00692         [propget, id(2), helpstring(hsIPartici
pant_DispName), helpcontext(37)] HRESULT DisplayNa
me([out, retval] BSTR *pVal);
00693         [propget, id(3), helpstring(hsIPartici
pant_CallType), helpcontext(37)] HRESULT CallType(
[out, retval] TCallType *pVal);
00694         [propget, id(4), helpstring(hsIPartici
pant_CallStat), helpcontext(37)] HRESULT CallStatu

```

```

s([out, retval] TCallStatus *pVal);
00695     };
00696     [
00697         object,
00698         uuid(F825FE12-9826-4BD2-BF7E-1ED95B848
0FE),
00699         dual,
00700         nonextensible,
00701         pointer_default(unique),
00702         oleautomation,
00703         helpstring(hsIParticipantColl),
00704         helpcontext(38)
00705     ]
00706     interface IParticipantCollection : IDispa
tch
00707     {
00708         [propget, id(1), helpstring(hsIPartici
pantColl_Count), helpcontext(38)] HRESULT Count([o
ut, retval] LONG* pVal);
00709         [id(2), helpstring(hsIParticipantColl_
Add), helpcontext(38)] HRESULT Add([in] IParticipa
nt* pParticipant);
00710         [id(3), helpstring(hsIParticipantColl_
Remove), helpcontext(38)] HRESULT Remove([in] long
Index);
00711         [id(4), helpstring(hsIParticipantColl_
RemoveAll), helpcontext(38)] HRESULT RemoveAll(void
);
00712         [propget, id(DISPID_VALUE), helpstring
(hsIParticipantColl_Item), helpcontext(38)] HRESUL
T Item([in] LONG Index, [out, retval] IParticipant
** pVal);
00713         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIParticipantColl_NewEnum), helpcontext(38)] H
RESULT _NewEnum([out, retval] IUnknown** pVal);
00714     };
00715     [

```

```

00716         object,
00717         uuid(76036886-436C-405F-A96E-7835CCFB8
00718         2F3),
00719         dual,
00720         nonextensible,
00721         pointer_default(unique),
00722         oleautomation,
00723         helpstring(hsICall),
00724         helpcontext(5)
00725     ]
00726 interface ICall : IDispatch
00727 {
00728     [propget, id(1), helpstring(hsICall_Id
00729     ),helpcontext(5)] HRESULT Id([out, retval] LONG *p
00730     Val);
00731     [propget, id(2), helpstring(hsICall_Ti
00732     mestamp),helpcontext(5)] HRESULT Timestamp([out, r
00733     etval] DATE *pVal);
00734     [propget, id(3), helpstring(hsICall_Pa
00735     rtHandle),helpcontext(5)] HRESULT PartnerHandle([o
00736     ut, retval] BSTR *pVal);
00737     [propget, id(4), helpstring(hsICall_Pa
00738     rtDispName),helpcontext(5)] HRESULT PartnerDisplay
00739     Name([out, retval] BSTR *pVal);
00740     [propget, id(5), helpstring(hsICall_Co
00741     nferenceId),helpcontext(5)] HRESULT ConferenceId([
00742     out, retval] LONG *pVal);
00743     [propget, id(6), helpstring(hsICall_Ty
00744     pe),helpcontext(5)] HRESULT Type([out, retval] TCa
00745     llType *pVal);
00746     [propget, id(7), helpstring(hsICall_St
00747     atus),helpcontext(5)] HRESULT Status([out, retval]
00748     TCallStatus *pVal);
00749     [propput, id(7), helpstring(hsICall_St
00750     atus),helpcontext(5)] HRESULT Status([in] TCallSta
00751     tus newVal);
00752     [propget, id(8), helpstring(hsICall_Fa

```

```
ilReason),helpcontext(5)] HRESULT FailureReason([out, retval] TCallFailureReason *pVal);
00736      [propget, id(9), helpstring(hsICall_Subject),helpcontext(5)] HRESULT Subject([out, retval] BSTR *pVal);
00737      [propget, id(10), helpstring(hsICall_PSTN_Number),helpcontext(5)] HRESULT PstnNumber([out, retval] BSTR *pVal);
00738      [propget, id(11), helpstring(hsICall_Duration),helpcontext(5)] HRESULT Duration([out, retval] LONG *pVal);
00739      [propget, id(12), helpstring(hsICall_PSTN_Status),helpcontext(5)] HRESULT PstnStatus([out, retval] BSTR *pVal);
00740      [propput, id(13), helpstring(hsICall_Seen),helpcontext(5)] HRESULT Seen([in] VARIANT_BOOL newVal);
00741      [id(14), helpstring(hsICall_Hold),helpcontext(5)] HRESULT Hold();
00742      [id(15), helpstring(hsICall_Resume),helpcontext(5)] HRESULT Resume();
00743      [id(16), helpstring(hsICall_Finish),helpcontext(5)] HRESULT Finish();
00744      [id(17), helpstring(hsICall_Answer),helpcontext(5)] HRESULT Answer();
00745      [propput, id(18), helpstring(hsICall_DTMF),helpcontext(5)] HRESULT DTMF([in] BSTR newVal);
00746      [propget, id(19), helpstring(hsICall_Participants),helpcontext(5)] HRESULT Participants([out, retval] IParticipantCollection **pVal);
00747      [id(20), helpstring(hsICall_Join),helpcontext(5)] HRESULT Join([in] LONG Id);
00748      [propget, id(21), helpstring(hsICall_VmDuration),helpcontext(5)] HRESULT VmDuration([out, retval] LONG *pVal);
00749      [propget, id(22), helpstring(hsICall_V
```

```
mAllowDuration),helpcontext(5)] HRESULT VmAllowedDuration([out, retval] LONG *pVal);
00750      [propget, id(23), helpstring(hsICall_V
Status),helpcontext(5)] HRESULT VideoStatus([out,
retval] TCallVideoStatus *pVal);
00751      [propget, id(24), helpstring(hsICall_V
SendStatus),helpcontext(5)] HRESULT VideoSendStatu
s([out, retval] TCallVideoSendStatus *pVal);
00752      [propget, id(25), helpstring(hsICall_V
RecvStatus),helpcontext(5)] HRESULT VideoReceiveSt
atus([out, retval] TCallVideoSendStatus *pVal);
00753      [id(26), helpstring(hsICall_StartVSend
),helpcontext(5)] HRESULT StartVideoSend();
00754      [id(27), helpstring(hsICall_StopVSend)
,helpcontext(5)] HRESULT StopVideoSend();
00755      [id(28), helpstring(hsICall_StartVRecv
),helpcontext(5)] HRESULT StartVideoReceive();
00756      [id(29), helpstring(hsICall_StopVRecv)
,helpcontext(5)] HRESULT StopVideoReceive();
00757      [id(30), helpstring(hsICall_Redirect),
helpcontext(5)] HRESULT RedirectToVoicemail();
00758      [id(31), helpstring(hsICall_Forward),h
elpcontext(5)] HRESULT Forward();
00759      [propget, id(32), helpstring(hsICall_R
ate),helpcontext(5)] HRESULT Rate([out, retval] LO
NG *pVal);
00760      [propget, id(33), helpstring(hsICall_R
ateCurr),helpcontext(5)] HRESULT RateCurrency([out
, retval] BSTR *pVal);
00761      [propget, id(34), helpstring(hsICall_R
atePrecision),helpcontext(5)] HRESULT RatePrecisio
n([out, retval] LONG *pVal);
00762      [propget, id(35), helpstring(hsICall_I
nputDevice),helpcontext(5)] HRESULT InputDevice([i
n] TCallIoDeviceType DeviceType, [out, retval] BST
R* pVal);
00763      [propget, id(35), helpstring(hsICall_I
```



```
nputDevice),helpcontext(5)] HRESULT InputDevice([i
n] TCallIoDeviceType DeviceType, [in] BSTR newVal)
;
00764      [propget, id(36), helpstring(hsICall_0
utputDevice),helpcontext(5)] HRESULT OutputDevice(
[in] TCallIoDeviceType DeviceType, [out, retval] B
STR* pVal);
00765      [propget, id(36), helpstring(hsICall_0
utputDevice),helpcontext(5)] HRESULT OutputDevice(
[in] TCallIoDeviceType DeviceType, [in] BSTR newVa
l);
00766      [propget, id(37), helpstring(hsICall_M
icDevice),helpcontext(5)] HRESULT CaptureMicDevice
([in] TCallIoDeviceType DeviceType, [out, retval]
BSTR* pVal);
00767      [propget, id(37), helpstring(hsICall_M
icDevice),helpcontext(5)] HRESULT CaptureMicDevice
([in] TCallIoDeviceType DeviceType, [in] BSTR newV
al);
00768      [propget, id(38), helpstring(hsICall_I
nputStatus),helpcontext(5)] HRESULT InputStatus([o
ut, retval] VARIANT_BOOL* pVal);
00769      [propget, id(39), helpstring(hsICall_F
orwardedBy),helpcontext(5)] HRESULT ForwardedBy([o
ut, retval] BSTR* pVal);
00770      [propget, id(13), helpstring(hsICall_S
een),helpcontext(5)] HRESULT Seen([out, retval] VA
RIANT_BOOL *pVal);
00771      [propget, id(40), helpstring(hsICall_C
anTransfer),helpcontext(5)] HRESULT CanTransfer([i
n] BSTR Target, [out, retval] VARIANT_BOOL* pVal);
00772      [propget, id(41), helpstring(hsICall_T
ransferStatus),helpcontext(5)] HRESULT TransferSta
tus([out, retval] TCallStatus* pVal);
00773      [propget, id(42), helpstring(hsICall_T
ransferActive),helpcontext(5)] HRESULT TransferAct
ive([out, retval] VARIANT_BOOL* pVal);
```

```

00774      [propget, id(43), helpstring(hsICall_T
ransferredBy),helpcontext(5)] HRESULT TransferredB
y([out, retval] BSTR* pVal);
00775      [propget, id(44), helpstring(hsICall_T
ransferredTo),helpcontext(5)] HRESULT TransferredT
o([out, retval] BSTR* pVal);
00776      [id(45), helpstring(hsICall_Transfer),
helpcontext(5)] HRESULT Transfer([in] BSTR Target)
;
00777      [propget, id(46), helpstring(hsICall_T
argetIdentity),helpcontext(5)] HRESULT TargetIdent
ity([out, retval] BSTR *pVal);
00778      };
00779      [
00780          object,
00781          uuid(72E1EC13-0DF9-48AE-8A31-E2900D85B
A55),
00782          dual,
00783          nonextensible,
00784          pointer_default(unique),
00785          oleautomation,
00786          helpstring(hsICallColl),
00787          helpcontext(6)
00788      ]
00789      interface ICallCollection : IDispatch
00790      {
00791          [propget, id(1), helpstring(hsICallCol
l_Count),helpcontext(6)] HRESULT Count([out, retva
l] LONG* pVal);
00792          [id(2), helpstring(hsICallColl_Add),he
lpcontext(6)] HRESULT Add([in] ICall* pCall);
00793          [id(3), helpstring(hsICallColl_Remove)
,helpcontext(6)] HRESULT Remove([in] long Index);
00794          [id(4), helpstring(hsICallColl_RemoveA
ll),helpcontext(6)] HRESULT RemoveAll(void);
00795          [propget, id(DISPID_VALUE), helpstring
(hsICallColl_Item),helpcontext(6)] HRESULT Item([i

```

```

n] LONG Index, [out, retval] ICall ** pVal);
00796     [propget, id(DISPID_NEWENUM), helpstri
ng(hsICallColl_NewEnum), helpcontext(6)] HRESULT _N
ewEnum([out, retval] IUnknown** pVal);
00797     };
00798
00799     interface IChat;
00800     [
00801         object,
00802         uuid(4CFF5C70-3C95-4566-824A-FA164586D
535),
00803         dual,
00804         nonextensible,
00805         pointer_default(unique),
00806         oleautomation,
00807         helpstring(hsIChatMessage),
00808         helpcontext(11)
00809     ]
00810     interface IChatMessage : IDispatch
00811     {
00812         [propget, id(1), helpstring(hsIChatMsg
_Id), helpcontext(11)] HRESULT Id([out, retval] LO
NG *pVal);
00813         [propget, id(2), helpstring(hsIChatMsg
_Timestamp), helpcontext(11)] HRESULT Timestamp([o
ut, retval] DATE *pVal);
00814         [propget, id(3), helpstring(hsIChatMsg
_FromHandle), helpcontext(11)] HRESULT FromHandle(
[out, retval] BSTR *pVal);
00815         [propget, id(4), helpstring(hsIChatMsg
_FromDispName), helpcontext(11)] HRESULT FromDispl
ayName([out, retval] BSTR *pVal);
00816         [propget, id(5), helpstring(hsIChatMsg
_Type), helpcontext(11)] HRESULT Type([out, retval
] TChatMessageType *pVal);
00817         [propget, id(6), helpstring(hsIChatMsg
_Status), helpcontext(11)] HRESULT Status([out, re

```

```

tval] TChatMessageStatus *pVal);
00818      [propget, id(7), helpstring(hsIChatMsg
_LLeaveReason), helpcontext(11)] HRESULT LeaveReason
n([out, retval] TChatLeaveReason *pVal);
00819      [propget, id(8), helpstring(hsIChatMsg
_Body), helpcontext(11)] HRESULT Body([out, retval
] BSTR *pVal);
00820      [propget, id(9), helpstring(hsIChatMsg
_ChatName), helpcontext(11)] HRESULT ChatName([out
, retval] BSTR *pVal);
00821      [propget, id(10), helpstring(hsIChatMs
g_Users), helpcontext(11)] HRESULT Users([out, ret
val] IUserCollection **pVal);
00822      [propput, id(11), helpstring(hsIChatMs
g_Seen), helpcontext(11)] HRESULT Seen([in] VARIAN
T_BOOL newVal);
00823      [propget, id(12), helpstring(hsIChatMs
g_Chat), helpcontext(11)] HRESULT Chat([out, retva
l] IChat** pVal);
00824      [propget, id(13), helpstring(hsIChatMs
g_Sender), helpcontext(11)] HRESULT Sender([out, r
etval] IUser* *pVal);
00825      [propget, id(14), helpstring(hsIChatMs
g_EditedBy), helpcontext(11)] HRESULT EditedBy([ou
t, retval] BSTR* pVal);
00826      [propget, id(15), helpstring(hsIChatMs
g_EditTime), helpcontext(11)] HRESULT EditedTimest
amp([out, retval] DATE* pVal);
00827      [propput, id(8), helpstring(hsIChatMsg
_Body), helpcontext(11)] HRESULT Body([in] BSTR ne
wVal);
00828      [propget, id(16), helpstring(hsIChatMs
g_Role), helpcontext(11)] HRESULT Role([out, retva
l] TChatMemberRole* pVal);
00829      [propget, id(17), helpstring(hsIChatMs
g_Options), helpcontext(11)] HRESULT Options([out,
retval] LONG* pVal);

```

```

00830         [propget, id(18), helpstring(hsIChatMsg
g_Editable), helpcontext(11)] HRESULT IsEditable([
out, retval] VARIANT_BOOL* pVal);
00831     };
00832     [
00833         object,
00834         uuid(9511117A-7BCA-4669-BE96-6EA686697
5C1),
00835         dual,
00836         nonextensible,
00837         pointer_default(unique),
00838         oleautomation,
00839         helpstring(hsIChatMessageColl),
00840         helpcontext(12)
00841     ]
00842     interface IChatMessageCollection : IDispa
tch
00843     {
00844         [propget, id(1), helpstring(hsIChatMsg
Coll_Count), helpcontext(12)] HRESULT Count([out, r
etval] LONG* pVal);
00845         [id(2), helpstring(hsIChatMsgColl_Add)
, helpcontext(12)] HRESULT Add([in] IChatMessage* p
Item);
00846         [id(3), helpstring(hsIChatMsgColl_Remo
ve), helpcontext(12)] HRESULT Remove([in] long Inde
x);
00847         [id(4), helpstring(hsIChatMsgColl_Remo
veAll), helpcontext(12)] HRESULT RemoveAll(void);
00848         [propget, id(DISPID_VALUE), helpstring
(hsIChatMsgColl_Item), helpcontext(12)] HRESULT Ite
m([in] LONG Index, [out, retval] IChatMessage ** p
Val);
00849         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIChatMsgColl_NewEnum), helpcontext(12)] HRESUL
T _NewEnum([out, retval] IUnknown** pVal);
00850     };

```

```

00851
00852     [
00853         object,
00854         uuid(9E9F0473-94C2-4FDD-85D2-43E347876
00855         5F9),
00856         dual,
00857         nonextensible,
00858         pointer_default(unique),
00859         oleautomation,
00860         helpstring(hsIChatMember),
00861         helpcontext(85)
00862     ]
00863     interface IChatMember : IDispatch
00864     {
00865         [propget, id(1), helpstring(hsIChatMem
00866         ber_Id), helpcontext(85)] HRESULT Id([out, retval]
00867         BSTR* pVal);
00868         [propget, id(2), helpstring(hsIChatMem
00869         ber_Handle), helpcontext(85)] HRESULT Handle([out,
00870         retval] BSTR* pVal);
00871         [propget, id(3), helpstring(hsIChatMem
00872         ber_Role), helpcontext(85)] HRESULT Role([out, ret
00873         val] TChatMemberRole* pVal);
00874         [propput, id(3), helpstring(hsIChatMem
00875         ber_Role), helpcontext(85)] HRESULT Role([in] TCha
00876         tMemberRole newVal);
00877         [propget, id(4), helpstring(hsIChatMem
00878         ber_CanSetRole), helpcontext(85)] HRESULT CanSetRo
00879         leTo([in] TChatMemberRole Role, [out, retval] VARI
00880         ANT_BOOL* pVal);
00881         [propget, id(5), helpstring(hsIChatMem
00882         ber_IsActive), helpcontext(85)] HRESULT IsActive([
00883         out, retval] VARIANT_BOOL* pVal);
00884         [propget, id(6), helpstring(hsIChatMem
00885         ber_Chat), helpcontext(85)] HRESULT Chat([out, ret
00886         val] IChat** pVal);
00887     };

```

```

00872
00873     [
00874         object,
00875         uuid(8FFC686B-2E5E-4763-8B03-FF4FD3A0C
00876         4DA),
00877         dual,
00878         pointer_default(unique),
00879         helpstring(hsIChatMemberCollection),
00880         helpcontext(86)
00881     ]
00882 interface IChatMemberCollection : IDispatch
00883 {
00884     [propget, id(1), helpstring(hsIChatMemberCollection_Count),helpcontext(86)] HRESULT Count([out, retval] LONG* pVal);
00885     [propget, id(DISPID_VALUE), helpstring(hsIChatMemberCollection_Item),helpcontext(86)] HRESULT Item([in] LONG Index, [out, retval] IChatMember** pVal);
00886     [propget, id(DISPID_NEWENUM), helpstring(hsIChatMemberCollection_NewEnum),helpcontext(86)] HRESULT _NewEnum([out, retval] IUnknown** pVal);
00887 };
00888 [
00889     object,
00890     uuid(8C24106C-3368-4580-93E5-5ED4ACCDE
00891     306),
00892     dual,
00893     nonextensible,
00894     pointer_default(unique),
00895     oleautomation,
00896     helpstring(hsIChat),
00897     helpcontext(31)
00898 ]

```

```
00898     interface IChat : IDispatch
00899     {
00900         [propget, id(1), helpstring(hsIChat_Name), helpcontext(31)] HRESULT Name([out, retval] BSTR *pVal);
00901         [propget, id(2), helpstring(hsIChat_Messages), helpcontext(31)] HRESULT Messages([out, retval] IChatMessageCollection **pVal);
00902         [propget, id(3), helpstring(hsIChat_Timestamp), helpcontext(31)] HRESULT Timestamp([out, retval] DATE *pVal);
00903         [propget, id(4), helpstring(hsIChat_Adder), helpcontext(31)] HRESULT Adder([out, retval] IUser **pVal);
00904         [propget, id(5), helpstring(hsIChat_Status), helpcontext(31)] HRESULT Status([out, retval] TChatStatus *pVal);
00905         [propget, id(6), helpstring(hsIChat_Posters), helpcontext(31)] HRESULT Posters([out, retval] IUserCollection **pVal);
00906         [propget, id(7), helpstring(hsIChat_Members), helpcontext(31)] HRESULT Members([out, retval] IUserCollection **pVal);
00907         [propget, id(8), helpstring(hsIChat_Topic), helpcontext(31)] HRESULT Topic([out, retval] BSTR *pVal);
00908         [propput, id(8), helpstring(hsIChat_Topic), helpcontext(31)] HRESULT Topic([in] BSTR newVal);
00909         [propget, id(9), helpstring(hsIChat_ActiveMembers), helpcontext(31)] HRESULT ActiveMembers([out, retval] IUserCollection **pVal);
00910         [propget, id(10), helpstring(hsIChat_FriendlyName), helpcontext(31)] HRESULT FriendlyName([out, retval] BSTR *pVal);
00911         [id(11), helpstring(hsIChat_OpenWindow), helpcontext(31)] HRESULT OpenWindow();
```



```
00912      [id(12), helpstring(hsIChat_SendMessage), helpcontext(31)] HRESULT SendMessage([in] BSTR MessageText, [out, retval] IChatMessage **pVal);
00913      [id(13), helpstring(hsIChat_Leave), helpcontext(31)] HRESULT Leave();
00914      [id(14), helpstring(hsIChat_AddMembers), helpcontext(31)] HRESULT AddMembers([in] IUserCollection* pMembers);
00915      [propget, id(15), helpstring(hsIChat_RecentMessages), helpcontext(31)] HRESULT RecentMessages([out, retval] IChatMessageCollection **pVal);
;
00916      [propget, id(16), helpstring(hsIChat_Bookmarked)] HRESULT Bookmarked([out, retval] VARIANT_BOOL *pVal);
00917      [id(17), helpstring(hsIChat_Bookmark)] HRESULT Bookmark();
00918      [id(18), helpstring(hsIChat_Bookmark)] HRESULT Unbookmark();
00919      [propget, id(19), helpstring(hsIChat_TopicXML), helpcontext(31)] HRESULT TopicXML([out, retval] BSTR *pVal);
00920      [propput, id(19), helpstring(hsIChat_TopicXML), helpcontext(31)] HRESULT TopicXML([in] BSTR newVal);
00921      [propget, id(20), helpstring(hsIChat_MemberObjs), helpcontext(31)] HRESULT MemberObjects([out, retval] IChatMemberCollection **pVal);
00922      [propget, id(21), helpstring(hsIChat_Blob), helpcontext(31)] HRESULT Blob([out, retval] BSTR* pVal);
00923      [propget, id(22), helpstring(hsIChat_Options), helpcontext(31)] HRESULT Options([out, retval] LONG* pVal);
00924      [propput, id(22), helpstring(hsIChat_Options), helpcontext(31)] HRESULT Options([in] LONG newVal);
```

```
00925      [propget, id(23), helpstring(hsIChat_P
asswdhint), helpcontext(31)] HRESULT PasswordHint(
[out, retval] BSTR* pVal);
00926      [id(24), helpstring(hsIChat_Password),
helpcontext(31)] HRESULT SetPassword([in] BSTR Pa
ssword, [in, defaultvalue("")] BSTR Hint);
00927      [propget, id(25), helpstring(hsIChat_G
uidelines), helpcontext(31)] HRESULT GuideLines([o
ut, retval] BSTR* pVal);
00928      [propput, id(25), helpstring(hsIChat_G
uidelines), helpcontext(31)] HRESULT GuideLines([i
n] BSTR newVal);
00929      [propget, id(26), helpstring(hsIChat_D
escription), helpcontext(31)] HRESULT Description(
[out, retval] BSTR* pVal);
00930      [propput, id(26), helpstring(hsIChat_D
escription), helpcontext(31)] HRESULT Description(
[in] BSTR newVal);
00931      [propget, id(27), helpstring(hsIChat_D
ialogPartner), helpcontext(31)] HRESULT DialogPart
ner([out, retval] BSTR* pVal);
00932      [propget, id(28), helpstring(hsIChat_A
ctiveTime), helpcontext(31)] HRESULT ActivityTimes
tamp([out, retval] DATE* pVal);
00933      [propget, id(29), helpstring(hsIChat_M
yRole), helpcontext(31)] HRESULT MyRole([out, retv
al] TChatMemberRole* pVal);
00934      [propget, id(30), helpstring(hsIChat_A
pplicants), helpcontext(31)] HRESULT Applicants([o
ut, retval] IUserCollection** pVal);
00935      [id(31), helpstring(hsIChat_Join), hel
pcontext(31)] HRESULT Join(void);
00936      [id(32), helpstring(hsIChat_Kick), hel
pcontext(31)] HRESULT Kick([in] BSTR Handle);
00937      [id(33), helpstring(hsIChat_Kickban),
helpcontext(31)] HRESULT KickBan([in] BSTR Handle)
;
```

```

00938      [id(34), helpstring(hsIChat_Disband),
helpcontext(31)] HRESULT Disband(void);
00939      [id(35), helpstring(hsIChat_EnterPassw
d), helpcontext(31)] HRESULT EnterPassword([in] BS
TR Password);
00940      [id(36), helpstring(hsIChat_ClearRecen
tMsg), helpcontext(31)] HRESULT ClearRecentMessage
s(void);
00941      [id(37), helpstring(hsIChat_AcceptAdd)
, helpcontext(31)] HRESULT AcceptAdd(void);
00942      [propput, id(38), helpstring(hsIChat_A
lertString), helpcontext(31)] HRESULT AlertString(
[in] BSTR newVal);
00943      [propget, id(39), helpstring(hsIChat_T
ype), helpcontext(31)] HRESULT Type([out, retval]
TChatType* pVal);
00944      [propget, id(40), helpstring(hsIChat_M
yStatus), helpcontext(31)] HRESULT MyStatus([out,
retval] TChatMyStatus* pVal);
00945      };
00946      [
00947          object,
00948          uuid(052A4165-010A-457D-A874-E661B6322
846),
00949          dual,
00950          nonextensible,
00951          pointer_default(unique),
00952          oleautomation,
00953          helpstring(hsIChatColl),
00954          helpcontext(32)
00955      ]
00956      interface IChatCollection : IDispatch
00957      {
00958          [propget, id(1), helpstring(hsIChatCol
l_Count),helpcontext(32)] HRESULT Count([out, retv
al] LONG* pVal);
00959          [id(2), helpstring(hsIChatColl_Add),he

```

```

lpcontext(32)] HRESULT Add([in] IChat* pItem);
00960      [id(3), helpstring(hsIChatColl_Remove)
,helpcontext(32)] HRESULT Remove([in] long Index);
00961      [id(4), helpstring(hsIChatColl_RemoveA
ll),helpcontext(32)] HRESULT RemoveAll(void);
00962      [propget, id(DISPID_VALUE), helpstring
(hsIChatColl_Item),helpcontext(32)] HRESULT Item([
in] LONG Index, [out, retval] IChat ** pVal);
00963      [propget, id(DISPID_NEWENUM), helpstri
ng(hsIChatColl_NewEnum),helpcontext(32)] HRESULT _
NewEnum([out, retval] IUnknown** pVal);
00964      };
00965      [
00966          object,
00967          uuid(AEB48497-5090-479C-8BE0-BD5159715
6A1),
00968          dual,
00969          nonextensible,
00970          pointer_default(unique),
00971          oleautomation,
00972          helpstring(hsIConference),
00973          helpcontext(39)
00974      ]
00975      interface IConference : IDispatch
00976      {
00977          [propget, id(1), helpstring(hsIConfere
nce_Id), helpcontext(39)] HRESULT Id([out, retval]
LONG *pVal);
00978          [propget, id(2), helpstring(hsIConfere
nce_Calls), helpcontext(39)] HRESULT Calls([out, r
etval] ICallCollection **pVal);
00979          [propget, id(3), helpstring(hsIConfere
nce_ActiveCalls), helpcontext(39)] HRESULT ActiveC
alls([out, retval] ICallCollection **pVal);
00980          [id(4), helpstring(hsIConference_Hold)
, helpcontext(39)] HRESULT Hold();
00981          [id(5), helpstring(hsIConference_Resum

```

```

e), helpcontext(39)] HRESULT Resume();
00982     [id(6), helpstring(hsIConference_Finis
h), helpcontext(39)] HRESULT Finish();
00983     };
00984     [
00985         object,
00986         uuid(F3E0C529-7D4F-4AF5-B501-27D25D4D2
825),
00987         dual,
00988         nonextensible,
00989         pointer_default(unique),
00990         oleautomation,
00991         helpstring(hsIConferenceColl),
00992         helpcontext(40)
00993     ]
00994     interface IConferenceCollection : IDispat
ch
00995     {
00996         [propget, id(1), helpstring(hsIConfere
nceColl_Count), helpcontext(40)] HRESULT Count([ou
t, retval] LONG* pVal);
00997         [id(2), helpstring(hsIConferenceColl_A
dd), helpcontext(40)] HRESULT Add([in] IConference
* pItem);
00998         [id(3), helpstring(hsIConferenceColl_R
emove), helpcontext(40)] HRESULT Remove([in] long
Index);
00999         [id(4), helpstring(hsIConferenceColl_R
emoveAll), helpcontext(40)] HRESULT RemoveAll(void
);
01000         [propget, id(DISPID_VALUE), helpstring
(hsIConferenceColl_Item), helpcontext(40)] HRESULT
Item([in] LONG Index, [out, retval] IConference *
* pVal);
01001         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIConferenceColl_NewEnum), helpcontext(40)] HR
ESULT _NewEnum([out, retval] IUnknown** pVal);

```

```

01002     };
01003     [
01004         object,
01005         uuid(8D82F88A-B307-4074-8ED5-11C3BD213
452),
01006         dual,
01007         nonextensible,
01008         pointer_default(unique),
01009         oleautomation,
01010         helpstring(hsIConversion),
01011         helpcontext(10)
01012     ]
01013     interface IConversion : IDispatch
01014     {
01015         [id(1), helpstring(hsIConversion_Onlin
eStatus), helpcontext(10)] HRESULT OnlineStatusToT
ext([in] TOnlineStatus status, [out, retval] BSTR
*pVal);
01016         [id(2), helpstring(hsIConversion_TxOnl
ineStatus), helpcontext(10)] HRESULT TextToOnlineS
tatus([in] BSTR Text, [out, retval] TOnlineStatus
*pVal);
01017         [id(3), helpstring(hsIConversion_Buddy
Status), helpcontext(10)] HRESULT BuddyStatusToTex
t([in] TBuddyStatus status, [out, retval] BSTR *pV
al);
01018         [id(4), helpstring(hsIConversion_TxBud
dyStatus), helpcontext(10)] HRESULT TextToBuddySta
tus([in] BSTR Text, [out, retval] TBuddyStatus *pV
al);
01019         [id(5), helpstring(hsIConversion_CallS
tatus), helpcontext(10)] HRESULT CallStatusToText(
[in] TCallStatus Status, [out, retval] BSTR *pVal)
;
01020         [id(6), helpstring(hsIConversion_TxCa
llStatus), helpcontext(10)] HRESULT TextToCallStatu
s([in] BSTR Text, [out, retval] TCallStatus *pVal)

```

```
;
01021      [id(7), helpstring(hsIConversion_CallT
ype), helpcontext(10)] HRESULT CallTypeToText([in]
TCallType CallType, [out, retval] BSTR *pVal);
01022      [id(8), helpstring(hsIConversion_TxCa
llType), helpcontext(10)] HRESULT TextToCallType([i
n] BSTR Text, [out, retval] TCallType *pVal);
01023      [id(9), helpstring(hsIConversion_UserS
ex), helpcontext(10)] HRESULT UserSexToText([in] T
UserSex Sex, [out, retval] BSTR *pVal);
01024      [id(10), helpstring(hsIConversion_TxUs
erSex), helpcontext(10)] HRESULT TextToUserSex([in
] BSTR Text, [out, retval] TUserSex *pVal);
01025      [id(11), helpstring(hsIConversion_Conn
Status), helpcontext(10)] HRESULT ConnectionStatus
ToText([in] TConnectionStatus Status, [out, retval
] BSTR *pVal);
01026      [id(12), helpstring(hsIConversion_TxCo
nnStatus), helpcontext(10)] HRESULT TextToConnecti
onStatus([in] BSTR Text, [out, retval] TConnection
Status *pVal);
01027      [id(13), helpstring(hsIConversion_User
Status), helpcontext(10)] HRESULT UserStatusToText
([in] TUserStatus Status, [out, retval] BSTR *pVal
);
01028      [id(14), helpstring(hsIConversion_TxUs
erStatus), helpcontext(10)] HRESULT TextToUserStat
us([in] BSTR Text, [out, retval] TUserStatus *pVal
);
01029      [id(15), helpstring(hsIConversion_Fail
Reason), helpcontext(10)] HRESULT CallFailureReaso
nToText([in] TCallFailureReason reason, [out, retv
al] BSTR *pVal);
01030      [id(16), helpstring(hsIConversion_Atta
chStatus), helpcontext(10)] HRESULT AttachmentStat
usToText([in] TAttachmentStatus status, [out, retv
al] BSTR *pVal);
```

```
01031      [id(17), helpstring(hsIConversion_LeaveReason), helpcontext(10)] HRESULT ChatLeaveReasonToText([in] TChatLeaveReason reason, [out, retval] BSTR *pVal);
01032      [id(18), helpstring(hsIConversion_ChatStatus), helpcontext(10)] HRESULT ChatStatusToText([in] TChatStatus status, [out, retval] BSTR *pVal);
01033      [id(19), helpstring(hsIConversion_VmType), helpcontext(10)] HRESULT VoicemailTypeToText([in] TVoicemailType type, [out, retval] BSTR *pVal);
01034      [id(20), helpstring(hsIConversion_VmStatus), helpcontext(10)] HRESULT VoicemailStatusToText([in] TVoicemailStatus status, [out, retval] BSTR *pVal);
01035      [id(21), helpstring(hsIConversion_TxVmStatus), helpcontext(10)] HRESULT TextToVoicemailsStatus([in] BSTR Text, [out, retval] TVoicemailStatus* pVal);
01036      [id(22), helpstring(hsIConversion_VmFailureReason), helpcontext(10)] HRESULT VoicemailFailureReasonToText([in] TVoicemailFailureReason code, [out, retval] BSTR *pVal);
01037      [id(23), helpstring(hsIConversion_MessStatus), helpcontext(10)] HRESULT ChatMessageStatusToText([in] TChatMessageStatus Status, [out, retval] BSTR *pVal);
01038      [id(24), helpstring(hsIConversion_TxMessStatus), helpcontext(10)] HRESULT TextToChatMessageStatus([in] BSTR Text, [out, retval] TChatMessageStatus *pVal);
01039      [id(25), helpstring(hsIConversion_MessType), helpcontext(10)] HRESULT ChatMessageTypeToText([in] TChatMessageType Type, [out, retval] BSTR *pVal);
01040      [id(26), helpstring(hsIConversion_TxMe
```



```
ssType), helpcontext(10)] HRESULT TextToChatMessage  
Type([in] BSTR Text, [out, retval] TChatMessageTy  
pe *pVal);  
01041 [id(27), helpstring(hsIConversion_TxAt  
tachStat), helpcontext(10)] HRESULT TextToAttachme  
ntStatus([in] BSTR Text, [out, retval] TAttachment  
Status *pVal);  
01042 [id(28), helpstring(hsIConversion_GrpT  
ype), helpcontext(10)] HRESULT GroupTypeToText([in  
] TGroupType Type, [out, retval] BSTR *pVal);  
01043 [id(29), helpstring(hsIConversion_TxGr  
pType), helpcontext(10)] HRESULT TextToGroupType([  
in] BSTR Text, [out, retval] TGroupType *pVal);  
01044 [id(30), helpstring(hsIConversion_Vide  
oStatus), helpcontext(10)] HRESULT CallVideoStatus  
ToText([in] TCallVideoStatus Status, [out, retval]  
BSTR *pVal);  
01045 [id(31), helpstring(hsIConversion_Vide  
oSendStatus), helpcontext(10)] HRESULT CallVideoSe  
ndStatusToText([in] TCallVideoSendStatus Status, [  
out, retval] BSTR *pVal);  
01046 [propget, id(32), helpstring(hsIConver  
sion_Language), helpcontext(10)] HRESULT Language(  
[out, retval] BSTR *pVal);  
01047 [propput, id(32), helpstring(hsIConver  
sion_Language), helpcontext(10)] HRESULT Language(  
[in] BSTR newVal);  
01048 [id(33), helpstring(hsIConversion_SmsS  
tatus), helpcontext(10)] HRESULT SmsMessageStatusT  
oText([in] TSmsMessageStatus status, [out, retval]  
BSTR *pVal);  
01049 [id(34), helpstring(hsIConversion_SmsT  
ype), helpcontext(10)] HRESULT SmsMessageTypeToTex  
t([in] TSmsMessageType type, [out, retval] BSTR *p  
Val);  
01050 [id(35), helpstring(hsIConversion_SmsT  
argetStatus), helpcontext(10)] HRESULT SmsTargetSt
```

```

atusToText([in] TSmsTargetStatus status, [out, ret
val] BSTR *pVal);
01051     };
01052     [
01053         object,
01054         uuid(120B425E-6313-4924-B5A9-3E9F2E444
A00),
01055         dual,
01056         nonextensible,
01057         pointer_default(unique),
01058         oleautomation,
01059         helpstring(hsIVoicemail),
01060         helpcontext(51)
01061     ]
01062     interface IVoicemail : IDispatch
01063     {
01064         [propget, id(1), helpstring(hsIVoicema
il_Type), helpcontext(51)] HRESULT Type([out, retv
al] TVoicemailType *pVal);
01065         [propget, id(2), helpstring(hsIVoicema
il_PartHandle), helpcontext(51)] HRESULT PartnerHa
ndle([out, retval] BSTR *pVal);
01066         [propget, id(3), helpstring(hsIVoicema
il_PartDispName), helpcontext(51)] HRESULT Partner
DisplayName([out, retval] BSTR *pVal);
01067         [propget, id(4), helpstring(hsIVoicema
il_Status), helpcontext(51)] HRESULT Status([out,
retval] TVoicemailStatus *pVal);
01068         [propget, id(5), helpstring(hsIVoicema
il_FailureReason), helpcontext(51)] HRESULT Failur
eReason([out, retval] TVoicemailFailureReason *pVa
l);
01069         [propget, id(6), helpstring(hsIVoicema
il_Timestamp), helpcontext(51)] HRESULT Timestamp(
[out, retval] DATE *pVal);
01070         [propget, id(7), helpstring(hsIVoicema
il_Duration), helpcontext(51)] HRESULT Duration([o

```

```

ut, retval] LONG *pVal);
01071      [propget, id(8), helpstring(hsIVoicemail_
AllowedDuration), helpcontext(51)] HRESULT Allo
wedDuration([out, retval] LONG *pVal);
01072      [propget, id(9), helpstring(hsIVoicemail_
Id), helpcontext(51)] HRESULT Id([out, retval]
LONG *pVal);
01073      [id(10), helpstring(hsIVoicemail_Open)
, helpcontext(51)] HRESULT Open();
01074      [id(11), helpstring(hsIVoicemail_Start
Play), helpcontext(51)] HRESULT StartPlayback();
01075      [id(12), helpstring(hsIVoicemail_StopP
lay), helpcontext(51)] HRESULT StopPlayback();
01076      [id(13), helpstring(hsIVoicemail_Uploa
d), helpcontext(51)] HRESULT Upload();
01077      [id(14), helpstring(hsIVoicemail_Downl
oad), helpcontext(51)] HRESULT Download();
01078      [id(15), helpstring(hsIVoicemail_Start
Rec), helpcontext(51)] HRESULT StartRecording();
01079      [id(16), helpstring(hsIVoicemail_StopR
ec), helpcontext(51)] HRESULT StopRecording();
01080      [id(17), helpstring(hsIVoicemail_Delet
e), helpcontext(51)] HRESULT Delete();
01081      [id(18), helpstring(hsIVoicemail_PlayI
nCall), helpcontext(51)] HRESULT StartPlaybackInCa
ll();
01082      [id(19), helpstring(hsIVoicemail_Unpla
yed), helpcontext(51)] HRESULT SetUnplayed();
01083      };
01084      [
01085          object,
01086          uuid(21494848-BBD5-4192-BE32-57D1A19D8
E94),
01087          dual,
01088          nonextensible,
01089          pointer_default(unique),
01090          oleautomation,

```

```

01091         helpstring(hsIVoicemailColl),
01092         helpcontext(52)
01093     ]
01094 ]
01095 interface IVoicemailCollection : IDispatch
01096     {
01097         [propget, id(1), helpstring(hsIVoicemailColl_Count),helpcontext(52)] HRESULT Count([out,
01098         retval] LONG* pVal);
01099         [id(2), helpstring(hsIVoicemailColl_Add),helpcontext(52)] HRESULT Add([in] IVoicemail* p
01100         Item);
01101         [id(3), helpstring(hsIVoicemailColl_Remove),helpcontext(52)] HRESULT Remove([in] long In
01102         dex);
01103         [id(4), helpstring(hsIVoicemailColl_RemoveAll),helpcontext(52)] HRESULT RemoveAll(void);
01104         [propget, id(DISPID_VALUE), helpstring
01105         (hsIVoicemailColl_Item),helpcontext(52)] HRESULT I
01106         tem([in] LONG Index, [out, retval] IVoicemail ** p
01107         Val);
01108         [propget, id(DISPID_NEWENUM), helpstri
01109         ng(hsIVoicemailColl_NewEnum),helpcontext(52)] HRES
01110         ULT _NewEnum([out, retval] IUnknown** pVal);
01111     };
01112 [
01113     object,
01114     uuid(E072F615-F833-4BCA-B473-CEE44D302
01115     030),
01116     dual,
01117     nonextensible,
01118     pointer_default(unique),
01119     oleautomation,
01120     helpstring(hsIApplicationStream),
01121     helpcontext(56)
01122 ]

```

```

01114     interface IApplicationStream : IDispatch
01115     {
01116         [propget, id(1), helpstring(hsIAppStm_
AppName), helpcontext(56)] HRESULT ApplicationName
([out, retval] BSTR *pVal);
01117         [propget, id(2), helpstring(hsIAppStm_
Handle), helpcontext(56)] HRESULT Handle([out, ret
val] BSTR *pVal);
01118         [id(3), helpstring(hsIAppStm_Read), he
lpcontext(56)] HRESULT Read([out, retval] BSTR *pT
ext);
01119         [id(4), helpstring(hsIAppStm_Write), h
elpcontext(56)] HRESULT Write([in] BSTR Text);
01120         [id(5), helpstring(hsIAppStm_SendData)
, helpcontext(56)] HRESULT SendDatagram([in] BSTR
Text);
01121         [propget, id(6), helpstring(hsIAppStm_
DataLen), helpcontext(56)] HRESULT DataLength([out
, retval] LONG *pVal);
01122         [id(7), helpstring(hsIAppStm_Disconnec
t), helpcontext(56)] HRESULT Disconnect();
01123         [propget, id(8), helpstring(hsIAppStm_
PartnerHandle), helpcontext(56)] HRESULT PartnerHa
ndle([out, retval] BSTR *pVal);
01124     };
01125     [
01126         object,
01127         uuid(D8F1BCC4-2A73-47B9-83B3-D4D299CA5
BC6),
01128         dual,
01129         nonextensible,
01130         pointer_default(unique),
01131         oleautomation,
01132         helpstring(hsIApplicationStreamColl),
01133         helpcontext(58)
01134     ]
01135     interface IApplicationStreamCollection :

```

```

IDispatch
01136     {
01137         [propget, id(1), helpstring(hsIAppStmC
oll_Count),helpcontext(58)] HRESULT Count([out, re
tval] LONG* pVal);
01138         [id(2), helpstring(hsIAppStmColl_Add),
helpcontext(58)] HRESULT Add([in] IApplicationStre
am* pItem);
01139         [id(3), helpstring(hsIAppStmColl_Remov
e),helpcontext(58)] HRESULT Remove([in] long Index
);
01140         [id(4), helpstring(hsIAppStmColl_Remov
eAll),helpcontext(58)] HRESULT RemoveAll(void);
01141         [propget, id(DISPID_VALUE), helpstring
(hsIAppStmColl_Item),helpcontext(58)] HRESULT Item
([in] LONG Index, [out, retval] IApplicationStream
\*\* pVal);
01142         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIAppStmColl_NewEnum),helpcontext(58)] HRESULT
_NewEnum([out, retval] IUnknown** pVal);
01143     };
01144     [
01145         object,
01146         uuid(F9ADA565-2FAD-424D-878C-7AD12DF9E
617),
01147         dual,
01148         nonextensible,
01149         pointer_default(unique),
01150         oleautomation,
01151         helpstring(hsIApplication),
01152         helpcontext(14)
01153     ]
01154     interface IApplication : IDispatch
01155     {
01156         [propget, id(1), helpstring(hsIApp_Nam
e),helpcontext(14)] HRESULT Name([out, retval] BST
R *pVal);

```

```

01157      [id(2), helpstring(hsIApp_Create),help
context(14)] HRESULT Create();
01158      [id(3), helpstring(hsIApp_Delete),help
context(14)] HRESULT Delete();
01159      [propget, id(4), helpstring(hsIApp_Str
eams),helpcontext(14)] HRESULT Streams([out, retva
l] IApplicationStreamCollection **pVal);
01160      [propget, id(5), helpstring(hsIApp_Con
nectableUsers),helpcontext(14)] HRESULT Connectabl
eUsers([out, retval] IUserCollection **pVal);
01161      [propget, id(6), helpstring(hsIApp_Con
nectingUsers),helpcontext(14)] HRESULT ConnectingU
sers([out, retval] IUserCollection **pVal);
01162      [id(7), helpstring(hsIApp_Connect),hel
pcontext(14)] HRESULT Connect([in] BSTR Username,
[in, defaultvalue(0)] VARIANT_BOOL WaitConnected);
01163      [propget, id(8), helpstring(hsIApp_Sen
dStreams),helpcontext(14)] HRESULT SendingStreams(
[out, retval] IApplicationStreamCollection **pVal)
;
01164      [propget, id(9), helpstring(hsIApp_Rec
vStreams),helpcontext(14)] HRESULT ReceivedStreams
([out, retval] IApplicationStreamCollection **pVal
);
01165      [id(10), helpstring(hsIApp_SendDatagra
m),helpcontext(14)] HRESULT SendDatagram([in] BSTR
Text, [in, defaultvalue(0)] IApplicationStreamCol
lection* pStreams);
01166      };
01167      [
01168      object,
01169      uuid(B9A3D72F-04C1-4878-993C-D89F83E20
569),
01170      dual,
01171      nonextensible,
01172      pointer_default(unique),
01173      oleautomation,

```

```
01174         helpstring(hsIProfile),
01175         helpcontext(60)
01176     ]
01177     interface IProfile : IDispatch
01178     {
01179         [propget, id(1), helpstring(hsIProfile
01180         _Fullname), helpcontext(60)] HRESULT FullName([out
01181         , retval] BSTR *pVal);
01182         [propput, id(1), helpstring(hsIProfile
01183         _Fullname), helpcontext(60)] HRESULT FullName([in]
01184         BSTR newVal);
01185         [propget, id(2), helpstring(hsIProfile
01186         _Birthday), helpcontext(60)] HRESULT Birthday([out
01187         , retval] BSTR *pVal);
01188         [propput, id(2), helpstring(hsIProfile
01189         _Birthday), helpcontext(60)] HRESULT Birthday([in]
01190         BSTR newVal);
01191         [propget, id(3), helpstring(hsIProfile
01192         _Sex), helpcontext(60)] HRESULT Sex([out, retval]
01193         TUserSex *pVal);
01194         [propput, id(3), helpstring(hsIProfile
01195         _Sex), helpcontext(60)] HRESULT Sex([in] TUserSex
01196         newVal);
01197         [propget, id(4), helpstring(hsIProfile
01198         _Language), helpcontext(60)] HRESULT Languages([ou
01199         t, retval] BSTR *pVal);
01200         [propput, id(4), helpstring(hsIProfile
01201         _Language), helpcontext(60)] HRESULT Languages([in
01202         ] BSTR newVal);
01203         [propget, id(5), helpstring(hsIProfile
01204         _Country), helpcontext(60)] HRESULT Country([out,
01205         retval] BSTR *pVal);
01206         [propput, id(5), helpstring(hsIProfile
01207         _Country), helpcontext(60)] HRESULT Country([in] B
01208         STR newVal);
01209         [propget, id(6), helpstring(hsIProfile
01210         _Province), helpcontext(60)] HRESULT Province([out
```



```
, retval] BSTR *pVal);
01190      [propput, id(6), helpstring(hsIProfile
_Province), helpcontext(60)] HRESULT Province([in]
BSTR newVal);
01191      [propget, id(7), helpstring(hsIProfile
_City), helpcontext(60)] HRESULT City([out, retval
] BSTR *pVal);
01192      [propput, id(7), helpstring(hsIProfile
_City), helpcontext(60)] HRESULT City([in] BSTR ne
wVal);
01193      [propget, id(8), helpstring(hsIProfile
_PhoneHome), helpcontext(60)] HRESULT PhoneHome([o
ut, retval] BSTR *pVal);
01194      [propput, id(8), helpstring(hsIProfile
_PhoneHome), helpcontext(60)] HRESULT PhoneHome([i
n] BSTR newVal);
01195      [propget, id(9), helpstring(hsIProfile
_PhoneOffice), helpcontext(60)] HRESULT PhoneOffic
e([out, retval] BSTR *pVal);
01196      [propput, id(9), helpstring(hsIProfile
_PhoneOffice), helpcontext(60)] HRESULT PhoneOffic
e([in] BSTR newVal);
01197      [propget, id(10), helpstring(hsIProfil
e_PhoneMobile), helpcontext(60)] HRESULT PhoneMobi
le([out, retval] BSTR *pVal);
01198      [propput, id(10), helpstring(hsIProfil
e_PhoneMobile), helpcontext(60)] HRESULT PhoneMobi
le([in] BSTR newVal);
01199      [propget, id(11), helpstring(hsIProfil
e_Homepage), helpcontext(60)] HRESULT Homepage([ou
t, retval] BSTR *pVal);
01200      [propput, id(11), helpstring(hsIProfil
e_Homepage), helpcontext(60)] HRESULT Homepage([in
] BSTR newVal);
01201      [propget, id(12), helpstring(hsIProfil
e_About), helpcontext(60)] HRESULT About([out, ret
val] BSTR *pVal);
```

```
01202      [propput, id(12), helpstring(hsIProfil
e_About), helpcontext(60)] HRESULT About([in] BSTR
newVal);
01203      [propget, id(13), helpstring(hsIProfil
e_MoodText), helpcontext(60)] HRESULT MoodText([ou
t, retval] BSTR *pVal);
01204      [propput, id(13), helpstring(hsIProfil
e_MoodText), helpcontext(60)] HRESULT MoodText([in
] BSTR newVal);
01205      [propget, id(14), helpstring(hsIProfil
e_Timezone), helpcontext(60)] HRESULT Timezone([ou
t, retval] LONG *pVal);
01206      [propput, id(14), helpstring(hsIProfil
e_Timezone), helpcontext(60)] HRESULT Timezone([in
] LONG newVal);
01207      [propget, id(15), helpstring(hsIProfil
e_NATimeout), helpcontext(60)] HRESULT CallNoAnsw
erTimeout([out, retval] short *pVal);
01208      [propput, id(15), helpstring(hsIProfil
e_NATimeout), helpcontext(60)] HRESULT CallNoAnsw
erTimeout([in] short newVal);
01209      [propget, id(16), helpstring(hsIProfil
e_ApplyCF), helpcontext(60)] HRESULT CallApplyCF([
out, retval] VARIANT_BOOL *pVal);
01210      [propput, id(16), helpstring(hsIProfil
e_ApplyCF), helpcontext(60)] HRESULT CallApplyCF([
in] VARIANT_BOOL newVal);
01211      [propget, id(17), helpstring(hsIProfil
e_SendToVM), helpcontext(60)] HRESULT CallSendToVM
([out, retval] VARIANT_BOOL *pVal);
01212      [propput, id(17), helpstring(hsIProfil
e_SendToVM), helpcontext(60)] HRESULT CallSendToVM
([in] VARIANT_BOOL newVal);
01213      [propget, id(18), helpstring(hsIProfil
e_CFRules), helpcontext(60)] HRESULT CallForwardRu
les([out, retval] BSTR *pVal);
01214      [propput, id(18), helpstring(hsIProfil
```

```

e_CFRules), helpcontext(60)] HRESULT CallForwardRu
les([in] BSTR newVal);
01215      [propget, id(19), helpstring(hsIProfil
e_Balance), helpcontext(60)] HRESULT Balance([out,
retval] LONG *pVal);
01216      [propget, id(20), helpstring(hsIProfil
e_BalanceCurr), helpcontext(60)] HRESULT BalanceCu
rrency([out, retval] BSTR *pVal);
01217      [propget, id(21), helpstring(hsIProfil
e_BalanceText), helpcontext(60)] HRESULT BalanceTo
Text([out, retval] BSTR *pVal);
01218      [propget, id(22), helpstring(hsIProfil
e_IPCountry), helpcontext(60)] HRESULT IPCountry([
out, retval] BSTR *pVal);
01219      [propget, id(23), helpstring(hsIProfil
e_ValidSmsNums), helpcontext(60)] HRESULT Validate
dSmsNumbers([out, retval] BSTR* pVal);
01220      [propget, id(24), helpstring(hsIProfil
e_RichMoodText), helpcontext(60)] HRESULT RichMood
Text([out, retval] BSTR* pVal);
01221      [propput, id(24), helpstring(hsIProfil
e_RichMoodText), helpcontext(60)] HRESULT RichMood
Text([in] BSTR newVal);
01222      };
01223      [
01224          object,
01225          uuid(84513274-9C49-4AAA-B8FC-316EB32CF
E95),
01226          dual,
01227          nonextensible,
01228          pointer_default(unique),
01229          oleautomation,
01230          helpstring(hsIGroup),
01231          helpcontext(9)
01232      ]
01233      interface IGroup : IDispatch
01234      {

```

```
01235      [propget, id(1), helpstring(hsIGroup_I
d), helpcontext(9)] HRESULT Id([out, retval] LONG
*pVal);
01236      [propget, id(2), helpstring(hsIGroup_T
ype), helpcontext(9)] HRESULT Type([out, retval] T
GroupType *pVal);
01237      [propget, id(3), helpstring(hsIGroup_C
ustomId), helpcontext(9)] HRESULT CustomGroupId([o
ut, retval] BSTR *pVal);
01238      [propget, id(4), helpstring(hsIGroup_D
isplayName), helpcontext(9)] HRESULT DisplayName([
out, retval] BSTR *pVal);
01239      [propput, id(4), helpstring(hsIGroup_D
isplayName), helpcontext(9)] HRESULT DisplayName([
in] BSTR newVal);
01240      [propget, id(5), helpstring(hsIGroup_U
sers), helpcontext(9)] HRESULT Users([out, retval]
IUserCollection **pVal);
01241      [propget, id(6), helpstring(hsIGroup_O
nlineUsers), helpcontext(9)] HRESULT OnlineUsers([
out, retval] IUserCollection **pVal);
01242      [propget, id(7), helpstring(hsIGroup_I
sVisible), helpcontext(9)] HRESULT IsVisible([out,
retval] VARIANT_BOOL *pVal);
01243      [propget, id(8), helpstring(hsIGroup_I
sExpanded), helpcontext(9)] HRESULT IsExpanded([ou
t, retval] VARIANT_BOOL *pVal);
01244      [id(9), helpstring(hsIGroup_AddUser),
helpcontext(9)] HRESULT AddUser([in] BSTR Username
);
01245      [id(10), helpstring(hsIGroup_RemoveUse
r), helpcontext(9)] HRESULT RemoveUser([in] BSTR U
sername);
01246      [id(11), helpstring(hsIGroup_Share), h
elpcontext(9)] HRESULT Share([in, defaultvalue("")
] BSTR MessageText);
01247      [id(12), helpstring(hsIGroup_Accept),
```

```

helpcontext(9)] HRESULT Accept();
01248     [id(13), helpstring(hsIGroup_Decline)]
    HRESULT Decline();
01249     };
01250     [
01251         object,
01252         uuid(4C5C99DC-09CF-4A9C-BD94-8D655454A
1F4),
01253         dual,
01254         nonextensible,
01255         pointer_default(unique),
01256         oleautomation,
01257         helpstring(hsIGroupCollection),
01258         helpcontext(8)
01259     ]
01260     interface IGroupCollection : IDispatch
01261     {
01262         [propget, id(1), helpstring(hsIGroupCo
ll_Count),helpcontext(8)] HRESULT Count([out, retv
al] LONG* pVal);
01263         [id(2), helpstring(hsIGroupColl_Add),h
elpcontext(8)] HRESULT Add([in] IGroup* pGroup);
01264         [id(3), helpstring(hsIGroupColl_Remove
),helpcontext(8)] HRESULT Remove([in] long Index);
01265         [id(4), helpstring(hsIGroupColl_Remove
All),helpcontext(8)] HRESULT RemoveAll(void);
01266         [propget, id(DISPID_VALUE), helpstring
(hsIGroupColl_Item),helpcontext(8)] HRESULT Item([
in] LONG Index, [out, retval] IGroup ** pVal);
01267         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIGroupColl_NewEnum),helpcontext(8)] HRESULT _
NewEnum([out, retval] IUnknown** pVal);
01268     };
01269     interface ISmsMessage;
01270     [
01271         object,
01272         uuid(A9062508-C926-4415-ABB7-A5A46DB34

```

```

456),
01273     dual,
01274     nonextensible,
01275     pointer_default(unique),
01276     oleautomation,
01277     helpstring(hsISmsChunk),
01278     helpcontext(72)
01279 ]
01280 interface ISmsChunk : IDispatch
01281 {
01282     [propget, id(1), helpstring(hsISmsChun
k_Id), helpcontext(72)] HRESULT Id([out, retval] L
ONG *pVal);
01283     [propget, id(2), helpstring(hsISmsChun
k_CharsLeft), helpcontext(72)] HRESULT CharactersL
eft([out, retval] LONG *pVal);
01284     [propget, id(3), helpstring(hsISmsChun
k_Text), helpcontext(72)] HRESULT Text([out, retva
l] BSTR *pVal);
01285     [propget, id(4), helpstring(hsISmsChun
k_Message), helpcontext(72)] HRESULT Message([out,
retval] ISmsMessage** pVal);
01286 };
01287 [
01288     object,
01289     uuid(BBDA7D2D-B1C2-4AF9-AB5B-D762341D8
903),
01290     dual,
01291     nonextensible,
01292     pointer_default(unique),
01293     oleautomation,
01294     helpstring(hsISmsChunkCollection),
01295     helpcontext(73)
01296 ]
01297 interface ISmsChunkCollection : IDispatch
01298 {
01299     [propget, id(1), helpstring(hsISmsChun

```

```

kColl_Count),helpcontext(73)] HRESULT Count([out,
retval] LONG* pVal);
01300      [id(2), helpstring(hsISmsChunkColl_Add
),helpcontext(73)] HRESULT Add([in] ISmsChunk* pIt
em);
01301      [id(3), helpstring(hsISmsChunkColl_Rem
ove),helpcontext(73)] HRESULT Remove([in] long Ind
ex);
01302      [id(4), helpstring(hsISmsChunkColl_Rem
oveAll),helpcontext(73)] HRESULT RemoveAll(void);
01303      [propget, id(DISPID_VALUE), helpstring
(hsISmsChunkColl_Item),helpcontext(73)] HRESULT It
em([in] LONG Index, [out, retval] ISmsChunk ** pVa
l);
01304      [propget, id(DISPID_NEWENUM), helpstri
ng(hsISmsChunkColl_NewEnum),helpcontext(73)] HRESU
LT _NewEnum([out, retval] IUnknown** pVal);
01305      };
01306      [
01307          object,
01308          uuid(A9B9C33A-49A6-41D5-B13F-2AA4F284F
406),
01309          dual,
01310          nonextensible,
01311          pointer_default(unique),
01312          oleautomation,
01313          helpstring(hsISmsTarget),
01314          helpcontext(74)
01315      ]
01316      interface ISmsTarget : IDispatch
01317      {
01318          [propget, id(1), helpstring(hsISmsTarg
et_Status), helpcontext(74)] HRESULT Status([out,
retval] TSmsTargetStatus *pVal);
01319          [propget, id(2), helpstring(hsISmsTarg
et_Number), helpcontext(74)] HRESULT Number([out,
retval] BSTR *pVal);

```

```

01320         [propget, id(3), helpstring(hsISmsTarget_Message), helpcontext(74)] HRESULT Message([out, retval] ISmsMessage** pVal);
01321     };
01322     [
01323         object,
01324         uuid(A2365EF3-4805-4DD3-A9D4-1A8AE3E17D84),
01325         dual,
01326         nonextensible,
01327         pointer_default(unique),
01328         oleautomation,
01329         helpstring(hsISmsTargetCollection),
01330         helpcontext(75)
01331     ]
01332     interface ISmsTargetCollection : IDispatch
01333     {
01334         [propget, id(1), helpstring(hsISmsTargetColl_Count), helpcontext(75)] HRESULT Count([out, retval] LONG* pVal);
01335         [id(2), helpstring(hsISmsTargetColl_Add), helpcontext(75)] HRESULT Add([in] ISmsTarget* pItem);
01336         [id(3), helpstring(hsISmsTargetColl_Remove), helpcontext(75)] HRESULT Remove([in] long Index);
01337         [id(4), helpstring(hsISmsTargetColl_RemoveAll), helpcontext(75)] HRESULT RemoveAll(void);
01338         [propget, id(DISPID_VALUE), helpstring(hsISmsTargetColl_Item), helpcontext(75)] HRESULT Item([in] LONG Index, [out, retval] ISmsTarget** pVal);
01339         [propget, id(DISPID_NEWENUM), helpstring(hsISmsTargetColl_NewEnum), helpcontext(75)] HRESULT _NewEnum([out, retval] IUnknown** pVal);
01340     };

```



```

01341     [
01342         object,
01343         uuid(82D97F2A-0E17-40F3-8E01-24937F936
FF4),
01344         dual,
01345         nonextensible,
01346         pointer_default(unique),
01347         oleautomation,
01348         helpstring(hsISmsMessage),
01349         helpcontext(76)
01350     ]
01351     interface ISmsMessage : IDispatch
01352     {
01353         [propget, id(1), helpstring(hsISmsMess
age_Id), helpcontext(76)] HRESULT Id([out, retval]
LONG *pVal);
01354         [propget, id(2), helpstring(hsISmsMess
age_Type), helpcontext(76)] HRESULT Type([out, ret
val] TSmsMessageType *pVal);
01355         [propget, id(3), helpstring(hsISmsMess
age_Status), helpcontext(76)] HRESULT Status([out,
retval] TSmsMessageStatus *pVal);
01356         [propget, id(4), helpstring(hsISmsMess
age_FailureReason), helpcontext(76)] HRESULT Failu
reReason([out, retval] TSmsFailureReason *pVal);
01357         [propget, id(5), helpstring(hsISmsMess
age_FailedUnseen), helpcontext(76)] HRESULT IsFail
edUnseen([out, retval] VARIANT_BOOL *pVal);
01358         [propput, id(6), helpstring(hsISmsMess
age_Seen), helpcontext(76)] HRESULT Seen([in] VARI
ANT_BOOL newVal);
01359         [propget, id(7), helpstring(hsISmsMess
age_Price), helpcontext(76)] HRESULT Price([out, r
etval] LONG *pVal);
01360         [propget, id(8), helpstring(hsISmsMess
age_PricePrec), helpcontext(76)] HRESULT PricePrec
ision([out, retval] LONG *pVal);

```

```
01361      [propget, id(9), helpstring(hsISmsMessage_PriceCurr), helpcontext(76)] HRESULT PriceCurrency([out, retval] BSTR *pVal);
01362      [propget, id(10), helpstring(hsISmsMessage_ReplyTo), helpcontext(76)] HRESULT ReplyToNumber([out, retval] BSTR *pVal);
01363      [propput, id(10), helpstring(hsISmsMessage_ReplyTo), helpcontext(76)] HRESULT ReplyToNumber([in] BSTR newVal);
01364      [propget, id(11), helpstring(hsISmsMessage_Targets), helpcontext(76)] HRESULT Targets([out, retval] ISmsTargetCollection* *pVal);
01365      [propget, id(12), helpstring(hsISmsMessage_Body), helpcontext(76)] HRESULT Body([out, retval] BSTR *pVal);
01366      [propput, id(12), helpstring(hsISmsMessage_Body), helpcontext(76)] HRESULT Body([in] BSTR newVal);
01367      [propget, id(13), helpstring(hsISmsMessage_Chunks), helpcontext(76)] HRESULT Chunks([out, retval] ISmsChunkCollection* *pVal);
01368      [propget, id(14), helpstring(hsISmsMessage_Timestamp), helpcontext(76)] HRESULT Timestamp([out, retval] DATE* pVal);
01369      [id(15), helpstring(hsISmsMessage_Send), helpcontext(76)] HRESULT Send();
01370      [id(16), helpstring(hsISmsMessage_Delete), helpcontext(76)] HRESULT Delete();
01371      [propget, id(17), helpstring(hsISmsMessage_TargetNumbers), helpcontext(76)] HRESULT TargetNumbers([out, retval] BSTR* pVal);
01372      [propput, id(17), helpstring(hsISmsMessage_TargetNumbers), helpcontext(76)] HRESULT TargetNumbers([in] BSTR newVal);
01373      };
01374      [
01375      object,
```

```

01376         uuid(1D5BAB2E-69CC-4136-8E02-FC20767DC
01377         9E2),
01378         dual,
01379         nonextensible,
01380         pointer_default(unique),
01381         oleautomation,
01382         helpstring(hsISmsMessageCollection),
01383         helpcontext(77)
01384     ]
01385     interface ISmsMessageCollection : IDispatch
01386     {
01387         [propget, id(1), helpstring(hsISmsMessageColl_Count),helpcontext(77)] HRESULT Count([out, retval] LONG* pVal);
01388         [id(2), helpstring(hsISmsMessageColl_Add),helpcontext(77)] HRESULT Add([in] ISmsMessage* pItem);
01389         [id(3), helpstring(hsISmsMessageColl_Remove),helpcontext(77)] HRESULT Remove([in] long Index);
01390         [id(4), helpstring(hsISmsMessageColl_RemoveAll),helpcontext(77)] HRESULT RemoveAll(void);
01391         [propget, id(DISPID_VALUE), helpstring(hsISmsMessageColl_Item),helpcontext(77)] HRESULT Item([in] LONG Index, [out, retval] ISmsMessage** pVal);
01392         [propget, id(DISPID_NEWENUM), helpstring(hsISmsMessageColl_NewEnum),helpcontext(77)] HRESULT _NewEnum([out, retval] IUnknown** pVal);
01393     };
01394     [
01395         object,
01396         uuid(2BC8C53B-3312-4A77-AC87-31DF18A1E
01397         C51),

```

```

01397         dual,
01398         nonextensible,
01399         pointer_default(unique),
01400         oleautomation,
01401         helpstring(hsISettings),
01402         helpcontext(2)
01403     ]
01404     interface ISettings : IDispatch
01405     {
01406         [propget, id(1), helpstring(hsISetting
s_AudioIn),helpcontext(2)] HRESULT AudioIn([out, r
etval] BSTR *pVal);
01407         [propput, id(1), helpstring(hsISetting
s_AudioIn),helpcontext(2)] HRESULT AudioIn([in] BS
TR newVal);
01408         [propget, id(2), helpstring(hsISetting
s_AudioOut),helpcontext(2)] HRESULT AudioOut([out,
retval] BSTR *pVal);
01409         [propput, id(2), helpstring(hsISetting
s_AudioOut),helpcontext(2)] HRESULT AudioOut([in]
BSTR newVal);
01410         [propget, id(3), helpstring(hsISetting
s_AutoAway),helpcontext(2)] HRESULT AutoAway([out,
retval] VARIANT_BOOL *pVal);
01411         [propput, id(3), helpstring(hsISetting
s_AutoAway),helpcontext(2)] HRESULT AutoAway([in]
VARIANT_BOOL newVal);
01412         [propget, id(4), helpstring(hsISetting
s_Ringer),helpcontext(2)] HRESULT Ringer([out, ret
val] BSTR *pVal);
01413         [propput, id(4), helpstring(hsISetting
s_Ringer),helpcontext(2)] HRESULT Ringer([in] BSTR
newVal);
01414         [propput, id(5), helpstring(hsISetting
s_Avatar),helpcontext(2)] HRESULT Avatar([in, defa
ultvalue("1")] BSTR Id, [in] BSTR newVal);
01415         [propget, id(6), helpstring(hsISetting

```

```
s_RingToneStatus),helpcontext(2)] HRESULT RingTone
Status([in, defaultvalue("1")] BSTR Id,[out, retva
l] VARIANT_BOOL *pVal);
01416      [propput, id(7), helpstring(hsISetting
s_RingTone),helpcontext(2)] HRESULT RingTone([in,
defaultvalue("1")] BSTR Id, [in] BSTR newVal);
01417      [propget, id(8), helpstring(hsISetting
s_VideoIn),helpcontext(2)] HRESULT VideoIn([out, r
etval] BSTR *pVal);
01418      [propput, id(8), helpstring(hsISetting
s_VideoIn),helpcontext(2)] HRESULT VideoIn([in] BS
TR newVal);
01419      [propget, id(9), helpstring(hsISetting
s_PCSpeaker),helpcontext(2)] HRESULT PCSpeaker([ou
t, retval] VARIANT_BOOL *pVal);
01420      [propput, id(9), helpstring(hsISetting
s_PCSpeaker),helpcontext(2)] HRESULT PCSpeaker([in
] VARIANT_BOOL newVal);
01421      [propget, id(10), helpstring(hsISettin
gs_AGC),helpcontext(2)] HRESULT AGC([out, retval]
VARIANT_BOOL *pVal);
01422      [propput, id(10), helpstring(hsISettin
gs_AGC),helpcontext(2)] HRESULT AGC([in] VARIANT_B
OOL newVal);
01423      [propget, id(11), helpstring(hsISettin
gs_AEC),helpcontext(2)] HRESULT AEC([out, retval]
VARIANT_BOOL *pVal);
01424      [propput, id(11), helpstring(hsISettin
gs_AEC),helpcontext(2)] HRESULT AEC([in] VARIANT_B
OOL newVal);
01425      [propget, id(12), helpstring(hsISettin
gs_Language),helpcontext(2)] HRESULT Language([out
, retval] BSTR *pVal);
01426      [propput, id(12), helpstring(hsISettin
gs_Language),helpcontext(2)] HRESULT Language([in]
BSTR newVal);
01427      [propput, id(6), helpstring(hsISetting
```

```

s_RingToneStatus),helpcontext(2)] HRESULT RingTone
Status([in, defaultvalue("1")] BSTR Id,[in] VARIAN
T_BOOL newVal);
01428      [id(13), helpstring(hsISettings_SaveAv
atar), helpcontext(2)] HRESULT SaveAvatarToFile([i
n] BSTR Filename, [in, defaultvalue("1")] BSTR Ava
tarId);
01429      [id(14), helpstring(hsISettings_LoadAv
atar), helpcontext(2)] HRESULT LoadAvatarFromFile(
[in] BSTR Filename, [in, defaultvalue("1")] BSTR A
vatarId);
01430      [id(15), helpstring(hsISettings_IdleTi
mer),helpcontext(2)] HRESULT ResetIdleTimer(void);
01431      };
01432      [
01433          object,
01434          uuid(C4BDF667-3FF7-4B44-A9F4-F3937E3E6
D86),
01435          dual,
01436          nonextensible,
01437          pointer_default(unique),
01438          oleautomation,
01439          helpstring(hsIPluginMenuItem),
01440          helpcontext(87)
01441      ]
01442      interface IPluginMenuItem : IDispatch
01443      {
01444          [propget, id(1), helpstring(hsIPluginM
enuItem_Id), helpcontext(87)] HRESULT Id([out, ret
val] BSTR* pVal);
01445          [propput, id(2), helpstring(hsIPluginM
enuItem_Caption), helpcontext(87)] HRESULT Caption
([in] BSTR newVal);
01446          [propput, id(3), helpstring(hsIPluginM
enuItem_Hint), helpcontext(87)] HRESULT Hint([in]
BSTR newVal);
01447          [propput, id(4), helpstring(hsIPluginM

```

```

enuItem_Enabled), helpcontext(87)] HRESULT Enabled
([in] VARIANT_BOOL newVal);
01448     [id(5), helpstring(hsIPluginMenuItem_De
lete), helpcontext(87)] HRESULT Delete(void);
01449     };
01450     [
01451         object,
01452         uuid(4DF5F83A-0ABA-417D-A6FC-62A68AE06
EF7),
01453         dual,
01454         nonextensible,
01455         pointer_default(unique),
01456         oleautomation,
01457         helpstring(hsIPluginEvent),
01458         helpcontext(88)
01459     ]
01460     interface IPluginEvent : IDispatch
01461     {
01462         [propget, id(1), helpstring(hsIPluginE
vent_Id), helpcontext(88)] HRESULT Id([out, retval]
BSTR* pVal);
01463         [id(2), helpstring(hsIPluginEvent_Dele
te), helpcontext(88)] HRESULT Delete(void);
01464     };
01465     [
01466         object,
01467         uuid(4D36B368-B88C-45fa-B136-3EF77B2EA
D39),
01468         dual,
01469         pointer_default(unique),
01470         helpstring(hsIFileTransfer),
01471         helpcontext(89)
01472     ]
01473     interface IFileTransfer : IDispatch
01474     {
01475         [propget, id(1), helpstring(hsIFileTra
nsfer_Id), helpcontext(89)] HRESULT Id([out, retva

```

```
1] BSTR* pVal);
01476         [propget, id(2), helpstring(hsIFileTra
nsfer_Type), helpcontext(89)] HRESULT Type([out, r
etval] TFileTransferType* pVal);
01477         [propget, id(3), helpstring(hsIFileTra
nsfer_Status), helpcontext(89)] HRESULT Status([ou
t, retval] TFileTransferStatus* pVal);
01478         [propget, id(4), helpstring(hsIFileTra
nsfer_FailReason), helpcontext(89)] HRESULT Failur
eReason([out, retval] TFileTransferFailureReason*
pVal);
01479         [propget, id(5), helpstring(hsIFileTra
nsfer_PartnerHandle), helpcontext(89)] HRESULT Par
tnerHandle([out, retval] BSTR* pVal);
01480         [propget, id(6), helpstring(hsIFileTra
nsfer_PartnerDispName), helpcontext(89)] HRESULT P
artnerDisplayName([out, retval] BSTR* pVal);
01481         [propget, id(7), helpstring(hsIFileTra
nsfer_StartTime), helpcontext(89)] HRESULT StartTi
me([out, retval] DATE* pVal);
01482         [propget, id(8), helpstring(hsIFileTra
nsfer_FinishTime), helpcontext(89)] HRESULT Finish
Time([out, retval] DATE* pVal);
01483         [propget, id(9), helpstring(hsIFileTra
nsfer_FilePath), helpcontext(89)] HRESULT FilePath
([out, retval] BSTR* pVal);
01484         [propget, id(10), helpstring(hsIFileTr
ansfer_FileName), helpcontext(89)] HRESULT FileNam
e([out, retval] BSTR* pVal);
01485         [propget, id(11), helpstring(hsIFileTr
ansfer_ByteSec), helpcontext(89)] HRESULT BytesPer
Second([out, retval] BSTR* pVal);
01486         [propget, id(12), helpstring(hsIFileTr
ansfer_ByteTransferred), helpcontext(89)] HRESULT
BytesTransferred([out, retval] BSTR* pVal);
01487     };
01488     [
```



```

01489         object,
01490         uuid(70A59A25-E823-4c3f-8F33-775008895
EE3),
01491         dual,
01492         pointer_default(unique),
01493         helpstring(hsIFileTransferColl),
01494         helpcontext(90)
01495     ]
01496     interface IFileTransferCollection : IDisp
atch
01497     {
01498         [propget, id(1), helpstring(hsIFileTra
nsferColl_Count),helpcontext(90)] HRESULT Count([o
ut, retval] LONG* pVal);
01499         [id(2), helpstring(hsIFileTransferColl
_Add),helpcontext(90)] HRESULT Add([in] IFileTrans
fer* pItem);
01500         [id(3), helpstring(hsIFileTransferColl
_Remove),helpcontext(90)] HRESULT Remove([in] long
Index);
01501         [id(4), helpstring(hsIFileTransferColl
_RemoveAll),helpcontext(90)] HRESULT RemoveAll(void
);
01502         [propget, id(DISPID_VALUE), helpstring
(hsIFileTransferColl_Item),helpcontext(90)] HRESUL
T Item([in] LONG Index, [out, retval] IFileTransfer
** pVal);
01503         [propget, id(DISPID_NEWENUM), helpstri
ng(hsIFileTransferColl_NewEnum),helpcontext(90)] H
RESULT _NewEnum([out, retval] IUnknown** pVal);
01504     };
01505     [
01506         object,
01507         uuid(838731B0-88E7-4BED-81DC-B35CA8433
341),
01508         dual,
01509         nonextensible,

```

```

01510         pointer_default(unique),
01511         oleautomation,
01512         helpstring(hsIClient),
01513         helpcontext(36)
01514     ]
01515     interface IClient : IDispatch
01516     {
01517         [id(1), helpstring(hsIClient_Start), helpcontext(36)] HRESULT Start([in, defaultvalue(0)] VARIANT_BOOL Minimized, [in, defaultvalue(0)] VARIANT_BOOL Nosplash);
01518         [id(2), helpstring(hsIClient_Minimize), helpcontext(36)] HRESULT Minimize();
01519         [id(3), helpstring(hsIClient_Shutdown), helpcontext(36)] HRESULT Shutdown();
01520         [propget, id(4), helpstring(hsIClient_Running), helpcontext(36)] HRESULT IsRunning([out, retval] VARIANT_BOOL *pVal);
01521         [id(5), helpstring(hsIClient_OpenProfile), helpcontext(36)] HRESULT OpenProfileDialog();
01522         [id(6), helpstring(hsIClient_OpenUserInfo), helpcontext(36)] HRESULT OpenUserInfoDialog([in] BSTR Username);
01523         [id(7), helpstring(hsIClient_OpenConference), helpcontext(36)] HRESULT OpenConferenceDialog();
01524         [id(8), helpstring(hsIClient_OpenSearch), helpcontext(36)] HRESULT OpenSearchDialog();
01525         [id(9), helpstring(hsIClient_OpenOptions), helpcontext(36)] HRESULT OpenOptionsDialog([in] BSTR Page);
01526         [id(10), helpstring(hsIClient_OpenCallHist), helpcontext(36)] HRESULT OpenCallHistoryTab();
01527         [id(11), helpstring(hsIClient_OpenContacts), helpcontext(36)] HRESULT OpenContactsTab();
01528         [id(12), helpstring(hsIClient_OpenDial

```

```
pad),helpcontext(36)] HRESULT OpenDialpadTab();
01529      [id(13), helpstring(hsIClient_OpenSend
Cont),helpcontext(36)] HRESULT OpenSendContactsDia
log([in, defaultvalue("")] BSTR Username);
01530      [id(14), helpstring(hsIClient_OpenBloc
kUser),helpcontext(36)] HRESULT OpenBlockedUsersDi
alog();
01531      [id(15), helpstring(hsIClient_OpenImpo
rtContact),helpcontext(36)] HRESULT OpenImportCont
actsWizard();
01532      [id(16), helpstring(hsIClient_OpenGets
tart),helpcontext(36)] HRESULT OpenGettingStartedW
izard();
01533      [id(17), helpstring(hsIClient_OpenAuth
oriz),helpcontext(36)] HRESULT OpenAuthorizationDi
alog([in] BSTR Username);
01534      [id(18), helpstring(hsIClient_OpenDial
og),helpcontext(36)] HRESULT OpenDialog([in] BSTR
Name, [in, defaultvalue("")] BSTR Param1, [in, def
aultvalue("")] BSTR Param2);
01535      [id(19), helpstring(hsIClient_OpenVide
oTest),helpcontext(36)] HRESULT OpenVideoTestDialo
g();
01536      [id(20), helpstring(hsIClient_OpenAddC
ontact),helpcontext(36)] HRESULT OpenAddContactDia
log([in, defaultvalue("")] BSTR Username);
01537      [id(21), helpstring(hsIClient_OpenMess
age),helpcontext(36)] HRESULT OpenMessageDialog([i
n] BSTR Username, [in, defaultvalue("")] BSTR Text
);
01538      [id(22), helpstring(hsIClient_OpenFile
Trans),helpcontext(36)] HRESULT OpenFileTransferDi
alog([in] BSTR User, [in] BSTR Folder);
01539      [id(23), helpstring(hsIClient_Focus),h
elpcontext(36)] HRESULT Focus();
01540      [id(24), helpstring(hsIClient_BtnPress
ed),helpcontext(36)] HRESULT ButtonPressed([in] BS
```

```

TR Key);
01541      [id(25), helpstring(hsIClient_BtnReleased),helpcontext(36)] HRESULT ButtonReleased([in]
BSTR Key);
01542      [id(26), helpstring(hsIClient_OpenSms)
] HRESULT OpenSmsDialog([in] BSTR SmsId);
01543      [id(27), helpstring(hsIClient_CreateEvent), helpcontext(36)] HRESULT CreateEvent([in] BS
TR EventId, [in] BSTR Caption, [in] BSTR Hint, [ou
t,retval] IPluginEvent** pEvent);
01544      [id(28), helpstring(hsIClient_CreateMe
nuItem), helpcontext(36)] HRESULT CreateMenuItem([
in] BSTR MenuItemId, [in] TPluginContext PluginCon
text, [in] BSTR CaptionText, [in,defaultvalue("")]
BSTR HintText, [in, defaultvalue("")] BSTR IconPa
th, [in, defaultvalue(-1)] VARIANT_BOOL Enabled, [
in, defaultvalue(pluginContactTypeAll)] TPluginCon
tactType ContactType, [in, defaultvalue(0)] VARIAN
T_BOOL MultipleContacts, [out,retval] IPluginMenuI
tem** pMenuItem);
01545      [propget, id(29), helpstring(hsIClient
_Wallpaper), helpcontext(36)] HRESULT Wallpaper([o
ut, retval] BSTR* pVal);
01546      [propput, id(29), helpstring(hsIClient
_Wallpaper), helpcontext(36)] HRESULT Wallpaper([i
n] BSTR newVal);
01547      [id(30), helpstring(hsIClient_OpenLive
),helpcontext(36)] HRESULT OpenLiveTab();
01548      };
01549      [
01550      object,
01551      uuid(B1878BFE-53D3-402E-8C86-190B19AF7
0D5),
01552      dual,
01553      nonextensible,
01554      pointer_default(unique),
01555      oleautomation,

```

```

01556         helpstring(hsISkype),
01557         helpcontext(13)
01558     ]
01559     interface ISkype : IDispatch
01560     {
01561         [propget, id(1), helpstring(hsISkype_T
imeout), helpcontext(13)] HRESULT Timeout([out, re
tval] LONG *pVal);
01562         [propput, id(1), helpstring(hsISkype_T
imeout), helpcontext(13)] HRESULT Timeout([in] LON
G newVal);
01563         [propget, id(2), helpstring(hsISkype_P
roperty), helpcontext(13)] HRESULT Property([in] B
STR ObjectType, [in] BSTR ObjectId, [in] BSTR Prop
Name, [out, retval] BSTR *pVal);
01564         [propput, id(2), helpstring(hsISkype_P
roperty), helpcontext(13)] HRESULT Property([in] B
STR ObjectType, [in] BSTR ObjectId, [in] BSTR Prop
Name, [in] BSTR newVal);
01565         [propget, id(3), helpstring(hsISkype_V
ariable), helpcontext(13)] HRESULT Variable([in] B
STR Name, [out, retval] BSTR *pVal);
01566         [propput, id(3), helpstring(hsISkype_V
ariable), helpcontext(13)] HRESULT Variable([in] B
STR Name, [in] BSTR newVal);
01567         [propget, id(4), helpstring(hsISkype_C
urrentUserHandle), helpcontext(13)] HRESULT Curren
tUserHandle([out, retval] BSTR *pVal);
01568         [propget, id(5), helpstring(hsISkype_C
urrentUserStatus), helpcontext(13)] HRESULT Curren
tUserStatus([out, retval] TUserStatus *pVal);
01569         [propput, id(5), helpstring(hsISkype_C
urrentUserStatus), helpcontext(13)] HRESULT Curren
tUserStatus([in] TUserStatus newVal);
01570         [propget, id(6), helpstring(hsISkype_C
onnectionStatus), helpcontext(13)] HRESULT Connect
ionStatus([out, retval] TConnectionStatus *pVal);

```

```
01571      [propget, id(7), helpstring(hsISkype_M
ute), helpcontext(13)] HRESULT Mute([out, retval]
VARIANT_BOOL *pVal);
01572      [propget, id(7), helpstring(hsISkype_M
ute), helpcontext(13)] HRESULT Mute([in] VARIANT_B
OOL newVal);
01573      [propget, id(8), helpstring(hsISkype_V
ersion), helpcontext(13)] HRESULT Version([out, re
tval] BSTR *pVal);
01574      [propget, id(9), helpstring(hsISkype_P
rivilege), helpcontext(13)] HRESULT Privilege([in]
BSTR Name, [out, retval] VARIANT_BOOL *pVal);

01575      [propget, id(10), helpstring(hsISkype_
CurrentUser), helpcontext(13)] HRESULT CurrentUser
([out, retval] IUser **pVal);
01576      [propget, id(11), helpstring(hsISkype_
Convert), helpcontext(13)] HRESULT Convert([out, r
etval] IConversion **pVal);
01577      [propget, id(12), helpstring(hsISkype_
Friends), helpcontext(13)] HRESULT Friends([out, r
etval] IUserCollection **pVal);
01578      [id(13), helpstring(hsISkype_Users), h
elpcontext(13)] HRESULT SearchForUsers([in] BSTR T
arget, [out, retval] IUserCollection **pVal);
01579      [propget, id(14), helpstring(hsISkype_
Calls), helpcontext(13)] HRESULT Calls([in, defaul
tvalue("")] BSTR Target, [out, retval] ICallCollec
tion **pVal);
01580      [propget, id(15), helpstring(hsISkype_
ActiveCalls), helpcontext(13)] HRESULT ActiveCalls
([out, retval] ICallCollection **pVal);
01581      [propget, id(16), helpstring(hsISkype_
MissedCalls), helpcontext(13)] HRESULT MissedCalls
([out, retval] ICallCollection **pVal);
01582      [propget, id(17), helpstring(hsISkype_
Messages), helpcontext(13)] HRESULT Messages([in,
```

```

defaultvalue("")) BSTR Target, [out, retval] IChat
MessageCollection **pVal);
01583      [propget, id(18), helpstring(hsISkype_
MissedMessages), helpcontext(13)] HRESULT MissedMe
ssages([out, retval] IChatMessageCollection **pVal
);
01584      [propget, id(19), helpstring(hsISkype_
AttachmentStatus), helpcontext(13)] HRESULT Attach
mentStatus([out, retval] TAttachmentStatus *pVal);
01585      [propget, id(20), helpstring(hsISkype_
Protocol), helpcontext(13)] HRESULT Protocol([out,
retval] LONG *pVal);
01586      [propput, id(20), helpstring(hsISkype_
Protocol), helpcontext(13)] HRESULT Protocol([in]
LONG newVal);
01587      [id(21), helpstring(hsISkype_Attach),
helpcontext(13)] HRESULT Attach([in, defaultvalue(
5)] LONG Protocol, [in, defaultvalue(-1)] VARIANT_
BOOL Wait);
01588      [id(22), helpstring(hsISkype_PlaceCall
), helpcontext(13)] HRESULT PlaceCall([in] BSTR Ta
rget, [in, defaultvalue("")] BSTR Target2, [in, de
faultvalue("")] BSTR Target3, [in, defaultvalue(""]
)] BSTR Target4, [out, retval] ICall **pCall);
01589      [id(23), helpstring(hsISkype_SendMessa
ge), helpcontext(13)] HRESULT SendMessage([in] BST
R Username, [in] BSTR Text, [out, retval] IChatMes
sage **pMessage);
01590      [propget, id(24), helpstring(hsISkype_
User), helpcontext(13)] HRESULT User([in, defaultv
alue("")] BSTR Username, [out, retval] IUser **pVa
l);
01591      [propget, id(25), helpstring(hsISkype_
Message), helpcontext(13)] HRESULT Message([in, de
faultvalue(0)] LONG Id, [out, retval] IChatMessage
**pVal);
01592      [propget, id(26), helpstring(hsISkype_

```

```
Call), helpcontext(13)] HRESULT Call([in, defaultvalue(0)] LONG Id, [out, retval] ICall **pVal);
01593      [id(27), helpstring(hsISkype_SendCommand), helpcontext(13)] HRESULT SendCommand([in] ICommand* pCommand);
01594      [propget, id(28), helpstring(hsISkype_Chats), helpcontext(13)] HRESULT Chats([out, retval] IChatCollection **pVal);
01595      [propget, id(29), helpstring(hsISkype_Chat), helpcontext(13)] HRESULT Chat([in, defaultvalue("")] BSTR Name, [out, retval] IChat **pVal);
01596      [id(30), helpstring(hsISkype_ChangeUserStatus), helpcontext(13)] HRESULT ChangeUserStatus([in] TUserStatus newVal);
01597      [propget, id(31), helpstring(hsISkype_Conference), helpcontext(13)] HRESULT Conference([in, defaultvalue(0)] LONG Id, [out, retval] IConference **pVal);
01598      [propget, id(32), helpstring(hsISkype_Conferences), helpcontext(13)] HRESULT Conferences([out, retval] IConferenceCollection **pVal);

01599      [propget, id(33), helpstring(hsISkype_Profile), helpcontext(13)] HRESULT Profile([in] BSTR Property, [out, retval] BSTR *pVal);
01600      [propput, id(33), helpstring(hsISkype_Profile), helpcontext(13)] HRESULT Profile([in] BSTR Property, [in] BSTR newVal);
01601      [propget, id(34), helpstring(hsISkype_ActiveChats), helpcontext(13)] HRESULT ActiveChats([out, retval] IChatCollection **pVal);
01602      [propget, id(35), helpstring(hsISkype_MissedChats), helpcontext(13)] HRESULT MissedChats([out, retval] IChatCollection **pVal);
01603      [propget, id(36), helpstring(hsISkype_RecentChats), helpcontext(13)] HRESULT RecentChats([out, retval] IChatCollection **pVal);
```



```
01604      [propget, id(37), helpstring(hsISkype_
BookedChats), helpcontext(13)] HRESULT BookmarkedC
hats([out, retval] IChatCollection **pVal);
01605      [id(38), helpstring(hsISkype_CreateCha
t), helpcontext(13)] HRESULT CreateChatWith([in] B
STR Username, [out, retval] IChat **pVal);
01606      [id(39), helpstring(hsISkype_CreateCha
tMulti), helpcontext(13)] HRESULT CreateChatMultip
le([in] IUserCollection *pMembers, [out, retval] I
Chat **pChat);
01607      [propget, id(40), helpstring(hsISkype_
Voicemails), helpcontext(13)] HRESULT Voicemails([
out, retval] IVoicemailCollection **pVal);
01608      [id(41), helpstring(hsISkype_SendVoice
mail), helpcontext(13)] HRESULT SendVoicemail([in]
  BSTR Username, [out, retval] IVoicemail **pVoicem
ail);
01609      [propget, id(42), helpstring(hsISkype_
UsersWaitAuth), helpcontext(13)] HRESULT UsersWait
ingAuthorization([out, retval] IUserCollection **p
Val);
01610      [id(43), helpstring(hsISkype_ClearChat
s), helpcontext(13)] HRESULT ClearChatHistory();
01611      [id(44), helpstring(hsISkype_ClearVMai
ls), helpcontext(13)] HRESULT ClearVoicemailHistor
y();
01612      [id(45), helpstring(hsISkype_ClearCall
s), helpcontext(13)] HRESULT ClearCallHistory([in,
  defaultvalue("ALL")] BSTR Username, [in, defaultv
alue(chsAllCalls)] TCallHistory type);
01613      [propget, id(46), helpstring(hsISkype_
CommandId), helpcontext(13)] HRESULT CommandId([ou
t, retval] VARIANT_BOOL *pVal);
01614      [propput, id(46), helpstring(hsISkype_
CommandId), helpcontext(13)] HRESULT CommandId([in
] VARIANT_BOOL newVal);
01615      [propget, id(47), helpstring(hsISkype_
```

```
Application), helpcontext(13)] HRESULT Application
([in] BSTR Name, [out, retval] IApplication **pVal
);
01616 [propget, id(48), helpstring(hsISkype_
Greeting), helpcontext(13)] HRESULT Greeting([in,
defaultvalue("")] BSTR Username, [out, retval] IVo
icemail **pVal);
01617 [propget, id(49), helpstring(hsISkype_
Cache), helpcontext(13)] HRESULT Cache([out, retva
l] VARIANT_BOOL *pVal);
01618 [propput, id(49), helpstring(hsISkype_
Cache), helpcontext(13)] HRESULT Cache([in] VARIAN
T_BOOL newVal);
01619 [id(50), helpstring(hsISkype_ResetCach
e), helpcontext(13)] HRESULT ResetCache();
01620 [propget, id(51), helpstring(hsISkype_
CurrUserProf), helpcontext(13)] HRESULT CurrentUse
rProfile([out, retval] IProfile **pVal);
01621 [propget, id(52), helpstring(hsISkype_
Groups), helpcontext(13)] HRESULT Groups([out, ret
val] IGroupCollection* *pVal);
01622 [propget, id(53), helpstring(hsISkype_
CustGroups), helpcontext(13)] HRESULT CustomGroups
([out, retval] IGroupCollection* *pVal);
01623 [propget, id(54), helpstring(hsISkype_
HardGroups), helpcontext(13)] HRESULT HardwiredGro
ups([out, retval] IGroupCollection* *pVal);
01624 [id(55), helpstring(hsISkype_CreateGro
up), helpcontext(13)] HRESULT CreateGroup([in] BST
R GroupName, [out, retval] IGroup** pGroup);
01625 [id(56), helpstring(hsISkype_DeleteGro
up), helpcontext(13)] HRESULT DeleteGroup([in] LON
G GroupId);
01626 [propget, id(57), helpstring(hsISkype_
Settings), helpcontext(13)] HRESULT Settings([out,
retval] ISettings* *pVal);
01627 [propget, id(58), helpstring(hsISkype_
```

```
Client), helpcontext(13)] HRESULT Client([out, ret  
val] IClient* *pVal);  
01628 [propget, id(59), helpstring(hsISkype_  
FriendlyName), helpcontext(13)] HRESULT FriendlyNam  
e([in] BSTR newVal);  
01629 [propget, id(60), helpstring(hsISkype_  
Command), helpcontext(13)] HRESULT Command([in] LO  
NG Id, [in] BSTR Command, [in, defaultvalue("")] B  
STR Reply, [in, defaultvalue(0)] VARIANT_BOOL Bloc  
k, [in, defaultvalue(30000)] LONG Timeout, [out, r  
etval] ICommand **pVal);  
01630 [propget, id(61), helpstring(hsISkype_  
Voicemail), helpcontext(13)] HRESULT Voicemail([in  
] LONG Id, [out, retval] IVoicemail** pVal);  
01631 [propget, id(62), helpstring(hsISkype_  
MissedVoicemails), helpcontext(13)] HRESULT Missed  
Voicemails([out, retval] IVoicemailCollection* *pV  
al);  
01632 [id(63), helpstring(hsISkype_EnableApi  
Context), helpcontext(13), hidden] HRESULT EnableA  
piSecurityContext([in] TApiSecurityContext Context  
);  
01633 [propget, id(64), helpstring(hsISkype_  
ApiContextEnabled), helpcontext(13), hidden] HRESU  
LT ApiSecurityContextEnabled([in] TApiSecurityCont  
ext Context, [out, retval] VARIANT_BOOL* pVal);  
01634 [id(65), helpstring(hsISkype_CreateSms  
) , helpcontext(13)] HRESULT CreateSms([in] TSmsMes  
sageType MessageType, [in] BSTR TargetNumbers, [ou  
t, retval] ISmsMessage** pMessage);  
01635 [propget, id(66), helpstring(hsISkype_  
Smss), helpcontext(13)] HRESULT Smss([out, retval]  
ISmsMessageCollection** pVal);  
01636 [propget, id(67), helpstring(hsISkype_  
MissedSmss), helpcontext(13)] HRESULT MissedSmss([  
out, retval] ISmsMessageCollection** pVal);  
01637 [id(68), helpstring(hsISkype_SendSms),
```

```

    helpcontext(13)] HRESULT SendSms([in] BSTR Target
Numbers, [in] BSTR MessageText, [in, defaultvalue(
"")] BSTR ReplyToNumber, [out, retval] ISmsMessage
** pMessage);
01638      [id(69), helpstring(hsISkype_AsyncSear
chUsers), helpcontext(13)] HRESULT AsyncSearchUser
s([in] BSTR Target, [out,retval] LONG* pCookie);
01639      [propget, id(70), helpstring(hsISkype_
WrapperVersion), helpcontext(13)] HRESULT ApiWrapp
erVersion([out, retval] BSTR* pVal);
01640      [propget, id(71), helpstring(hsISkype_
SilentMode), helpcontext(13)] HRESULT SilentMode([
out, retval] VARIANT_BOOL* pVal);
01641      [propput, id(71), helpstring(hsISkype_
SilentMode), helpcontext(13)] HRESULT SilentMode([
in] VARIANT_BOOL newVal);
01642      [propget, id(72), helpstring(hsISkype_
FileTransfers), helpcontext(13)] HRESULT FileTrans
fers([out, retval] IFileTransferCollection **pVal)
;
01643      [propget, id(73), helpstring(hsISkype_
ActiveFileTransfers), helpcontext(13)] HRESULT Act
iveFileTransfers([out, retval] IFileTransferCollec
tion **pVal);
01644      [propget, id(74), helpstring(hsISkype_
FocusedContacts), helpcontext(13)] HRESULT Focused
Contacts([out, retval] IUserCollection **pVal);
01645      [id(75), helpstring(hsISkype_BlobFind)
, helpcontext(13)] HRESULT FindChatUsingBlob([in]
BSTR Blob, [out,retval] IChat** pChat);
01646      [id(76), helpstring(hsISkype_BlobCreat
), helpcontext(13)] HRESULT CreateChatUsingBlob([i
n] BSTR Blob, [out,retval] IChat** pChat);
01647      [propget, id(77), helpstring(""), help
context(13)] HRESULT PredictiveDialerCountry([out,
retval] BSTR *pVal);
01648      };

```

```
01649 #ifndef DOXYGEN_SHOULD_SKIP_THIS
01650     [
01651         object,
01652         uuid(6D0B4547-771D-4C4F-B3E8-98A33FD24
01653         D2B),
01654         dual,
01655         nonextensible,
01656         pointer_default(unique),
01657         oleautomation,
01658         helpstring(hsISkypeSms),
01659         helpcontext(78),
01660         hidden
01661     ]
01662     interface ISkypeSms : IDispatch
01663     {
01664         [propget, id(1), helpstring(hsISmsMess
01665         age_Type), helpcontext(78)] HRESULT SmsMessageType
01666         ([in] BSTR SmsId, [out, retval] TSmsMessageTyp
01667         e *pVal);
01668         [propget, id(2), helpstring(hsISmsMess
01669         age_Status), helpcontext(78)] HRESULT SmsMessageSt
01670         atus([in] BSTR SmsId, [out, retval] TSmsMessageSta
01671         tus *pVal);
01672         [propget, id(3), helpstring(hsISmsMess
01673         age_FailureReason), helpcontext(78)] HRESULT SmsFa
01674         ilureReason([in] BSTR SmsId, [out, retval] TSmsFai
01675         lureReason *pVal);
01676         [propget, id(4), helpstring(hsISmsMess
01677         age_FailedUnseen), helpcontext(78)] HRESULT SmsIsF
01678         ailedUnseen([in] BSTR SmsId, [out, retval] VARIANT
01679         _BOOL *pVal);
01680         [propget, id(5), helpstring(hsISmsMess
01681         age_Seen), helpcontext(78)] HRESULT SmsSeen([in] B
01682         STR SmsId);
01683         [propget, id(6), helpstring(hsISmsMess
01684         age_Price), helpcontext(78)] HRESULT SmsPrice([in]
01685         BSTR SmsId, [out, retval] LONG *pVal);
```

```
01669      [propget, id(7), helpstring(hsISmsMess
age_PriceCurr), helpcontext(78)] HRESULT SmsPriceC
urrency([in] BSTR SmsId, [out, retval] BSTR *pVal)
;
01670      [propget, id(8), helpstring(hsISmsMess
age_PricePrec), helpcontext(78)] HRESULT SmsPriceP
recision([in] BSTR SmsId, [out, retval] LONG *pVal
);
01671      [propget, id(9), helpstring(hsISmsMess
age_ReplyTo), helpcontext(78)] HRESULT SmsReplyToN
umber([in] BSTR SmsId, [out, retval] BSTR* pVal);
01672      [propput, id(9), helpstring(hsISmsMess
age_ReplyTo), helpcontext(78)] HRESULT SmsReplyToN
umber([in] BSTR SmsId, [in] BSTR newVal);
01673      [propget, id(10), helpstring(hsISmsMes
sage_Targets), helpcontext(78)] HRESULT SmsTargets
([in] BSTR SmsId, [out, retval] ISmsTargetCollecti
on\*\* pVal);
01674      [propget, id(11), helpstring(hsISmsMes
sage_Body), helpcontext(78)] HRESULT SmsBody([in]
BSTR SmsId, [out, retval] BSTR* pVal);
01675      [propput, id(11), helpstring(hsISmsMes
sage_Body), helpcontext(78)] HRESULT SmsBody([in]
BSTR SmsId, [in] BSTR newVal);
01676      [propget, id(12), helpstring(hsISmsMes
sage_Chunks), helpcontext(78)] HRESULT SmsChunks([
in] BSTR SmsId, [out, retval] ISmsChunkCollection\*
\* pVal);
01677      [id(13), helpstring(hsISmsMessage_Send
), helpcontext(78)] HRESULT SendSmsMessage([in] BS
TR SmsId);
01678      [id(14), helpstring(hsISmsMessage_Dele
te), helpcontext(78)] HRESULT DeleteSmsMessage([in
] BSTR SmsId);
01679      [propget, id(15), helpstring(hsISmsTar
get_Status), helpcontext(78)] HRESULT SmsTargetSta
tus([in] BSTR SmsId, [in] BSTR Number, [out, retva
```

```

1] TSmsTargetStatus* pVal);
01680     [propget, id(16), helpstring(hsISmsChu
nk_Text), helpcontext(78)] HRESULT SmsChunkText([i
n] BSTR SmsId, [in] BSTR ChunkId, [out, retval] BS
TR* pVal);
01681     [propget, id(17), helpstring(hsISmsChu
nk_CharsLeft), helpcontext(78)] HRESULT SmsChunkCh
aractersLeft([in] BSTR SmsId, [in] BSTR ChunkId, [
out, retval] LONG* pVal);
01682     [propget, id(18), helpstring(hsISmsMes
sage_Timestamp), helpcontext(78)] HRESULT SmsTimes
tamp([in] BSTR SmsId, [out, retval] DATE* pVal);
01683     [propget, id(19), helpstring(hsISmsMes
sage_TargetNumbers), helpcontext(78)] HRESULT SmsT
argetNumbers([in] BSTR SmsId, [out, retval] BSTR*
pVal);
01684     [propput, id(19), helpstring(hsISmsMes
sage_TargetNumbers), helpcontext(78)] HRESULT SmsT
argetNumbers([in] BSTR SmsId, [in] BSTR newVal);
01685     };
01686     [
01687         object,
01688         uuid(54590FC3-0405-4A2D-B4A5-BEAC02657
7F2),
01689         dual,
01690         nonextensible,
01691         pointer_default(unique),
01692         oleautomation,
01693         helpstring(hsISkypeCall),
01694         helpcontext(45),
01695         hidden
01696     ]
01697     interface ISkypeCall : IDispatch
01698     {
01699         [propget, id(1), helpstring(hsICall_Ti
mestamp), helpcontext(45), ] HRESULT CallTimestamp
([in] BSTR CallId, [out, retval] DATE *pVal);

```

```
01700      [propget, id(2), helpstring(hsICall_Pa
rtHandle), helpcontext(45)] HRESULT CallPartnerHan
dle([in] BSTR CallId, [out, retval] BSTR *pVal);
01701      [propget, id(3), helpstring(hsICall_Pa
rtDispName), helpcontext(45)] HRESULT CallPartnerD
isplayName([in] BSTR CallId, [out, retval] BSTR *p
Val);
01702      [propget, id(4), helpstring(hsICall_Co
nferenceId), helpcontext(45)] HRESULT CallConferen
ceId([in] BSTR CallId, [out, retval] BSTR *pVal);
01703      [propget, id(5), helpstring(hsICall_Ty
pe), helpcontext(45)] HRESULT CallType([in] BSTR C
allId, [out, retval] TCallType *pVal);
01704      [propget, id(6), helpstring(hsICall_St
atus), helpcontext(45)] HRESULT CallStatus([in] BS
TR CallId, [out, retval] TCallStatus *pVal);
01705      [propput, id(6), helpstring(hsICall_St
atus), helpcontext(45)] HRESULT CallStatus([in] BS
TR CallId, [in] TCallStatus newVal);
01706      [propget, id(7), helpstring(hsICall_Fa
ilReason), helpcontext(45)] HRESULT CallFailureRea
son([in] BSTR CallId, [out, retval] TCallFailureRe
ason *pVal);
01707      [propget, id(8), helpstring(hsICall_Su
bject), helpcontext(45)] HRESULT CallSubject([in]
BSTR CallId, [out, retval] BSTR *pVal);
01708      [propget, id(9), helpstring(hsICall_PS
TN_Number), helpcontext(45)] HRESULT CallPstnNumbe
r([in] BSTR CallId, [out, retval] BSTR *pVal);
01709      [propget, id(10), helpstring(hsICall_D
uration), helpcontext(45)] HRESULT CallDuration([i
n] BSTR CallId, [out, retval] LONG *pVal);
01710      [propget, id(11), helpstring(hsICall_P
STN_Status), helpcontext(45)] HRESULT CallPstnStat
us([in] BSTR CallId, [out, retval] BSTR *pVal);
01711      [propput, id(12), helpstring(hsICall_S
een), helpcontext(45)] HRESULT CallSeenStatus([in]
```



```
BSTR CallId, [in] VARIANT_BOOL newVal);
01712      [id(13), helpstring(hsICall_Hold), helpcontext(45)] HRESULT CallHold([in] BSTR CallId);
01713      [id(14), helpstring(hsICall_Resume), helpcontext(45)] HRESULT CallResume([in] BSTR CallId);
01714      [id(15), helpstring(hsICall_Finish), helpcontext(45)] HRESULT CallFinish([in] BSTR CallId);
01715      [id(16), helpstring(hsICall_Answer), helpcontext(45)] HRESULT CallAnswer([in] BSTR CallId);
01716      [propget, id(17), helpstring(hsICall_DTMF), helpcontext(45)] HRESULT CallDTMF([in] BSTR CallId, [in] BSTR newVal);
01717      [propget, id(18), helpstring(hsICall_Participants), helpcontext(45)] HRESULT CallParticipants([in] BSTR CallId, [out, retval] IParticipantCollection **pVal);
01718      [id(19), helpstring(hsICall_Join), helpcontext(45)] HRESULT CallJoin([in] BSTR CallId1, [in] BSTR CallId2);
01719      [propget, id(20), helpstring(hsICall_VmDuration), helpcontext(45)] HRESULT CallVmDuration([in] BSTR CallId, [out, retval] LONG *pVal);
01720      [propget, id(21), helpstring(hsICall_VmAllowDuration), helpcontext(45)] HRESULT CallVmAllowedDuration([in] BSTR CallId, [out, retval] LONG *pVal);
01721      [propget, id(22), helpstring(hsICall_VStatus), helpcontext(45)] HRESULT CallVideoStatus([in] BSTR CallId, [out, retval] TCallVideoStatus *pVal);
01722      [propget, id(23), helpstring(hsICall_VSendStatus), helpcontext(45)] HRESULT CallVideoSendStatus([in] BSTR CallId, [out, retval] TCallVideoSendStatus *pVal);
```

```
01723      [propget, id(24), helpstring(hsICall_V
RecvStatus), helpcontext(45)] HRESULT CallVideoRec
eiveStatus([in] BSTR CallId, [out, retval] TCallVi
deoSendStatus *pVal);
01724      [id(25), helpstring(hsICall_StartVSend
), helpcontext(45)] HRESULT StartCallVideoSend([in
] BSTR CallId);
01725      [id(26), helpstring(hsICall_StopVSend)
, helpcontext(45)] HRESULT StopCallVideoSend([in]
BSTR CallId);
01726      [id(27), helpstring(hsICall_StartVRecv
), helpcontext(45)] HRESULT StartCallVideoReceive(
[in] BSTR CallId);
01727      [id(28), helpstring(hsICall_StopVRecv)
, helpcontext(45)] HRESULT StopCallVideoReceive([i
n] BSTR CallId);
01728      [id(29), helpstring(hsICall_Redirect),
helpcontext(45)] HRESULT RedirectCallToVoicemail(
[in] BSTR CallId);
01729      [id(30), helpstring(hsICall_Forward),
helpcontext(45)] HRESULT ForwardCall([in] BSTR Cal
lId);
01730      [propget, id(31), helpstring(hsICall_R
ate), helpcontext(45)] HRESULT CallRate([in] BSTR
CallId, [out, retval] LONG *pVal);
01731      [propget, id(32), helpstring(hsICall_R
ateCurr), helpcontext(45)] HRESULT CallRateCurrenc
y([in] BSTR CallId, [out, retval] BSTR *pVal);
01732      [propget, id(33), helpstring(hsICall_R
atePrecision), helpcontext(45)] HRESULT CallRatePr
ecision([in] BSTR CallId, [out, retval] LONG *pVal
);
01733      [propget, id(34), helpstring(hsICall_I
nputDevice), helpcontext(45)] HRESULT CallInputDev
ice([in] BSTR CallId, [in] TCallIoDeviceType Devic
eType, [out, retval] BSTR* pVal);
01734      [propput, id(34), helpstring(hsICall_I
```

```
nputDevice), helpcontext(45)] HRESULT CallInputDev  
ice([in] BSTR CallId, [in] TCallIoDeviceType Devic  
eType, [in] BSTR newVal);  
01735 [propget, id(35), helpstring(hsICall_0  
utputDevice), helpcontext(45)] HRESULT CallOutputD  
evice([in] BSTR CallId, [in] TCallIoDeviceType Dev  
iceType, [out, retval] BSTR* pVal);  
01736 [propput, id(35), helpstring(hsICall_0  
utputDevice), helpcontext(45)] HRESULT CallOutputD  
evice([in] BSTR CallId, [in] TCallIoDeviceType Dev  
iceType, [in] BSTR newVal);  
01737 [propget, id(36), helpstring(hsICall_M  
icDevice), helpcontext(45)] HRESULT CallCaptureMic  
Device([in] BSTR CallId, [in] TCallIoDeviceType De  
viceType, [out, retval] BSTR* pVal);  
01738 [propput, id(36), helpstring(hsICall_M  
icDevice), helpcontext(45)] HRESULT CallCaptureMic  
Device([in] BSTR CallId, [in] TCallIoDeviceType De  
viceType, [in] BSTR newVal);  
01739 [propget, id(37), helpstring(hsICall_I  
nputStatus), helpcontext(45)] HRESULT CallInputSta  
tus([in] BSTR CallId, [out, retval] VARIANT_BOOL*  
pVal);  
01740 [propget, id(38), helpstring(hsICall_F  
orwardedBy), helpcontext(45)] HRESULT CallForwarde  
dBy([in] BSTR CallId, [out, retval] BSTR* pVal);  
01741 [propget, id(12), helpstring(hsICall_S  
een), helpcontext(45)] HRESULT CallSeenStatus([in]  
BSTR CallId, [out, retval] VARIANT_BOOL *pVal);  
01742 [propget, id(39), helpstring(hsICall_C  
anTransfer), helpcontext(45)] HRESULT CallCanTrans  
fer([in] BSTR CallId, [in] BSTR Target, [out, retv  
al] VARIANT_BOOL *pVal);  
01743 [propget, id(40), helpstring(hsICall_T  
ransferStatus), helpcontext(45)] HRESULT CallTrans  
ferStatus([in] BSTR CallId, [out, retval] TCallSta  
tus *pVal);
```

```

01744      [propget, id(41), helpstring(hsICall_T
ransferActive), helpcontext(45)] HRESULT CallTrans
ferActive([in] BSTR CallId, [out, retval] VARIANT_
BOOL *pVal);
01745      [propget, id(42), helpstring(hsICall_T
ransferredBy), helpcontext(45)] HRESULT CallTransf
erredBy([in] BSTR CallId, [out, retval] BSTR *pVal
);
01746      [propget, id(43), helpstring(hsICall_T
ransferredTo), helpcontext(45)] HRESULT CallTransf
erredTo([in] BSTR CallId, [out, retval] BSTR *pVal
);
01747      [id(44), helpstring(hsICall_Transfer),
helpcontext(45)] HRESULT TransferCall([in] BSTR C
allId, [in] BSTR Target);
01748      [propget, id(46), helpstring(hsICall_T
argetIdentity), helpcontext(45)] HRESULT CallTarge
tIdentity([in] BSTR CallId, [out, retval] BSTR *pV
al);
01749      };
01750      [
01751          object,
01752          uuid(C219279C-F557-4BAD-B3BE-750E91CA9
944),
01753          dual,
01754          nonextensible,
01755          pointer_default(unique),
01756          oleautomation,
01757          helpstring(hsISkypeUser),
01758          helpcontext(46),
01759          hidden
01760      ]
01761      interface ISkypeUser : IDispatch
01762      {
01763          [propget, id(1), helpstring(hsIUser_Fu
llName), helpcontext(46)] HRESULT UserFullName([in
] BSTR Username, [out, retval] BSTR *pVal);

```

```
01764      [propget, id(2), helpstring(hsIUser_Bi
rthday), helpcontext(46)] HRESULT UserBirthday([in
] BSTR Username, [out, retval] DATE *pVal);
01765      [propget, id(3), helpstring(hsIUser_Se
x), helpcontext(46)] HRESULT UserSex([in] BSTR Use
rname, [out, retval] TUserSex *pVal);
01766      [propget, id(4), helpstring(hsIUser_Co
untry), helpcontext(46)] HRESULT UserCountry([in]
BSTR Username, [out, retval] BSTR *pVal);
01767      [propget, id(5), helpstring(hsIUser_Pr
ovince), helpcontext(46)] HRESULT UserProvince([in
] BSTR Username, [out, retval] BSTR *pVal);
01768      [propget, id(6), helpstring(hsIUser_Ci
ty), helpcontext(46)] HRESULT UserCity([in] BSTR U
sername, [out, retval] BSTR *pVal);
01769      [propget, id(7), helpstring(hsIUser_Ph
oneHome), helpcontext(46)] HRESULT UserPhoneHome([
in] BSTR Username, [out, retval] BSTR *pVal);
01770      [propget, id(8), helpstring(hsIUser_Ph
oneOffice), helpcontext(46)] HRESULT UserPhoneOffi
ce([in] BSTR Username, [out, retval] BSTR *pVal);
01771      [propget, id(9), helpstring(hsIUser_Ph
oneMobile), helpcontext(46)] HRESULT UserPhoneMobi
le([in] BSTR Username, [out, retval] BSTR *pVal);
01772      [propget, id(10), helpstring(hsIUser_H
omepage), helpcontext(46)] HRESULT UserHomepage([i
n] BSTR Username, [out, retval] BSTR *pVal);
01773      [propget, id(11), helpstring(hsIUser_A
bout), helpcontext(46)] HRESULT UserAbout([in] BST
R Username, [out, retval] BSTR *pVal);
01774      [propget, id(12), helpstring(hsIUser_H
asCallEquipment), helpcontext(46)] HRESULT UserHas
CallEquipment([in] BSTR Username, [out, retval] VA
RIANT_BOOL *pVal);
01775      [propget, id(13), helpstring(hsIUser_B
uddyStatus), helpcontext(46)] HRESULT UserBuddySta
tus([in] BSTR Username, [out, retval] TBuddyStatus
```

```

    *pVal);
01776      [propput, id(13), helpstring(hsIUser_B
uddyStatus), helpcontext(46)] HRESULT UserBuddySta
tus([in] BSTR Username, [in] TBuddyStatus newVal);
01777      [propget, id(14), helpstring(hsIUser_A
uthorized), helpcontext(46)] HRESULT UserIsAuthori
zed([in] BSTR Username, [out, retval] VARIANT_BOOL
    *pVal);
01778      [propput, id(14), helpstring(hsIUser_A
uthorized), helpcontext(46)] HRESULT UserIsAuthori
zed([in] BSTR Username, [in] VARIANT_BOOL newVal);
01779      [propget, id(15), helpstring(hsIUser_B
locked), helpcontext(46)] HRESULT UserIsBlocked([i
n] BSTR Username, [out, retval] VARIANT_BOOL *pVal
);
01780      [propput, id(15), helpstring(hsIUser_B
locked), helpcontext(46)] HRESULT UserIsBlocked([i
n] BSTR Username, [in] VARIANT_BOOL newVal);
01781      [propget, id(16), helpstring(hsIUser_D
isplayName), helpcontext(46)] HRESULT UserDisplayN
ame([in] BSTR Username, [out, retval] BSTR *pVal);
01782      [propget, id(17), helpstring(hsIUser_O
nlineStatus), helpcontext(46)] HRESULT UserOnlineS
tatus([in] BSTR Username, [out, retval] TOnlineSta
tus *pVal);
01783      [propget, id(18), helpstring(hsIUser_L
astOnline), helpcontext(46)] HRESULT UserLastOnlin
e([in] BSTR Username, [out, retval] DATE *pVal);
01784      [propget, id(19), helpstring(hsIUser_C
ountryCode), helpcontext(46)] HRESULT UserCountryC
ode([in] BSTR Username, [out, retval] BSTR *pVal);
01785      [propget, id(20), helpstring(hsIUser_R
ecvAuthReq), helpcontext(46)] HRESULT UserReceived
AuthRequest([in] BSTR Username, [out, retval] BSTR
    *pVal);
01786      [propget, id(21), helpstring(hsIUser_S
peedDial), helpcontext(46)] HRESULT UserSpeedDial(

```

```
[in] BSTR Username, [out, retval] BSTR *pVal);
01787     [propput, id(21), helpstring(hsIUser_SpeedDial), helpcontext(46)] HRESULT UserSpeedDial(
[in] BSTR Username, [in] BSTR newVal);
01788     [propget, id(22), helpstring(hsIUser_CanLeaveVm), helpcontext(46)] HRESULT UserCanLeaveV
oicemail([in] BSTR Username, [out, retval] VARIANT
_BOOL *pVal);
01789     [propget, id(23), helpstring(hsIUser_MoodText), helpcontext(46)] HRESULT UserMoodText([i
n] BSTR Username, [out, retval] BSTR *pVal);
01790     [propget, id(24), helpstring(hsIUser_Aliases), helpcontext(46)] HRESULT UserAliases([in]
BSTR Username, [out, retval] BSTR *pVal);
01791     [propget, id(25), helpstring(hsIUser_Timezone), helpcontext(46)] HRESULT UserTimezone([i
n] BSTR Username, [out, retval] LONG *pVal);
01792     [propget, id(26), helpstring(hsIUser_IsCFActive), helpcontext(46)] HRESULT IsCallForward
Active([in] BSTR Username, [out, retval] VARIANT_B
OOL *pVal);
01793     [propget, id(27), helpstring(hsIUser_Language), helpcontext(46)] HRESULT UserLanguage([i
n] BSTR Username, [out, retval] BSTR *pVal);
01794     [propget, id(28), helpstring(hsIUser_LanguageCode), helpcontext(46)] HRESULT UserLanguag
eCode([in] BSTR Username, [out, retval] BSTR *pVal
);
01795     [propget, id(29), helpstring(hsIUser_VideoCapable), helpcontext(46)] HRESULT UserIsVideo
Capable([in] BSTR Username, [out, retval] VARIANT_
BOOL *pVal);
01796     [propget, id(30), helpstring(hsIUser_NumberOfAuthBuddies), helpcontext(46)] HRESULT User
NumberOfAuthBuddies([in] BSTR Username, [out, retv
al] LONG* pVal);
01797     [propput, id(16), helpstring(hsIUser_D
```

```

isplayName), helpcontext(46)] HRESULT UserDisplayN
ame([in] BSTR Username, [in] BSTR newVal);
01798      [propget, id(31), helpstring(hsIUser_R
ichMoodText), helpcontext(46)] HRESULT UserRichMoo
dText([in] BSTR Username, [out, retval] BSTR *pVal
);
01799      [propget, id(32), helpstring(hsIUser_I
sVmCapable), helpcontext(46)] HRESULT UserIsVoicem
ailCapable([in] BSTR Username, [out, retval] VARIA
NT_BOOL *pVal);
01800      };
01801      [
01802          object,
01803          uuid(A569B841-CC8A-4C12-B937-CBC17D9E6
4F0),
01804          dual,
01805          nonextensible,
01806          pointer_default(unique),
01807          oleautomation,
01808          helpstring(hsISkypeChatMessage),
01809          helpcontext(47),
01810          hidden
01811      ]
01812      interface ISkypeChatMessage : IDispatch
01813      {
01814          [propget, id(1), helpstring(hsIChatMsg
_Timestamp), helpcontext(47)] HRESULT ChatMessageTi
mestamp([in] BSTR MessageId, [out, retval] DATE *p
Val);
01815          [propget, id(2), helpstring(hsIChatMsg
_FromHandle), helpcontext(47)] HRESULT ChatMessageF
romHandle([in] BSTR MessageId, [out, retval] BSTR
*pVal);
01816          [propget, id(3), helpstring(hsIChatMsg
_FromDispName), helpcontext(47)] HRESULT ChatMessag
eFromDisplayName([in] BSTR MessageId, [out, retval
] BSTR *pVal);

```



```
01817      [propget, id(4), helpstring(hsIChatMsg
_Type),helpcontext(47)] HRESULT ChatMessageType([i
n] BSTR MessageId, [out, retval] TChatMessageType
*pVal);
01818      [propget, id(5), helpstring(hsIChatMsg
_Status),helpcontext(47)] HRESULT ChatMessageStatu
s([in] BSTR MessageId, [out, retval] TChatMessageS
tatus *pVal);
01819      [propget, id(6), helpstring(hsIChatMsg
_LeaveReason),helpcontext(47)] HRESULT ChatMessage
LeaveReason([in] BSTR MessageId, [out, retval] TCh
atLeaveReason *pVal);
01820      [propget, id(7), helpstring(hsIChatMsg
_Body),helpcontext(47)] HRESULT ChatMessageBody([i
n] BSTR MessageId, [out, retval] BSTR *pVal);
01821      [propget, id(8), helpstring(hsIChatMsg
_ChatName),helpcontext(47)] HRESULT ChatMessageCha
tname([in] BSTR MessageId, [out, retval] BSTR *pVa
l);
01822      [propget, id(9), helpstring(hsIChatMsg
_Users),helpcontext(47)] HRESULT ChatMessageUsers(
[in] BSTR MessageId, [out, retval] IUserCollection
* *pVal);
01823      [propput, id(10), helpstring(hsIChatMs
g_Seen),helpcontext(47)] HRESULT ChatMessageSeen([
in] BSTR MessageId);
01824      [propget, id(11), helpstring(hsIChatMs
g_Sender), helpcontext(47)] HRESULT ChatMessageSen
der([in] BSTR MessageId, [out, retval] IUser* *pVa
l);
01825      [propget, id(12), helpstring(hsIChatMs
g_Editable), helpcontext(47)] HRESULT ChatMessageI
sEditable([in] BSTR MessageId, [out, retval] VARIAN
T_BOOL* pVal);
01826      [propget, id(13), helpstring(hsIChatMs
g_EditedBy), helpcontext(47)] HRESULT ChatMessageE
ditedBy([in] BSTR MessageId, [out, retval] BSTR* p
```

```

Val);
01827      [propget, id(14), helpstring(hsIChatMs
g_EditTime), helpcontext(47)] HRESULT ChatMessageE
ditedTimestamp(BSTR MessageId, [out, retval] DATE*
pVal);
01828      [propput, id(7), helpstring(hsIChatMsg
_Body), helpcontext(47)] HRESULT ChatMessageBody([
in] BSTR MessageId, [in] BSTR newVal);
01829      [propget, id(15), helpstring(hsIChatMs
g_Role), helpcontext(47)] HRESULT ChatMessageRole(
[in] BSTR MessageId, [out, retval] TChatMemberRole
* pVal);
01830      [propget, id(16), helpstring(hsIChatMs
g_Options), helpcontext(47)] HRESULT ChatMessageOp
tions([in] BSTR MessageId, [out, retval] LONG* pVal
);
01831      };
01832      [
01833          object,
01834          uuid(6CF6DBDE-AB7C-4635-96ED-2FF525AB4
457),
01835          dual,
01836          nonextensible,
01837          pointer_default(unique),
01838          oleautomation,
01839          helpstring(hsIChatMember),
01840          helpcontext(0),
01841          hidden
01842      ]
01843      interface ISkypeChatMember : IDispatch
01844      {
01845          [propget, id(1), helpstring(hsIChatMem
ber_Id)] HRESULT ChatMemberHandle([in] BSTR Member
Id, [out, retval] BSTR *pVal);
01846          [propget, id(2), helpstring(hsIChatMem
ber_Role)] HRESULT ChatMemberRole([in] BSTR Member
Id, [out, retval] TChatMemberRole* pVal);

```

```

01847         [propget, id(2), helpstring(hsIChatMem
ber_Role)] HRESULT ChatMemberRole([in] BSTR Member
Id, [in] TChatMemberRole newVal);
01848         [propget, id(3), helpstring(hsIChatMem
ber_CanSetRole)] HRESULT CanSetChatMemberRoleTo([i
n] BSTR MemberId, [in] TChatMemberRole Role, [out,
retval] VARIANT_BOOL* pVal);
01849         [propget, id(4), helpstring(hsIChatMem
ber_IsActive)] HRESULT IsChatMemberActive([in] BST
R MemberId, [out, retval] VARIANT_BOOL* pVal);
01850         [propget, id(5), helpstring(hsIChatMem
ber_Chat)] HRESULT ChatMemberChat([in] BSTR Member
Id, [out, retval] IChat** pVal);
01851     };
01852     [
01853         object,
01854         uuid(30678F8A-C104-40C0-A6AE-7C150E833
00E),
01855         dual,
01856         nonextensible,
01857         pointer_default(unique),
01858         oleautomation,
01859         helpstring(hsISkypeConference),
01860         helpcontext(49),
01861         hidden
01862     ]
01863     interface ISkypeConference : IDispatch
01864     {
01865         [propget, id(1), helpstring(hsIConfere
nce_Calls), helpcontext(49)] HRESULT ConferenceCal
ls([in] BSTR ConfId, [out, retval] ICallCollection
**pVal);
01866         [propget, id(2), helpstring(hsIConfere
nce_ActiveCalls), helpcontext(49)] HRESULT Confere
nceActiveCalls([in] BSTR ConfId, [out, retval] ICa
llCollection **pVal);
01867     };

```

```

01868     [
01869         object,
01870         uuid(4D33E14E-9921-4860-92F4-5DC1586F4
01871         03C),
01872         dual,
01873         nonextensible,
01874         pointer_default(unique),
01875         oleautomation,
01876         helpstring(hsISkypeVoicemail),
01877         helpcontext(50),
01878         hidden
01879     ]
01880     interface ISkypeVoicemail : IDispatch
01881     {
01882         [propget, id(1), helpstring(hsIVoicema
01883         il_Type), helpcontext(50)] HRESULT VoicemailType([
01884         in] BSTR VoicemailId, [out, retval] TVoicemailType
01885         *pVal);
01886         [propget, id(2), helpstring(hsIVoicema
01887         il_PartHandle), helpcontext(50)] HRESULT Voicemail
01888         PartnerHandle([in] BSTR VoicemailId, [out, retval]
01889         BSTR *pVal);
01890         [propget, id(3), helpstring(hsIVoicema
01891         il_PartDispName), helpcontext(50)] HRESULT Voicema
01892         ilPartnerDisplayName([in] BSTR VoicemailId, [out,
01893         retval] BSTR *pVal);
01894         [propget, id(4), helpstring(hsIVoicema
01895         il_Status), helpcontext(50)] HRESULT VoicemailStat
01896         us([in] BSTR VoicemailId, [out, retval] TVoicemail
01897         Status *pVal);
01898         [propget, id(5), helpstring(hsIVoicema
01899         il_FailureReason), helpcontext(50)] HRESULT Voicem
01900         ailFailureReason([in] BSTR VoicemailId, [out, retv
01901         al] TVoicemailFailureReason *pVal);
01902         [propget, id(6), helpstring(hsIVoicema
01903         il_Timestamp), helpcontext(50)] HRESULT VoicemailT
01904         imESTAMP([in] BSTR VoicemailId, [out, retval] DATE

```

```
*pVal);
01887      [propget, id(7), helpstring(hsIVoicemail_Duration), helpcontext(50)] HRESULT VoicemailDuration([in] BSTR VoicemailId, [out, retval] LONG *pVal);
01888      [propget, id(8), helpstring(hsIVoicemail_AllowedDuration), helpcontext(50)] HRESULT VoicemailAllowedDuration([in] BSTR VoicemailId, [out, retval] LONG *pVal);
01889      [id(9), helpstring(hsIVoicemail_Open), helpcontext(50)] HRESULT OpenVoicemail([in] BSTR VoicemailId);
01890      [id(10), helpstring(hsIVoicemail_StartPlay), helpcontext(50)] HRESULT StartVoicemailPlayback([in] BSTR VoicemailId);
01891      [id(11), helpstring(hsIVoicemail_StopPlay), helpcontext(50)] HRESULT StopVoicemailPlayback([in] BSTR VoicemailId);
01892      [id(12), helpstring(hsIVoicemail_Upload), helpcontext(50)] HRESULT UploadVoicemail([in] BSTR VoicemailId);
01893      [id(13), helpstring(hsIVoicemail_Download), helpcontext(50)] HRESULT DownloadVoicemail([in] BSTR VoicemailId);
01894      [id(14), helpstring(hsIVoicemail_StartRec), helpcontext(50)] HRESULT StartVoicemailRecording([in] BSTR VoicemailId);
01895      [id(15), helpstring(hsIVoicemail_StopRec), helpcontext(50)] HRESULT StopVoicemailRecording([in] BSTR VoicemailId);
01896      [id(16), helpstring(hsIVoicemail_Delete), helpcontext(50)] HRESULT DeleteVoicemail([in] BSTR VoicemailId);
01897      [id(17), helpstring(hsIVoicemail_PlayInCall), helpcontext(50)] HRESULT StartVoicemailPlaybackInCall([in] BSTR VoicemailId);
01898      [id(18), helpstring(hsIVoicemail_Unpla
```

```

yed), helpcontext(50)] HRESULT SetVoicemailUnplaye
d([in] BSTR VoicemailId);
01899     };
01900     [
01901         object,
01902         uuid(55A5200C-B2E8-4795-A6FA-858CA3FE2
F26),
01903         dual,
01904         nonextensible,
01905         pointer_default(unique),
01906         oleautomation,
01907         helpstring(hsISkypeApplication),
01908         helpcontext(3),
01909         hidden
01910     ]
01911     interface ISkypeApplication : IDispatch
01912     {
01913         [id(1), helpstring(hsIApp_Create),help
context(3)] HRESULT CreateApplication([in] BSTR Na
me);
01914         [id(2), helpstring(hsIApp_Delete),help
context(3)] HRESULT DeleteApplication([in] BSTR Na
me);
01915         [propget, id(3), helpstring(hsIApp_Str
eams),helpcontext(3)] HRESULT ApplicationStreams([
in] BSTR AppName, [out, retval] IApplicationStream
Collection **pVal);
01916         [id(4), helpstring(hsIApp_Connect),hel
pcontext(3)] HRESULT ConnectApplicationUser([in] B
STR AppName, [in] BSTR Username, [in, defaultvalue
(0)] VARIANT_BOOL WaitConnected);
01917         [id(5), helpstring(hsIAppStm_Read),hel
pcontext(3)] HRESULT ReadApplicationStream([in] BS
TR AppName, [in] BSTR StreamHandle, [out, retval]
BSTR* pVal);
01918         [id(6), helpstring(hsIAppStm_Write),he
lpcontext(3)] HRESULT WriteApplicationStream([in]

```

```

BSTR AppName, [in] BSTR StreamHandle, [in] BSTR Text);
01919      [propget, id(7), helpstring(hsIApp_ConnectableUsers),helpcontext(3)] HRESULT ApplicationConnectableUsers([in] BSTR AppName, [out, retval] IUserCollection* *pVal);
01920      [propget, id(8), helpstring(hsIApp_ConnectingUsers),helpcontext(3)] HRESULT ApplicationConnectingUsers([in] BSTR AppName, [out, retval] IUserCollection* *pVal);
01921      [id(9), helpstring(hsIApp_Disconnect),helpcontext(3)] HRESULT DisconnectApplicationUser([in] BSTR AppName, [in] BSTR Username);
01922      [id(10), helpstring(hsIAppStm_SendData),helpcontext(3)] HRESULT SendApplicationDatagram([in] BSTR AppName, [in] BSTR StreamHandle, [in] BSTR Text);
01923      [propget, id(11), helpstring(hsIApp_SendStreams),helpcontext(3)] HRESULT ApplicationSendingStreams([in] BSTR AppName, [out, retval] IApplicationStreamCollection **pVal);
01924      [propget, id(12), helpstring(hsIApp_RecvStreams),helpcontext(3)] HRESULT ApplicationReceivedStreams([in] BSTR AppName, [out, retval] IApplicationStreamCollection **pVal);
01925      };
01926      [
01927          object,
01928          uuid(2B929860-6AF4-40DF-9D77-308CA7C23E66),
01929          dual,
01930          pointer_default(unique),
01931          helpstring(""),
01932          helpcontext(0),
01933          hidden
01934      ]
01935      interface ISkypeFileTransfer : IDispatch

```

```
01936     {
01937         [propget, id(1), helpstring(hsIFileTra
nsfer_Type), helpcontext(0)] HRESULT FileTransferT
ype([in] BSTR TransferId, [out, retval] TFileTrans
ferType* pVal);
01938         [propget, id(2), helpstring(hsIFileTra
nsfer_Status), helpcontext(0)] HRESULT FileTransfe
rStatus([in] BSTR TransferId, [out, retval] TFileT
ransferStatus* pVal);
01939         [propget, id(3), helpstring(hsIFileTra
nsfer_FailReason), helpcontext(0)] HRESULT FileTra
nsferFailureReason([in] BSTR TransferId, [out, ret
val] TFileTransferFailureReason* pVal);
01940         [propget, id(4), helpstring(hsIFileTra
nsfer_PartnerHandle), helpcontext(0)] HRESULT File
TransferPartnerHandle([in] BSTR TransferId, [out,
retval] BSTR* pVal);
01941         [propget, id(5), helpstring(hsIFileTra
nsfer_PartnerDispName), helpcontext(0)] HRESULT Fi
leTransferPartnerDisplayName([in] BSTR TransferId,
 [out, retval] BSTR* pVal);
01942         [propget, id(6), helpstring(hsIFileTra
nsfer_StartTime), helpcontext(0)] HRESULT FileTran
sferStartTime([in] BSTR TransferId, [out, retval]
DATE* pVal);
01943         [propget, id(7), helpstring(hsIFileTra
nsfer_FinishTime), helpcontext(0)] HRESULT FileTra
nsferFinishTime([in] BSTR TransferId, [out, retval
] DATE* pVal);
01944         [propget, id(8), helpstring(hsIFileTra
nsfer_FilePath), helpcontext(0)] HRESULT FileTrans
ferFilePath([in] BSTR TransferId, [out, retval] BS
TR* pVal);
01945         [propget, id(9), helpstring(hsIFileTra
nsfer_FileName), helpcontext(0)] HRESULT FileTrans
ferFileName([in] BSTR TransferId, [out, retval] BS
TR* pVal);
```



```

01946         [propget, id(10), helpstring(hsIFileTr
ansfer_ByteSec), helpcontext(0)] HRESULT FileTrans
ferBytesPerSecond([in] BSTR TransferId, [out, retv
al] BSTR* pVal);
01947         [propget, id(11), helpstring(hsIFileTr
ansfer_ByteTransferred), helpcontext(0)] HRESULT F
ileTransferBytesTransferred([in] BSTR TransferId,
[out, retval] BSTR* pVal);
01948     };
01949 #endif /*DOXYGEN_SHOULD_SKIP_THIS*/
01950     [
01951         uuid(F4F90CDD-C620-4118-945E-CAA1BBEBA
435),
01952         helpstring(hsISkypeEvents),
01953         helpcontext(4)
01954     ]
01955     dispinterface _ISkypeEvents
01956     {
01957         properties:
01958         methods:
01959         [id(1), helpstring(hsISkypeEvents_Comm
and),helpcontext(4)] HRESULT Command([in] ICommand
* pCommand);
01960         [id(2), helpstring(hsISkypeEvents_Repl
y),helpcontext(4)] HRESULT Reply([in] ICommand* pC
ommand);
01961         [id(3), helpstring(hsISkypeEvents_Erro
r),helpcontext(4)] HRESULT Error([in] ICommand* pC
ommand, [in] LONG Number, [in] BSTR Description);
01962         [id(4), helpstring(hsISkypeEvents_Atta
chStat),helpcontext(4)] HRESULT AttachmentStatus([
in] TAttachmentStatus Status);
01963         [id(5), helpstring(hsISkypeEvents_Conn
Stat),helpcontext(4)] HRESULT ConnectionStatus([in
] TConnectionStatus Status);
01964         [id(6), helpstring(hsISkypeEvents_User
Stat),helpcontext(4)] HRESULT UserStatus([in] TUse

```

```
rStatus Status);
01965      [id(7), helpstring(hsISkypeEvents_Onli
neStat),helpcontext(4)] HRESULT OnlineStatus([in]
IUser* pUser, [in] TOnlineStatus Status);
01966      [id(8), helpstring(hsISkypeEvents_Call
Stat),helpcontext(4)] HRESULT CallStatus([in] ICall
* pCall, [in] TCallStatus Status);
01967      [id(9), helpstring(hsISkypeEvents_Call
Hist),helpcontext(4)] HRESULT CallHistory();
01968      [id(10), helpstring(hsISkypeEvents_Mut
e),helpcontext(4)] HRESULT Mute([in] VARIANT_BOOL
Mute);
01969      [id(11), helpstring(hsISkypeEvents_Mes
sStat),helpcontext(4)] HRESULT MessageStatus([in]
IChatMessage* pMessage, [in] TChatMessageStatus St
atus);
01970      [id(12), helpstring(hsISkypeEvents_Mes
sHist),helpcontext(4)] HRESULT MessageHistory([in]
BSTR Username);
01971      [id(13), helpstring(hsISkypeEvents_Aut
oAway),helpcontext(4)] HRESULT AutoAway([in] VARIA
NT_BOOL Automatic);
01972      [id(14), helpstring(hsISkypeEvents_Cal
lDtmf),helpcontext(4)] HRESULT CallDtmfReceived([i
n] ICall* pCall, [in] BSTR Code);
01973      [id(15), helpstring(hsISkypeEvents_Vma
ilStat),helpcontext(4)] HRESULT VoicemailStatus([i
n] IVoicemail* pMail, [in] TVoicemailStatus Status
);
01974      [id(16), helpstring(hsISkypeEvents_App
ConnUsr),helpcontext(4)] HRESULT ApplicationConnec
ting([in] IApplication* pApp, [in] IUserCollection
* pUsers);
01975      [id(17), helpstring(hsISkypeEvents_App
Streams),helpcontext(4)] HRESULT ApplicationStream
s([in] IApplication* pApp, [in] IApplicationStream
Collection* pStreams);
```

01976 [id(18), helpstring(hsISkypeEvents\_AppDatagram),helpcontext(4)] HRESULT ApplicationDatagram([in] IApplication\* pApp, [in] IApplicationStream\* pStream, [in] BSTR Text);

01977 [id(19), helpstring(hsISkypeEvents\_AppSending),helpcontext(4)] HRESULT ApplicationSending([in] IApplication\* pApp, [in] IApplicationStreamCollection\* pStreams);

01978 [id(20), helpstring(hsISkypeEvents\_AppReceiving),helpcontext(4)] HRESULT ApplicationReceiving([in] IApplication\* pApp, [in] IApplicationStreamCollection\* pStreams);

01979 [id(21), helpstring(hsISkypeEvents\_ContactFocus),helpcontext(4)] HRESULT ContactsFocused([in] BSTR Username);

01980 [id(22), helpstring(hsISkypeEvents\_GroupVisible),helpcontext(4)] HRESULT GroupVisible([in] IGroup\* pGroup, [in] VARIANT\_BOOL Visible);

01981 [id(23), helpstring(hsISkypeEvents\_GroupExpanded),helpcontext(4)] HRESULT GroupExpanded([in] IGroup\* pGroup, [in] VARIANT\_BOOL Expanded);

01982 [id(24), helpstring(hsISkypeEvents\_GroupUsers),helpcontext(4)] HRESULT GroupUsers([in] IGroup\* pGroup, [in] IUserCollection\* pUsers);

01983 [id(25), helpstring(hsISkypeEvents\_GroupDeleted),helpcontext(4)] HRESULT GroupDeleted([in] LONG GroupId);

01984 [id(26), helpstring(hsISkypeEvents\_UserMood),helpcontext(4)] HRESULT UserMood([in] IUser\* pUser, [in] BSTR MoodText);

01985 [id(27), helpstring(hsISkypeEvents\_SmsMessageStatus),helpcontext(4)] HRESULT SmsMessageStatusChanged([in] ISmsMessage\* pMessage, [in] TSmsMessageStatus Status);

01986 [id(28), helpstring(hsISkypeEvents\_SmsTargetStatus),helpcontext(4)] HRESULT SmsTargetStatusChanged([in] ISmsTarget\* pTarget, [in] TSmsTarg

```
etStatus Status);
01987      [id(29), helpstring(hsISkypeEvents_Voi
ceInputStatusChanged),helpcontext(4)] HRESULT Call
InputStatusChanged([in] ICall* pCall, [in] VARIANT
_BOOL Status);
01988      [id(30), helpstring(hsISkypeEvents_Asy
ncSearchFinished),helpcontext(4)] HRESULT AsyncSea
rchUsersFinished([in] LONG Cookie, [in] IUserColle
ction* pUsers);
01989      [id(31), helpstring(hsISkypeEvents_Cal
lSeenStatusChanged),helpcontext(4)] HRESULT CallSe
enStatusChanged([in] ICall* pCall, VARIANT_BOOL St
atus);
01990      [id(32), helpstring(hsISkypeEvents_Plu
ginEventClicked),helpcontext(4)] HRESULT PluginEve
ntClicked([in] IPluginEvent* pEvent);
01991      [id(33), helpstring(hsISkypeEvents_Plu
ginMenuItemClicked),helpcontext(4)] HRESULT Plugin
MenuItemClicked([in] IPluginMenuItem* pItem, [
in] IUserCollection* pUsers, [in] TPluginContext P
luginContext, [in] BSTR ContextId);
01992      [id(34), helpstring(hsISkypeEvents_Wal
lpaperChanged),helpcontext(4)] HRESULT WallpaperCh
anged([in] BSTR Path);
01993      [id(35), helpstring(hsISkypeEvents_Fil
eTransferStatusChanged),helpcontext(4)] HRESULT Fi
leTransferStatusChanged([in] IFileTransfer* pTrans
fer, [in] TFileTransferStatus Status);
01994      [id(36), helpstring(hsISkypeEvents_Cal
lTransferStatusChanged),helpcontext(4)] HRESULT Ca
llTransferStatusChanged([in] ICall* pCall, [in] TC
allStatus Status);
01995      [id(37), helpstring(hsISkypeEvents_Cha
tMembersChanged),helpcontext(4)] HRESULT ChatMembe
rsChanged([in] IChat* pChat, [in] IUserCollection*
pMembers);
01996      [id(38), helpstring(hsISkypeEvents_Cha
```

```

tMemberRoleChanged),helpcontext(4)] HRESULT ChatMe
mberRoleChanged([in] IChatMember* pMember, [in] TC
hatMemberRole Role);
01997      [id(39), helpstring(""),helpcontext(4)
] HRESULT CallVideoStatusChanged([in] ICall* pCall
, [in] TCallVideoStatus Status);
01998      [id(40), helpstring(""),helpcontext(4)
] HRESULT CallVideoSendStatusChanged([in] ICall* p
Call, [in] TCallVideoSendStatus Status);
01999      [id(41), helpstring(""),helpcontext(4)
] HRESULT CallVideoReceiveStatusChanged([in] ICall
* pCall, [in] TCallVideoSendStatus Status);
02000      [id(42), helpstring(""),helpcontext(4)
] HRESULT SilentModeStatusChanged([in] VARIANT_BOO
L Silent);
02001      [id(43), helpstring(""),helpcontext(4)
] HRESULT UILanguageChanged([in] BSTR Code);
02002      [id(44), helpstring(""),helpcontext(4)
] HRESULT UserAuthorizationRequestReceived([in] IU
ser* pUser);
02003      };
02004 #ifndef DOXYGEN_SHOULD_SKIP_THIS
02005      [
02006          object,
02007          uuid(3F6758D2-1D3C-4A8F-BD99-6FC6B0E2D
C8F),
02008          dual,
02009          nonextensible,
02010          pointer_default(unique),
02011          oleautomation,
02012          helpstring(hsISkypeChat),
02013          helpcontext(48),
02014          hidden
02015      ]
02016      interface ISkypeChat : IDispatch
02017      {
02018          [propget, id(1), helpstring(hsIChat_Ti

```

```
mestamp), helpcontext(48)] HRESULT ChatTimestamp([
in] BSTR ChatName, [out, retval] DATE *pVal);
02019      [propget, id(2), helpstring(hsIChat_Ad
der), helpcontext(48)] HRESULT ChatAdder([in] BSTR
ChatName, [out, retval] IUser **pVal);
02020      [propget, id(3), helpstring(hsIChat_St
atus), helpcontext(48)] HRESULT ChatStatus([in] BS
TR ChatName, [out, retval] TChatStatus *pVal);
02021      [propget, id(4), helpstring(hsIChat_Po
sters), helpcontext(48)] HRESULT ChatPosters([in]
BSTR ChatName, [out, retval] IUserCollection **pVa
l);
02022      [propget, id(5), helpstring(hsIChat_Me
mbers), helpcontext(48)] HRESULT ChatMembers([in]
BSTR ChatName, [out, retval] IUserCollection **pVa
l);
02023      [propget, id(6), helpstring(hsIChat_To
pic), helpcontext(48)] HRESULT ChatTopic([in] BSTR
ChatName, [out, retval] BSTR *pVal);
02024      [propput, id(6), helpstring(hsIChat_To
pic), helpcontext(48)] HRESULT ChatTopic([in] BSTR
ChatName, [in] BSTR newVal);
02025      [propget, id(7), helpstring(hsIChat_Ac
tiveMembers), helpcontext(48)] HRESULT ChatActiveM
embers([in] BSTR ChatName, [out, retval] IUserColl
ection **pVal);
02026      [propget, id(8), helpstring(hsIChat_Fr
riendlyName), helpcontext(48)] HRESULT ChatFriendly
Name([in] BSTR ChatName, [out, retval] BSTR *pVal)
;
02027      [propget, id(9), helpstring(hsIChat_Me
ssages), helpcontext(48)] HRESULT ChatMessages([in
] BSTR ChatName, [out, retval] IChatMessageCollect
ion **pVal);
02028      [id(10), helpstring(hsIChat_OpenWindow
), helpcontext(48)] HRESULT OpenChatWindow([in] BS
TR ChatName);
```

```
02029      [id(11), helpstring(hsIChat_SendMessage), helpcontext(48)] HRESULT SendChatMessage([in] BSTR ChatName, [in] BSTR MessageText, [out, retval] IChatMessage **pVal);
02030      [id(12), helpstring(hsIChat_Leave), helpcontext(48)] HRESULT LeaveChat([in] BSTR ChatName);
02031      [id(13), helpstring(hsIChat_AddMembers), helpcontext(48)] HRESULT AddChatMembers([in] BSTR ChatName, [in] IUserCollection* pMembers);
02032      [propget, id(14), helpstring(hsIChat_RecentMessages), helpcontext(48)] HRESULT ChatRecentMessages([in] BSTR ChatName, [out, retval] IChatMessageCollection **pVal);
02033      [propget, id(15), helpstring(hsIChat_Bookmarked), helpcontext(48)] HRESULT ChatBookmarked([in] BSTR ChatName, [out, retval] VARIANT_BOOL *pVal);
02034      [id(16), helpstring(hsIChat_Bookmark), helpcontext(48)] HRESULT BookmarkChat([in] BSTR ChatName);
02035      [id(17), helpstring(hsIChat_Unbookmark), helpcontext(48)] HRESULT UnbookmarkChat([in] BSTR ChatName);
02036      [propget, id(18), helpstring(hsIChat_TopicXML), helpcontext(48)] HRESULT ChatTopicXML([in] BSTR ChatName, [out, retval] BSTR *pVal);
02037      [propput, id(18), helpstring(hsIChat_TopicXML), helpcontext(48)] HRESULT ChatTopicXML([in] BSTR ChatName, [in] BSTR newVal);
02038      [propget, id(19), helpstring(hsIChat_MemberObjs), helpcontext(48)] HRESULT ChatMemberObjects([in] BSTR ChatName, [out, retval] IChatMemberCollection **pVal);
02039      [propget, id(20), helpstring(hsIChat_Blob), helpcontext(48)] HRESULT ChatBlob([in] BSTR ChatName, [out, retval] BSTR* pVal);
```

```
02040      [propget, id(21), helpstring(hsIChat_Options), helpcontext(48)] HRESULT ChatOptions([in] BSTR ChatName, [out, retval] LONG* pVal);
02041      [propput, id(21), helpstring(hsIChat_Options), helpcontext(48)] HRESULT ChatOptions([in] BSTR ChatName, [in] LONG newVal);
02042      [propget, id(22), helpstring(hsIChat_Passwordhint), helpcontext(48)] HRESULT ChatPasswordHint([in] BSTR ChatName, [out, retval] BSTR* pVal);
02043      [id(23), helpstring(hsIChat_Password), helpcontext(48)] HRESULT SetChatPassword([in] BSTR ChatName, [in] BSTR Password, [in, defaultvalue("")] BSTR Hint);
02044      [propget, id(24), helpstring(hsIChat_Guidelines), helpcontext(48)] HRESULT ChatGuidelines([in] BSTR ChatName, [out, retval] BSTR* pVal);
02045      [propput, id(24), helpstring(hsIChat_Guidelines), helpcontext(48)] HRESULT ChatGuidelines([in] BSTR ChatName, [in] BSTR newVal);
02046      [propget, id(25), helpstring(hsIChat_Description), helpcontext(48)] HRESULT ChatDescription([in] BSTR ChatName, [out, retval] BSTR* pVal);
02047      [propput, id(25), helpstring(hsIChat_Description), helpcontext(48)] HRESULT ChatDescription([in] BSTR ChatName, [in] BSTR newVal);
02048      [propget, id(26), helpstring(hsIChat_DialogPartner), helpcontext(48)] HRESULT ChatDialogPartner([in] BSTR ChatName, [out, retval] BSTR* pVal);
02049      [propget, id(27), helpstring(hsIChat_ActiveTime), helpcontext(48)] HRESULT ChatActivityTimestamp([in] BSTR ChatName, [out, retval] DATE* pVal);
02050      [propget, id(28), helpstring(hsIChat_MyRole), helpcontext(48)] HRESULT ChatMyRole([in] BSTR ChatName, [out, retval] TChatMemberRole* pVal);
;
```



```

02051      [propget, id(29), helpstring(hsIChat_A
pplicants), helpcontext(48)] HRESULT ChatApplicant
s([in] BSTR ChatName, [out, retval] IUserCollection
** pVal);
02052      [id(30), helpstring(hsIChat_Join), hel
pcontext(48)] HRESULT JoinChat([in] BSTR ChatName)
;
02053      [id(31), helpstring(hsIChat_Kick), hel
pcontext(48)] HRESULT KickChatMember([in] BSTR Cha
tName, [in] BSTR Handle);
02054      [id(32), helpstring(hsIChat_Kickban),
helpcontext(48)] HRESULT KickBanChatMember([in] BS
TR ChatName, [in] BSTR Handle);
02055      [id(33), helpstring(hsIChat_Disband),
helpcontext(48)] HRESULT DisbandChat([in] BSTR Cha
tName);
02056      [id(34), helpstring(hsIChat_EnterPassw
d), helpcontext(48)] HRESULT EnterChatPassword([in
] BSTR ChatName, [in] BSTR Password);
02057      [id(35), helpstring(hsIChat_ClearRecen
tMsg), helpcontext(48)] HRESULT ClearChatRecentMes
sages([in] BSTR ChatName);
02058      [id(36), helpstring(hsIChat_AcceptAdd)
, helpcontext(48)] HRESULT AcceptSharedGroupAdd([i
n] BSTR ChatName);
02059      [propget, id(37), helpstring(hsIChat_A
lertString), helpcontext(48)] HRESULT ChatAlertStr
ing([in] BSTR ChatName, [in] BSTR newVal);
02060      [propget, id(38), helpstring(hsIChat_T
ype), helpcontext(48)] HRESULT ChatType([in] BSTR
ChatName, [out, retval] TChatType* pVal);
02061      [propget, id(39), helpstring(hsIChat_M
yStatus), helpcontext(48)] HRESULT ChatMyStatus([i
n] BSTR ChatName, [out, retval] TChatMyStatus* pVa
l);
02062      };
02063      [

```

```

02064         object,
02065         uuid(4541C958-6341-4AC1-A6B0-722C89049
02066         928),
02067         dual,
02068         nonextensible,
02069         pointer_default(unique),
02070         oleautomation,
02071         helpstring(hsISkypeGroup),
02072         helpcontext(9),
02073         hidden
02074     ]
02075     interface ISkypeGroup : IDispatch
02076     {
02077         [propget, id(1), helpstring(hsIGroup_T
02078         ype), helpcontext(9)] HRESULT GroupType([in] BSTR
02079         GroupId, [out, retval] TGroupType *pVal);
02080         [propget, id(2), helpstring(hsIGroup_C
02081         ustomId), helpcontext(9)] HRESULT GroupCustomId([i
02082         n] BSTR GroupId, [out, retval] BSTR *pVal);
02083         [propget, id(3), helpstring(hsIGroup_D
02084         isplayName), helpcontext(9)] HRESULT GroupDisplayN
02085         ame([in] BSTR GroupId, [out, retval] BSTR *pVal);
02086         [propput, id(3), helpstring(hsIGroup_D
02087         isplayName), helpcontext(9)] HRESULT GroupDisplayN
02088         ame([in] BSTR GroupId, [in] BSTR newVal);
02089         [propget, id(4), helpstring(hsIGroup_U
02090         sers), helpcontext(9)] HRESULT GroupUsers([in] BST
02091         R GroupId, [out, retval] IUserCollection* *pVal);
02092         [propget, id(5), helpstring(hsIGroup_I
02093         sVisible), helpcontext(9)] HRESULT GroupIsVisible(
02094         [in] BSTR GroupId, [out, retval] VARIANT_BOOL *pVa
02095         l);
02096         [propget, id(6), helpstring(hsIGroup_I
02097         sExpanded), helpcontext(9)] HRESULT GroupIsExpande
02098         d([in] BSTR GroupId, [out, retval] VARIANT_BOOL *p
02099         Val);
02100         [id(7), helpstring(hsIGroup_AddUser),

```

```

helpcontext(9)] HRESULT AddGroupUser([in] BSTR GroupId, [in] BSTR Username);
02084         [id(8), helpstring(hsIGroup_RemoveUser), helpcontext(9)] HRESULT RemoveGroupUser([in] BSTR GroupId, [in] BSTR Username);
02085         [id(9), helpstring(hsIGroup_Share), helpcontext(9)] HRESULT ShareGroup([in] BSTR GroupId, [in] BSTR MessageText);
02086         [id(10), helpstring(hsIGroup_Accept), helpcontext(9)] HRESULT AcceptSharedGroup([in] BSTR GroupId);
02087         [id(11), helpstring(hsIGroup_Decline)] HRESULT DeclineSharedGroup([in] BSTR GroupId);
02088     };
02089     [
02090         object,
02091         uuid(B8731A95-8812-48D7-A1D1-438C108F8C2C),
02092         dual,
02093         nonextensible,
02094         pointer_default(unique),
02095         oleautomation,
02096         helpstring(hsISkypeSettings),
02097         helpcontext(2),
02098         hidden
02099     ]
02100     interface ISkypeSettings : IDispatch
02101     {
02102         [propget, id(1), helpstring(hsISettings_AudioIn), helpcontext(2)] HRESULT AudioIn([out, retval] BSTR *pVal);
02103         [propput, id(1), helpstring(hsISettings_AudioIn), helpcontext(2)] HRESULT AudioIn([in] BSTR newVal);
02104         [propget, id(2), helpstring(hsISettings_AudioOut), helpcontext(2)] HRESULT AudioOut([out, retval] BSTR *pVal);

```

```
02105      [propput, id(2), helpstring(hsISetting
s_AudioOut),helpcontext(2)] HRESULT AudioOut([in]
BSTR newVal);
02106      [propget, id(3), helpstring(hsISetting
s_AutoAway),helpcontext(2)] HRESULT AutoAway([out,
  retval] VARIANT_BOOL *pVal);
02107      [propput, id(3), helpstring(hsISetting
s_AutoAway),helpcontext(2)] HRESULT AutoAway([in]
VARIANT_BOOL newVal);
02108      [propget, id(4), helpstring(hsISetting
s_Ringer),helpcontext(2)] HRESULT Ringer([out, ret
val] BSTR *pVal);
02109      [propput, id(4), helpstring(hsISetting
s_Ringer),helpcontext(2)] HRESULT Ringer([in] BSTR
  newVal);
02110      [propput, id(5), helpstring(hsISetting
s_Avatar),helpcontext(2)] HRESULT Avatar([in, defa
ultvalue("1")] BSTR Id, [in] BSTR newVal);
02111      [propget, id(6), helpstring(hsISetting
s_RingToneStatus),helpcontext(2)] HRESULT RingTone
Status([in, defaultvalue("1")] BSTR Id,[out, retva
l] VARIANT_BOOL *pVal);
02112      [propput, id(7), helpstring(hsISetting
s_RingTone),helpcontext(2)] HRESULT RingTone([in,
defaultvalue("1")] BSTR Id, [in] BSTR newVal);
02113      [propget, id(8), helpstring(hsISetting
s_VideoIn),helpcontext(2)] HRESULT VideoIn([out, r
etval] BSTR *pVal);
02114      [propput, id(8), helpstring(hsISetting
s_VideoIn),helpcontext(2)] HRESULT VideoIn([in] BS
TR newVal);
02115      [propget, id(9), helpstring(hsISetting
s_PCSpeaker),helpcontext(2)] HRESULT PCSpeaker([ou
t, retval] VARIANT_BOOL *pVal);
02116      [propput, id(9), helpstring(hsISetting
s_PCSpeaker),helpcontext(2)] HRESULT PCSpeaker([in
] VARIANT_BOOL newVal);
```

```
02117      [propget, id(10), helpstring(hsISettin
gs_AGC),helpcontext(2)] HRESULT AGC([out, retval]
VARIANT_BOOL *pVal);
02118      [propput, id(10), helpstring(hsISettin
gs_AGC),helpcontext(2)] HRESULT AGC([in] VARIANT_B
OOL newVal);
02119      [propget, id(11), helpstring(hsISettin
gs_AEC),helpcontext(2)] HRESULT AEC([out, retval]
VARIANT_BOOL *pVal);
02120      [propput, id(11), helpstring(hsISettin
gs_AEC),helpcontext(2)] HRESULT AEC([in] VARIANT_B
OOL newVal);
02121      [propget, id(12), helpstring(hsISettin
gs_Language),helpcontext(2)] HRESULT Language([out
, retval] BSTR *pVal);
02122      [propput, id(12), helpstring(hsISettin
gs_Language),helpcontext(2)] HRESULT Language([in]
BSTR newVal);
02123      [propput, id(6), helpstring(hsISetting
s_RingToneStatus),helpcontext(2)] HRESULT RingTone
Status([in, defaultvalue("1")] BSTR Id, [in] VARIA
NT_BOOL newVal);
02124      [id(13), helpstring(hsISettings_SaveAv
atar), helpcontext(2)] HRESULT SaveAvatarToFile([i
n] BSTR Filename, [in, defaultvalue("1")] BSTR Ava
tarId);
02125      [id(14), helpstring(hsISettings_LoadAv
atar), helpcontext(2)] HRESULT LoadAvatarFromFile(
[in] BSTR Filename, [in, defaultvalue("1")] BSTR A
vatarId);
02126      [id(15), helpstring(hsISettings_IdleTi
mer),helpcontext(2)] HRESULT ResetIdleTimer(void);
02127      };
02128      [
02129      object,
02130      uuid(B6BED647-709E-4B4B-B1E3-B01059999
66F),
```

```

02131         dual,
02132         nonextensible,
02133         pointer_default(unique),
02134         oleautomation,
02135         helpstring(hsISkypeClient),
02136         helpcontext(36),
02137         hidden
02138     ]
02139     interface ISkypeClient : IDispatch
02140     {
02141         [id(1), helpstring(hsIClient_Start), h
elpcontext(36)] HRESULT Start([in, defaultvalue(0)
] VARIANT_BOOL Minimized, [in, defaultvalue(0)] VA
RIANT_BOOL Nosplash);
02142         [id(2), helpstring(hsIClient_Minimize)
, helpcontext(36)] HRESULT Minimize();
02143         [id(3), helpstring(hsIClient_Shutdown)
, helpcontext(36)] HRESULT Shutdown();
02144         [propget, id(4), helpstring(hsIClient_
Running),helpcontext(36)] HRESULT IsRunning([out,
retval] VARIANT_BOOL *pVal);
02145         [id(5), helpstring(hsIClient_OpenProfi
le),helpcontext(36)] HRESULT OpenProfileDialog();
02146         [id(6), helpstring(hsIClient_OpenUseri
nfo),helpcontext(36)] HRESULT OpenUserInfoDialog([
in] BSTR Username);
02147         [id(7), helpstring(hsIClient_OpenConfe
rence),helpcontext(36)] HRESULT OpenConferenceDial
og();
02148         [id(8), helpstring(hsIClient_OpenSearc
h),helpcontext(36)] HRESULT OpenSearchDialog();
02149         [id(9), helpstring(hsIClient_OpenOptio
ns),helpcontext(36)] HRESULT OpenOptionsDialog([in
] BSTR Page);
02150         [id(10), helpstring(hsIClient_OpenCall
Hist),helpcontext(36)] HRESULT OpenCallHistoryTab(
);

```

```
02151      [id(11), helpstring(hsIClient_OpenCont
acts),helpcontext(36)] HRESULT OpenContactsTab();
02152      [id(12), helpstring(hsIClient_OpenDial
pad),helpcontext(36)] HRESULT OpenDialpadTab();
02153      [id(13), helpstring(hsIClient_OpenSend
Cont),helpcontext(36)] HRESULT OpenSendContactsDia
log([in, defaultvalue("")] BSTR Username);
02154      [id(14), helpstring(hsIClient_OpenBloc
kUser),helpcontext(36)] HRESULT OpenBlockedUsersDi
alog();
02155      [id(15), helpstring(hsIClient_OpenImpo
rtContact),helpcontext(36)] HRESULT OpenImportCont
actsWizard();
02156      [id(16), helpstring(hsIClient_OpenGets
tart),helpcontext(36)] HRESULT OpenGettingStartedW
izard();
02157      [id(17), helpstring(hsIClient_OpenAuth
oriz),helpcontext(36)] HRESULT OpenAuthorizationDi
alog([in] BSTR Username);
02158      [id(18), helpstring(hsIClient_OpenDial
og),helpcontext(36)] HRESULT OpenDialog([in] BSTR
Name, [in, defaultvalue("")] BSTR Param1, [in, def
aultvalue("")] BSTR Param2);
02159      [id(19), helpstring(hsIClient_OpenVide
oTest),helpcontext(36)] HRESULT OpenVideoTestDialo
g();
02160      [id(20), helpstring(hsIClient_OpenAddC
ontact),helpcontext(36)] HRESULT OpenAddContactDia
log([in, defaultvalue("")] BSTR Username);
02161      [id(21), helpstring(hsIClient_OpenMess
age),helpcontext(36)] HRESULT OpenMessageDialog([i
n] BSTR Username, [in, defaultvalue("")] BSTR Text
);
02162      [id(22), helpstring(hsIClient_OpenFile
Trans),helpcontext(36)] HRESULT OpenFileTransferDi
alog([in] BSTR Username, [in] BSTR Folder);
02163      [id(23), helpstring(hsIClient_Focus),h
```

```

elpcontext(36)] HRESULT Focus();
02164      [id(24), helpstring(hsIClient_BtnPress
ed),helpcontext(36)] HRESULT ButtonPressed([in] BS
TR Key);
02165      [id(25), helpstring(hsIClient_BtnRelea
sed),helpcontext(36)] HRESULT ButtonReleased([in]
BSTR Key);
02166      [id(26), helpstring(hsIClient_OpenSms)
, helpcontext(36)] HRESULT OpenSmsDialog([in] BSTR
SmsId);
02167      [propget, id(27), helpstring(hsIClient
_Wallpaper), helpcontext(36)] HRESULT Wallpaper([o
ut, retval] BSTR* pVal);
02168      [propput, id(27), helpstring(hsIClient
_Wallpaper), helpcontext(36)] HRESULT Wallpaper([i
n] BSTR newVal);
02169      [id(28), helpstring(hsIClient_OpenLive
),helpcontext(36)] HRESULT OpenLiveTab();
02170      };
02171      [
02172          object,
02173          uuid(B77A3D1F-6847-402f-BB82-A05643790
88E),
02174          dual,
02175          nonextensible,
02176          pointer_default(unique),
02177          oleautomation,
02178          helpstring(""),
02179          helpcontext(0),
02180          hidden
02181      ]
02182      interface ISkypePlugin : IDispatch
02183      {
02184          [id(1), helpstring(""), helpcontext(0)
] HRESULT CreatePluginEvent([in] BSTR EventId, [in
] BSTR Caption, [in] BSTR Hint, [out,retval] IPlug
inEvent** pEvent);

```



```

02185         [id(2), helpstring(""), helpcontext(0)
] HRESULT DeletePluginEvent([in] BSTR EventId);
02186         [id(3), helpstring(""), helpcontext(0)
] HRESULT CreatePluginMenuItem([in] BSTR MenuItemI
d, [in] TPluginContext PluginContext, [in] BSTR Ca
ptionText, [in, defaultvalue("")] BSTR HintText, [i
n, defaultvalue("")] BSTR IconPath, [in, defaultva
lue(-1)] VARIANT_BOOL Enabled, [in, defaultvalue(p
luginContactTypeAll)] TPluginContactType ContactTy
pe, [in, defaultvalue(0)] VARIANT_BOOL MultipleCon
tacts, [out, retval] IPluginMenuItem** pMenuItem);
02187         [id(4), helpstring(""), helpcontext(0)
] HRESULT DeletePluginMenuItem([in] BSTR MenuItemI
d);
02188         [propput, id(5), helpstring(""), helpc
ontext(0)] HRESULT PluginMenuItemCaption([in] BSTR
MenuItemId, [in] BSTR newVal);
02189         [propput, id(6), helpstring(""), helpc
ontext(0)] HRESULT PluginMenuItemHint([in] BSTR Me
nuItemId, [in] BSTR newVal);
02190         [id(7), helpstring(""), helpcontext(0)
] HRESULT EnablePluginMenuItem([in] BSTR MenuItemI
d, [in] VARIANT_BOOL Enabled);
02191     };
02192 #endif /*DOXYGEN_SHOULD_SKIP_THIS*/
02193
02194     interface ICallChannelManager;
02195     [
02196         object,
02197         uuid(B76C04C9-0EB1-4748-B0C0-D01DEB595
906),
02198         dual,
02199         nonextensible,
02200         pointer_default(unique),
02201         oleautomation,
02202         helpstring(hsICallChannel),
02203         helpcontext(67)

```

```

02204     ]
02205     interface ICallChannel : IDispatch
02206     {
02207         [propget, id(2), helpstring(hsICallChannel_Type), helpcontext(67)] HRESULT Type([out, retval] TCallChannelType *pVal);
02208         [propput, id(2), helpstring("hsICallChannel_Type"), helpcontext(67)] HRESULT Type([in] TCallChannelType newVal);
02209         [id(3), helpstring(hsICallChannel_Send), helpcontext(67)] HRESULT SendMessage(BSTR Text);
02210         [propget, id(4), helpstring(hsICallChannel_Stream), helpcontext(67)] HRESULT Stream([out, retval] IApplicationStream* *pVal);
02211         [propget, id(5), helpstring(hsICallChannel_Manager), helpcontext(67)] HRESULT Manager([out, retval] ICallChannelManager* *pVal);
02212         [propget, id(6), helpstring(hsICallChannel_Call), helpcontext(67)] HRESULT Call([out, retval] ICall* *pVal);
02213     };
02214     [
02215         object,
02216         uuid(8CB09087-E389-4C6D-A6A2-7D4BCA8545D9),
02217         dual,
02218         nonextensible,
02219         pointer_default(unique),
02220         oleautomation,
02221         helpstring(hsICallChannelColl),
02222         helpcontext(68)
02223     ]
02224     interface ICallChannelCollection : IDispatch
02225     {
02226         [propget, id(1), helpstring(hsICallChannelColl), helpcontext(68)] HRESULT Collection([out, retval] ICallChannelCollection* *pVal);

```

```

nelColl_Count),helpcontext(68)] HRESULT Count([out, retval] LONG* pVal);
02227      [id(2), helpstring(hsICallChannelColl_Add), helpcontext(68)] HRESULT Add([in] ICallChannel* pChannel);
02228      [id(3), helpstring(hsICallChannelColl_Remove), helpcontext(68)] HRESULT Remove([in] long Index);
02229      [id(4), helpstring(hsICallChannelColl_RemoveAll), helpcontext(68)] HRESULT RemoveAll(void);
02230      [propget, id(DISPID_VALUE), helpstring(hsICallChannelColl_Item), helpcontext(68)] HRESULT Item([in] LONG Index, [out, retval] ICallChannel** pVal);
02231      [propget, id(DISPID_NEWENUM), helpstring(hsICallChannelColl_NewEnum), helpcontext(68)] HRESULT _NewEnum([out, retval] IUnknown** pVal);
02232      };
02233      [
02234          object,
02235          uuid(30C27C77-525F-4A3F-B9DD-C5A77C67250D),
02236          dual,
02237          nonextensible,
02238          pointer_default(unique),
02239          oleautomation,
02240          helpstring(hsICallChannelMgr),
02241          helpcontext(69)
02242      ]
02243      interface ICallChannelManager : IDispatch
02244      {
02245          [id(1), helpstring(hsICallChannelMgr_Connect), helpcontext(69)] HRESULT Connect([in] ISkype* pSkype);
02246          [id(2), helpstring(hsICallChannelMgr_Disconnect), helpcontext(69)] HRESULT Disconnect();

```

```

02247         [propget, id(3), helpstring(hsICallCha
nnelMgr_Channels), helpcontext(69)] HRESULT Channe
ls([out, retval] ICallChannelCollection** pVal);
02248         [propget, id(6), helpstring(hsICallCha
nnelMgr_ChannelType), helpcontext(69)] HRESULT Cha
nnelType([out, retval] TCallChannelType *pVal);
02249         [propput, id(6), helpstring(hsICallCha
nnelMgr_ChannelType), helpcontext(69)] HRESULT Cha
nnelType([in] TCallChannelType newVal);
02250         [propget, id(7), helpstring(hsICallCha
nnelMgr_Name), helpcontext(69)] HRESULT Name([out,
        retval] BSTR *pVal);
02251         [propput, id(7), helpstring(hsICallCha
nnelMgr_Name), helpcontext(69)] HRESULT Name([in]
BSTR newVal);
02252         [id(8), helpstring(hsICallChannelMgr_C
reate), helpcontext(69)] HRESULT CreateApplication
([in, defaultvalue("")] BSTR ApplicationName);
02253         [propget, id(9), helpstring(hsICallCha
nnelMgr_Created), helpcontext(69)] HRESULT Created
([out, retval] VARIANT_BOOL *pVal);
02254     };
02255     [
02256         object,
02257         uuid(0EF6FD5D-607D-4EA3-9C90-683D73449
E9D),
02258         dual,
02259         nonextensible,
02260         pointer_default(unique),
02261         oleautomation,
02262         helpstring(hsICallChannelMsg),
02263         helpcontext(70)
02264     ]
02265     interface ICallChannelMessage : IDispatch
02266     {
02267         [propget, id(1), helpstring(hsICallCha
nnelMsg_Text), helpcontext(70)] HRESULT Text([out,

```

```

    retval] BSTR *pVal);
02268     [propput, id(1), helpstring(hsICallChannelMsg_Text), helpcontext(70)] HRESULT Text([in]
BSTR newVal);
02269     };
02270     [
02271         uuid(497ABB45-20AE-49D1-A39D-CBE84A31B
20C),
02272         helpstring(ICallChannelManagerEvents),
        helpcontext(71)
02273     ]
02274     dispinterface _ICallChannelManagerEvents
02275     {
02276         properties:
02277         methods:
02278         [id(1), helpstring(ICallChannelManager
Events_Channels),helpcontext(71)] HRESULT Channels
([in] ICallChannelManager* pManager, [in] ICallChannelCollection* pChannels);
02279         [id(2), helpstring(ICallChannelManager
Events_Message),helpcontext(71)] HRESULT Message([
in] ICallChannelManager* pManager, [in] ICallChannel* pChannel, [in] ICallChannelMessage* pMessage);
02280         [id(3), helpstring(ICallChannelManager
Events_Created),helpcontext(71)] HRESULT Created()
;
02281     };
02282     [
02283         uuid(03282B5D-B38F-469D-849A-09B0A7F4881B
),
02284         version(1.0),
02285         helpstring(hsSkypeLib_Version), helpcontex
xt(1),
02286         helpfile(hsSkypeLib_HelpFile)
02287     ]
02288     library SKYPE4COMLib
02289     {

```

```
02290     importlib("stdole32.tlb");
02291     importlib("stdole2.tlb");
02292     interface IApplication;
02293     interface IApplicationStream;
02294     interface IApplicationStreamCollection;
02295     interface ICall;
02296     interface ICallChannel;
02297     interface ICallChannelManager;
02298     interface ICallChannelMessage;
02299     interface ICallCollection;
02300     interface ICallIoDevice;
02301     interface ICallIoDeviceCollection;
02302     interface IChat;
02303     interface IChatCollection;
02304     interface IChatMessage;
02305     interface IChatMessageCollection;
02306     interface IChatMember;
02307     interface IChatMemberCollection;
02308     interface IClient;
02309     interface ICommand;
02310     interface IConference;
02311     interface IConferenceCollection;
02312     interface IConversion;
02313     interface IFileTransfer;
02314     interface IFileTransferCollection;
02315     interface IParticipant;
02316     interface IParticipantCollection;
02317     interface ISettings;
02318     interface ISkype;
02319     interface ISkypeApplication;
02320     interface ISkypeCall;
02321     interface ISkypeChat;
02322     interface ISkypeChatMessage;
02323     interface ISkypeChatMember;
02324     interface ISkypeConference;
02325     interface ISkypeDevice;
02326     interface ISkypeFileTransfer;
```

```

02327     interface ISkypeSms;
02328     interface ISkypeUser;
02329     interface ISkypeVoicemail;
02330     interface IUser;
02331     interface IUserCollection;
02332     interface IVoicemail;
02333     interface IVoicemailCollection;
02334     dispinterface _ISkypeEvents;
02335     dispinterface _ICallChannelManagerEvents;
02336
02337     [
02338         uuid(830690FC-BF2F-47A6-AC2D-330BCB402
02339         664),
02340         helpstring(hsSkype), helpcontext(18)
02341     ]
02342     coclass Skype
02343     {
02344         [default] interface ISkype;
02345         [default, source] dispinterface _ISkyp
02346         eEvents;
02347         interface ISkypeApplication;
02348         interface ISkypeCall;
02349         interface ISkypeChat;
02350         interface ISkypeChatMessage;
02351         interface ISkypeConference;
02352         interface ISkypeFileTransfer;
02353         interface ISkypePlugin;
02354         interface ISkypeSms;
02355         interface ISkypeUser;
02356         interface ISkypeVoicemail;
02357     };
02358     [
02359         uuid(3E01D8E0-A72B-4C9F-99BD-8A6E7B97A
02360         48D),
02361         helpstring(hsUser), helpcontext(19)
02362     ]
02363     coclass User

```

```
02361     {
02362         [default] interface IUser;
02363     };
02364     [
02365         uuid(7B030003-037D-490D-9169-A4F391B3D
02366         831),
02367         helpstring(hsUserCollection), helpcont
02368         ext(20)
02369     ]
02370     coclass UserCollection
02371     {
02372         [default] interface IUserCollection;
02373     };
02374     [
02375         uuid(2EEAB6D0-491E-4962-BBA1-FF1CCA6D4
02376         DD0),
02377         helpstring(hsConversion), helpcontext(
02378         21)
02379     ]
02380     coclass Conversion
02381     {
02382         [default] interface IConversion;
02383     };
02384     [
02385         uuid(D26B1D42-9C42-4E7B-BB73-86384C4B4
02386         345),
02387         helpstring(hsCall), helpcontext(22)
02388     ]
02389     coclass Call
02390     {
02391         [default] interface ICall;
02392     };
02393     [
02394         uuid(F3188CF3-EF22-4C5B-92CB-605964761
02395         C3B),
02396         helpstring(hsCallCollection), helpcont
02397         ext(23)
```



```
02391     ]
02392     coclass CallCollection
02393     {
02394         [default] interface ICollection;
02395     };
02396     [
02397         uuid(7ACDC5B4-76A1-4BDF-918D-6962FCABB
AD3),
02398         helpstring(hsChat), helpcontext(34)
02399     ]
02400     coclass Chat
02401     {
02402         [default] interface IChat;
02403     };
02404     [
02405         uuid(15B6FEE5-5FB3-4071-AC1F-7AEDC0E2A
6BB),
02406         helpstring(hsChatColl), helpcontext(35
)
02407     ]
02408     coclass ChatCollection
02409     {
02410         [default] interface ICollection;
02411     };
02412     [
02413         uuid(52071016-E648-4D3B-B57E-2B46CC993
CE0),
02414         helpstring(hsParticipant), helpcontext
(41)
02415     ]
02416     coclass Participant
02417     {
02418         [default] interface IParticipant;
02419     };
02420     [
02421         uuid(E1BC9147-C3E3-4E8A-8304-5E6B5C1C0
774),
```

```
02422         helpstring(hsParticipantColl), helpcon
text(42)
02423     ]
02424     coclass ParticipantCollection
02425     {
02426         [default] interface IParticipantCollec
tion;
02427     };
02428     [
02429         uuid(A983C9EC-D73E-4364-B89B-ACD1E4056
74F),
02430         helpstring(hsConference), helpcontext(
43)
02431     ]
02432     coclass Conference
02433     {
02434         [default] interface IConference;
02435     };
02436     [
02437         uuid(3506CDB7-8BC6-40C0-B108-CEA0B9480
130),
02438         helpstring(hsConferenceColl), helpcont
ext(44)
02439     ]
02440     coclass ConferenceCollection
02441     {
02442         [default] interface IConferenceCollect
ion;
02443     };
02444     [
02445         uuid(DD0E8ED5-1494-4B87-A35C-39F6ED4B1
153),
02446         helpstring(hsVoicemail), helpcontext(5
3)
02447     ]
02448     coclass Voicemail
02449     {
```

```
02450     [default] interface IVoicemail;
02451 };
02452 [
02453     uuid(A7DF2611-D752-4C9F-A90A-B56F18485
02454     EE9),
02455     helpstring(hsVoicemailColl), helpconte
02456 xt(54)
02457 ]
02458 coclass VoicemailCollection
02459 {
02460     [default] interface IVoicemailCollecti
02461 on;
02462 };
02463 [
02464     uuid(29DCD339-D184-469B-8BFB-199A2CCF0
02465     14E),
02466     helpstring(hsApplication), helpcontext
02467 (29)
02468 ]
02469 coclass Application
02470 {
02471     [default] interface IApplication;
02472 };
02473 [
02474     uuid(6FA10A39-4760-4C94-A210-239884861
02475     8EC),
02476     helpstring(hsApplicationStream), helpc
02477 ontent(55)
02478 ]
02479 coclass ApplicationStream
02480 {
02481     [default] interface IApplicationStream
02482 ;
02483 };
02484 [
02485     uuid(4B42750B-57A1-47E7-B340-8EAE0E312
02486     6A4),
```

```
02478         helpstring(hsApplicationStreamColl), h
elpcontext(57)
02479     ]
02480     coclass ApplicationStreamCollection
02481     {
02482         [default] interface IApplicationStream
Collection;
02483     };
02484     [
02485         uuid(9017071A-2E34-4C3A-9BBB-688CBB5A9
FF2),
02486         helpstring(hsChatMessage), helpcontext
(24)
02487     ]
02488     coclass ChatMessage
02489     {
02490         [default] interface IChatMessage;
02491     };
02492     [
02493         uuid(10DD084E-A5AE-456F-A3BE-DA67EBE6B
090),
02494         helpstring(hsChatMessageColl), helpcon
text(25)
02495     ]
02496     coclass ChatMessageCollection
02497     {
02498         [default] interface IChatMessageCollec
tion;
02499     };
02500     [
02501         uuid(452CCB69-6A95-4370-9E5A-B3EFB06A7
651),
02502         helpstring(hsProfile), helpcontext(61)
02503     ]
02504     coclass Profile
02505     {
02506         [default] interface IProfile;
```

```
02507     };
02508     [
02509         uuid(222C0F35-3D78-4570-9F6D-BAEE289D0
02510         304),
02511         helpstring(hsGroup), helpcontext(27)
02512     ]
02513 coclass Group
02514 {
02515     [default] interface IGroup;
02516 };
02517 [
02518     uuid(A8109DB9-88E0-42FE-98EA-8A12BE539
02519     4C6),
02520     helpstring(hsGroupCollection), helpcon
02521     text(28)
02522 ]
02523 coclass GroupCollection
02524 {
02525     [default] interface IGroupCollection;
02526 };
02527 [
02528     uuid(B0FE88F0-C92F-46D6-878F-31599BEA9
02529     44C),
02530     helpstring(hsSettings), helpcontext(59
02531     )
02532 ]
02533 coclass Settings
02534 {
02535     [default] interface ISettings;
02536 };
02537 [
02538     uuid(B09AC3FF-0D5D-41C6-A34E-7C3F58A31
02539     27C),
02540     helpstring(hsClient), helpcontext(62)
02541 ]
02542 coclass Client
02543 {
```

```
02538     [default] interface IClient;
02539 };
02540 [
02541     uuid(2DBCDA9F-1248-400B-A382-A56D71BF7
02542     B15),
02543     helpstring(hsCommand), helpcontext(26)
02544 ]
02545 coclass Command
02546 {
02547     [default] interface ICommand;
02548 };
02549 [
02550     uuid(89DD2F9D-C325-48BF-A615-96BD039BB
02551     C83),
02552     helpstring(hsCallChannel), helpcontext
02553     (63)
02554 ]
02555 coclass CallChannel
02556 {
02557     [default] interface ICallChannel;
02558 };
02559 [
02560     uuid(42FE718B-A148-41D6-885B-01A0AFAE8
02561     723),
02562     helpstring(hsCallChannelColl), helpcon
02563     text(64)
02564 ]
02565 coclass CallChannelCollection
02566 {
02567     [default] interface ICallChannelCollec
02568     tion;
02569 };
02570 [
02571     uuid(5E541E71-A474-4EAD-8FCB-24D400D02
02572     3B7),
02573     helpstring(hsCallChannelMgr), helpcont
02574     ext(63)
```

```

02567     ]
02568     coclass CallChannelManager
02569     {
02570         [default] interface ICallChannelManage
r;
02571         [default, source] dispinterface _ICall
ChannelManagerEvents;
02572     };
02573     [
02574         uuid(3D3E7C1B-79A7-4CC7-8925-41FA813E9
913),
02575         helpstring(hsCallChannelMsg), helpcont
ext(66)
02576     ]
02577     coclass CallChannelMessage
02578     {
02579         [default] interface ICallChannelMessag
e;
02580     };
02581     #ifndef DOXYGEN_SHOULD_SKIP_THIS
02582     [
02583         uuid(FFC8B962-9B40-4DFF-9458-1830C7DD7
F5D),
02584         helpstring("IEProtocolHandler Class"),
        hidden
02585     ]
02586     coclass IEProtocolHandler
02587     {
02588         [default] interface IUnknown;
02589     };
02590     #endif /*DOXYGEN_SHOULD_SKIP_THIS*/
02591     [
02592         uuid(F278D870-7AF7-4957-96EE-E6AC72D0B
109),
02593         helpstring(hsSmsMessage), helpcontext(
79)
02594     ]

```

```
02595     coclass SmsMessage
02596     {
02597         [default] interface ISmsMessage;
02598     };
02599     [
02600         uuid(1BCA4635-F1FC-44C8-B829-48229AEB3
02601         2E3),
02602         helpstring(hsSmsMessageCollection), help
02603         lpcontext(80)
02604     ]
02605     coclass SmsMessageCollection
02606     {
02607         [default] interface ISmsMessageCollect
02608         ion;
02609     };
02610     [
02611         uuid(5792FC7D-5E1D-4F1A-BD4F-A7A50F92B
02612         C6E),
02613         helpstring(hsSmsChunk), helpcontext(83
02614         )
02615     ]
02616     coclass SmsChunk
02617     {
02618         [default] interface ISmsChunk;
02619     };
02620     [
02621         uuid(CC461FC3-C9BE-41FB-8E47-E0115CBC0
02622         1CC),
02623         helpstring(hsSmsChunkCollection), help
02624         context(84)
02625     ]
02626     coclass SmsChunkCollection
02627     {
02628         [default] interface ISmsChunkCollection
02629         ;
02630     };
02631     [
```



```
02624         uuid(61F8FAF0-82D0-407C-AE97-31441483A
E40),
02625         helpstring(hsSmsTarget), helpcontext(8
1)
02626     ]
02627     coclass SmsTarget
02628     {
02629         [default] interface ISmsTarget;
02630     };
02631     [
02632         uuid(6AC51E9C-7947-4B46-A978-0AD601C4E
FC9),
02633         helpstring(hsSmsTargetCollection), hel
pcontext(82)
02634     ]
02635     coclass SmsTargetCollection
02636     {
02637         [default] interface ISmsTargetCollecti
on;
02638     };
02639     [
02640         uuid(3F06DCD2-3A04-463D-A08B-1FFED02C4
D4C),
02641         helpstring(hsPluginMenuItem), helpcont
ext(91)
02642     ]
02643     coclass PluginMenuItem
02644     {
02645         [default] interface IPluginMenuItem;
02646     };
02647     [
02648         uuid(9D073235-D787-497D-8D1F-929559F1C
621),
02649         helpstring(hsPluginEvent), helpcontext
(92)
02650     ]
02651     coclass PluginEvent
```

```
02652     {  
02653     [default] interface IPluginEvent;  
02654     };  
02655 };  
02656
```

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## **\_ICallChannelManagerEvents Member List**

---

This is the complete list of members for **\_ICallChannelManagerEvents**, including all inherited members.

<b>Channels</b> ([in] ICallChannelManager *pManager,[in] ICallChannelCollection *pChannels)	<b>_ICallChannelManagerEvents</b>
<b>Created</b> ()	<b>_ICallChannelManagerEvents</b>
<b>Message</b> ([in] ICallChannelManager *pManager,[in] ICallChannel *pChannel,[in] ICallChannelMessage *pMessage)	<b>_ICallChannelManagerEvents</b>

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## **\_ISkypeEvents Member List**

This is the complete list of members for **\_ISkypeEvents**, including all inherited members.

<b>ApplicationConnecting</b> ([in] IApplication *pApp,[in] IUserCollection *pUsers)	<b>_ISkypeEvents</b>
<b>ApplicationDatagram</b> ([in] IApplication *pApp,[in] IApplicationStream *pStream,[in] BSTR Text)	<b>_ISkypeEvents</b>
<b>ApplicationReceiving</b> ([in] IApplication *pApp,[in] IApplicationStreamCollection *pStreams)	<b>_ISkypeEvents</b>
<b>ApplicationSending</b> ([in] IApplication *pApp,[in] IApplicationStreamCollection *pStreams)	<b>_ISkypeEvents</b>
<b>ApplicationStreams</b> ([in] IApplication *pApp,[in] IApplicationStreamCollection *pStreams)	<b>_ISkypeEvents</b>
<b>AsyncSearchUsersFinished</b> ([in] LONG Cookie,[in] IUserCollection *pUsers)	<b>_ISkypeEvents</b>
<b>AttachmentStatus</b> ([in] TAttachmentStatus Status)	<b>_ISkypeEvents</b>
<b>AutoAway</b> ([in] VARIANT_BOOL Automatic)	<b>_ISkypeEvents</b>
<b>CallDtmfReceived</b> ([in] ICall *pCall,[in] BSTR Code)	<b>_ISkypeEvents</b>
<b>CallHistory</b> ()	<b>_ISkypeEvents</b>
<b>CallInputStatusChanged</b> ([in] ICall *pCall,[in] VARIANT_BOOL Status)	<b>_ISkypeEvents</b>
<b>CallSeenStatusChanged</b> ([in] ICall *pCall, VARIANT_BOOL Status)	<b>_ISkypeEvents</b>
<b>CallStatus</b> ([in] ICall *pCall,[in] TCallStatus Status)	<b>_ISkypeEvents</b>
<b>CallTransferStatusChanged</b> ([in] ICall *pCall,[in] TCallStatus Status)	<b>_ISkypeEvents</b>
<b>CallVideoReceiveStatusChanged</b> ([in] ICall *pCall, [in] TCallVideoSendStatus Status)	<b>_ISkypeEvents</b>
<b>CallVideoSendStatusChanged</b> ([in] ICall *pCall,[in] TCallVideoSendStatus Status)	<b>_ISkypeEvents</b>
<b>CallVideoStatusChanged</b> ([in] ICall *pCall,[in] TCallVideoStatus Status)	<b>_ISkypeEvents</b>
<b>ChatMemberRoleChanged</b> ([in] IChatMember	

*pMember,[in] TChatMemberRole Role)	_ISkypeEvents
<b>ChatMembersChanged</b> ([in] IChat *pChat,[in] IUserCollection *pMembers)	_ISkypeEvents
<b>Command</b> ([in] ICommand *pCommand)	_ISkypeEvents
<b>ConnectionStatus</b> ([in] TConnectionStatus Status)	_ISkypeEvents
<b>ContactsFocused</b> ([in] BSTR Username)	_ISkypeEvents
<b>Error</b> ([in] ICommand *pCommand,[in] LONG Number,[in] BSTR Description)	_ISkypeEvents
<b>FileTransferStatusChanged</b> ([in] IFileTransfer *pTransfer,[in] TFileTransferStatus Status)	_ISkypeEvents
<b>GroupDeleted</b> ([in] LONG GroupId)	_ISkypeEvents
<b>GroupExpanded</b> ([in] IGroup *pGroup,[in] VARIANT_BOOL Expanded)	_ISkypeEvents
<b>GroupUsers</b> ([in] IGroup *pGroup,[in] IUserCollection *pUsers)	_ISkypeEvents
<b>GroupVisible</b> ([in] IGroup *pGroup,[in] VARIANT_BOOL Visible)	_ISkypeEvents
<b>MessageHistory</b> ([in] BSTR Username)	_ISkypeEvents
<b>MessageStatus</b> ([in] IChatMessage *pMessage,[in] TChatMessageStatus Status)	_ISkypeEvents
<b>Mute</b> ([in] VARIANT_BOOL Mute)	_ISkypeEvents
<b>OnlineStatus</b> ([in] IUser *pUser,[in] TOnlineStatus Status)	_ISkypeEvents
<b>PluginEventClicked</b> ([in] IPluginEvent *pEvent)	_ISkypeEvents
<b>PluginMenuItemClicked</b> ([in] IPluginMenuItem *pMenuItem,[in] IUserCollection *pUsers,[in] TPluginContext PluginContext,[in] BSTR ContextId)	_ISkypeEvents
<b>Reply</b> ([in] ICommand *pCommand)	_ISkypeEvents
<b>SilentModeStatusChanged</b> ([in] VARIANT_BOOL Silent)	_ISkypeEvents
<b>SmsMessageStatusChanged</b> ([in] ISmsMessage *pMessage,[in] TSmsMessageStatus Status)	_ISkypeEvents
<b>SmsTargetStatusChanged</b> ([in] ISmsTarget *pTarget,[in] TSmsTargetStatus Status)	_ISkypeEvents
<b>UILanguageChanged</b> ([in] BSTR Code)	_ISkypeEvents

<b>UserAuthorizationRequestReceived</b> ([in] IUser *pUser)	<b>_ISkypeEvents</b>
<b>UserMood</b> ([in] IUser *pUser,[in] BSTR MoodText)	<b>_ISkypeEvents</b>
<b>UserStatus</b> ([in] TUserStatus Status)	<b>_ISkypeEvents</b>
<b>VoicemailStatus</b> ([in] IVoicemail *pMail,[in] TVoicemailStatus Status)	<b>_ISkypeEvents</b>
<b>WallpaperChanged</b> ([in] BSTR Path)	<b>_ISkypeEvents</b>

---

Copyright  2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IApplication Member List

This is the complete list of members for **IApplication**, including all inherited members.

<b>Connect</b> ([in] BSTR Username,[in, defaultvalue(0)] VARIANT_BOOL WaitConnected)	<b>IApplication</b>
<b>ConnectableUsers</b>	<b>IApplication</b>
<b>ConnectingUsers</b>	<b>IApplication</b>
<b>Create</b> ()	<b>IApplication</b>
<b>Delete</b> ()	<b>IApplication</b>
<b>Name</b>	<b>IApplication</b>
<b>ReceivedStreams</b>	<b>IApplication</b>
<b>SendDatagram</b> ([in] BSTR Text,[in, defaultvalue(0)] IApplicationStreamCollection *pStreams)	<b>IApplication</b>
<b>SendingStreams</b>	<b>IApplication</b>
<b>Streams</b>	<b>IApplication</b>

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IApplicationStream Member List

---

This is the complete list of members for **IApplicationStream**, including all inherited members.

<b>ApplicationName</b>	<b>IApplicationStream</b>
<b>DataLength</b>	<b>IApplicationStream</b>
<b>Disconnect()</b>	<b>IApplicationStream</b>
<b>Handle</b>	<b>IApplicationStream</b>
<b>PartnerHandle</b>	<b>IApplicationStream</b>
<b>Read</b> ([out, retval] BSTR *pText)	<b>IApplicationStream</b>
<b>SendDatagram</b> ([in] BSTR Text)	<b>IApplicationStream</b>
<b>Write</b> ([in] BSTR Text)	<b>IApplicationStream</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IApplicationStreamCollection Member List

---

This is the complete list of members for **IApplicationStreamCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IApplicationStreamCollection</b>
<b>Add</b> ([in] IApplicationStream *pItem)	<b>IApplicationStreamCollection</b>
<b>Count</b>	<b>IApplicationStreamCollection</b>
<b>Item</b>	<b>IApplicationStreamCollection</b>
<b>Remove</b> ([in] long Index)	<b>IApplicationStreamCollection</b>
<b>RemoveAll</b> (void)	<b>IApplicationStreamCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICall Member List

This is the complete list of members for **ICall**, including all inherited members.

<b>Answer()</b>	<b>ICall</b>
<b>CanTransfer</b>	<b>ICall</b>
<b>CaptureMicDevice</b>	<b>ICall</b>
<b>Conferenceld</b>	<b>ICall</b>
<b>DTMF</b>	<b>ICall</b>
<b>Duration</b>	<b>ICall</b>
<b>FailureReason</b>	<b>ICall</b>
<b>Finish()</b>	<b>ICall</b>
<b>Forward()</b>	<b>ICall</b>
<b>ForwardedBy</b>	<b>ICall</b>
<b>Hold()</b>	<b>ICall</b>
<b>Id</b>	<b>ICall</b>
<b>InputDevice</b>	<b>ICall</b>
<b>InputStatus</b>	<b>ICall</b>
<b>Join([in] LONG Id)</b>	<b>ICall</b>
<b>OutputDevice</b>	<b>ICall</b>
<b>Participants</b>	<b>ICall</b>
<b>PartnerDisplayName</b>	<b>ICall</b>
<b>PartnerHandle</b>	<b>ICall</b>
<b>PstnNumber</b>	<b>ICall</b>
<b>PstnStatus</b>	<b>ICall</b>
<b>Rate</b>	<b>ICall</b>
<b>RateCurrency</b>	<b>ICall</b>
<b>RatePrecision</b>	<b>ICall</b>
<b>RedirectToVoicemail()</b>	<b>ICall</b>
<b>Resume()</b>	<b>ICall</b>
<b>Seen</b>	<b>ICall</b>
<b>StartVideoReceive()</b>	<b>ICall</b>



<b>StartVideoSend()</b>	<b>ICall</b>
<b>Status</b>	<b>ICall</b>
<b>StopVideoReceive()</b>	<b>ICall</b>
<b>StopVideoSend()</b>	<b>ICall</b>
<b>Subject</b>	<b>ICall</b>
<b>TargetIdentity</b>	<b>ICall</b>
<b>Timestamp</b>	<b>ICall</b>
<b>Transfer([in] BSTR Target)</b>	<b>ICall</b>
<b>TransferActive</b>	<b>ICall</b>
<b>TransferredBy</b>	<b>ICall</b>
<b>TransferredTo</b>	<b>ICall</b>
<b>TransferStatus</b>	<b>ICall</b>
<b>Type</b>	<b>ICall</b>
<b>VideoReceiveStatus</b>	<b>ICall</b>
<b>VideoSendStatus</b>	<b>ICall</b>
<b>VideoStatus</b>	<b>ICall</b>
<b>VmAllowedDuration</b>	<b>ICall</b>
<b>VmDuration</b>	<b>ICall</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICallChannel Member List

---

This is the complete list of members for **ICallChannel**, including all inherited members.

<b>Call</b>	<b>ICallChannel</b>
<b>Manager</b>	<b>ICallChannel</b>
<b>SendMessage</b> (BSTR Text)	<b>ICallChannel</b>
<b>Stream</b>	<b>ICallChannel</b>
<b>Type</b>	<b>ICallChannel</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICallChannelCollection Member List

---

This is the complete list of members for **ICallChannelCollection**, including all inherited members.

<b>_NewEnum</b>	<b>ICallChannelCollection</b>
<b>Add</b> ([in] ICallChannel *pChannel)	<b>ICallChannelCollection</b>
<b>Count</b>	<b>ICallChannelCollection</b>
<b>Item</b>	<b>ICallChannelCollection</b>
<b>Remove</b> ([in] long Index)	<b>ICallChannelCollection</b>
<b>RemoveAll</b> (void)	<b>ICallChannelCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICallChannelManager Member List

This is the complete list of members for **ICallChannelManager**, including all inherited members.

<b>Channels</b>	<b>ICallChannelManager</b>
<b>ChannelType</b>	<b>ICallChannelManager</b>
<b>Connect</b> ([in] ISkype *pSkype)	<b>ICallChannelManager</b>
<b>CreateApplication</b> ([in, defaultvalue("")] BSTR ApplicationName)	<b>ICallChannelManager</b>
<b>Created</b>	<b>ICallChannelManager</b>
<b>Disconnect</b> ()	<b>ICallChannelManager</b>
<b>Name</b>	<b>ICallChannelManager</b>

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



# ICallChannelMessage Member List

---

This is the complete list of members for **ICallChannelMessage**, including all inherited members.

**Text ICallChannelMessage**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICallCollection Member List

---

This is the complete list of members for **ICallCollection**, including all inherited members.

<b>_NewEnum</b>	<b>ICallCollection</b>
<b>Add</b> ([in] ICall *pCall)	<b>ICallCollection</b>
<b>Count</b>	<b>ICallCollection</b>
<b>Item</b>	<b>ICallCollection</b>
<b>Remove</b> ([in] long Index)	<b>ICallCollection</b>
<b>RemoveAll</b> (void)	<b>ICallCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IChat Member List

This is the complete list of members for **IChat**, including all inherited members.

<b>AcceptAdd</b> (void)	IChat
<b>ActiveMembers</b>	IChat
<b>ActivityTimestamp</b>	IChat
<b>Adder</b>	IChat
<b>AddMembers</b> ([in] IUserCollection *pMembers)	IChat
<b>AlertString</b>	IChat
<b>Applicants</b>	IChat
<b>Blob</b>	IChat
<b>Bookmark</b> ()	IChat
<b>Bookmarked</b>	IChat
<b>ClearRecentMessages</b> (void)	IChat
<b>Description</b>	IChat
<b>DialogPartner</b>	IChat
<b>Disband</b> (void)	IChat
<b>EnterPassword</b> ([in] BSTR Password)	IChat
<b>FriendlyName</b>	IChat
<b>GuideLines</b>	IChat
<b>Join</b> (void)	IChat
<b>Kick</b> ([in] BSTR Handle)	IChat
<b>KickBan</b> ([in] BSTR Handle)	IChat
<b>Leave</b> ()	IChat
<b>MemberObjects</b>	IChat
<b>Members</b>	IChat
<b>Messages</b>	IChat
<b>MyRole</b>	IChat
<b>MyStatus</b>	IChat
<b>Name</b>	IChat
<b>OpenWindow</b> ()	IChat

<b>Options</b>	<b>IChat</b>
<b>PasswordHint</b>	<b>IChat</b>
<b>Posters</b>	<b>IChat</b>
<b>RecentMessages</b>	<b>IChat</b>
<b>SendMessage</b> ([in] BSTR MessageText,[out, retval] IChatMessage **pVal)	<b>IChat</b>
<b>SetPassword</b> ([in] BSTR Password,[in, defaultvalue("")] BSTR Hint)	<b>IChat</b>
<b>Status</b>	<b>IChat</b>
<b>Timestamp</b>	<b>IChat</b>
<b>Topic</b>	<b>IChat</b>
<b>TopicXML</b>	<b>IChat</b>
<b>Type</b>	<b>IChat</b>
<b>Unbookmark()</b>	<b>IChat</b>

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IChatCollection Member List

---

This is the complete list of members for **IChatCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IChatCollection</b>
<b>Add</b> ([in] IChat *pItem)	<b>IChatCollection</b>
<b>Count</b>	<b>IChatCollection</b>
<b>Item</b>	<b>IChatCollection</b>
<b>Remove</b> ([in] long Index)	<b>IChatCollection</b>
<b>RemoveAll</b> (void)	<b>IChatCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IChatMember Member List

---

This is the complete list of members for **IChatMember**, including all inherited members.

<b>CanSetRoleTo</b>	<b>IChatMember</b>
<b>Chat</b>	<b>IChatMember</b>
<b>Handle</b>	<b>IChatMember</b>
<b>Id</b>	<b>IChatMember</b>
<b>IsActive</b>	<b>IChatMember</b>
<b>Role</b>	<b>IChatMember</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IChatMemberCollection Member List

---

This is the complete list of members for **IChatMemberCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IChatMemberCollection</b>
<b>Count</b>	<b>IChatMemberCollection</b>
<b>Item</b>	<b>IChatMemberCollection</b>

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IChatMessage Member List

---

This is the complete list of members for **IChatMessage**, including all inherited members.

<b>Body</b>	<b>IChatMessage</b>
<b>Chat</b>	<b>IChatMessage</b>
<b>ChatName</b>	<b>IChatMessage</b>
<b>EditedBy</b>	<b>IChatMessage</b>
<b>EditedTimestamp</b>	<b>IChatMessage</b>
<b>FromDisplayName</b>	<b>IChatMessage</b>
<b>FromHandle</b>	<b>IChatMessage</b>
<b>Id</b>	<b>IChatMessage</b>
<b>IsEditable</b>	<b>IChatMessage</b>
<b>LeaveReason</b>	<b>IChatMessage</b>
<b>Options</b>	<b>IChatMessage</b>
<b>Role</b>	<b>IChatMessage</b>
<b>Seen</b>	<b>IChatMessage</b>
<b>Sender</b>	<b>IChatMessage</b>
<b>Status</b>	<b>IChatMessage</b>
<b>Timestamp</b>	<b>IChatMessage</b>
<b>Type</b>	<b>IChatMessage</b>
<b>Users</b>	<b>IChatMessage</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IChatMessageCollection Member List

---

This is the complete list of members for **IChatMessageCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IChatMessageCollection</b>
<b>Add</b> ([in] IChatMessage *pItem)	<b>IChatMessageCollection</b>
<b>Count</b>	<b>IChatMessageCollection</b>
<b>Item</b>	<b>IChatMessageCollection</b>
<b>Remove</b> ([in] long Index)	<b>IChatMessageCollection</b>
<b>RemoveAll</b> (void)	<b>IChatMessageCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IClient Member List

This is the complete list of members for **IClient**, including all inherited members.

<b>ButtonPressed</b> ([in] BSTR Key)	<b>IClient</b>
<b>ButtonReleased</b> ([in] BSTR Key)	<b>IClient</b>
<b>CreateEvent</b> ([in] BSTR EventId,[in] BSTR Caption,[in] BSTR Hint,[out, retval] IPluginEvent **pEvent)	<b>IClient</b>
<b>CreateMenuItem</b> ([in] BSTR MenuItemId,[in] TPluginContext PluginContext,[in] BSTR CaptionText,[in, defaultvalue("")] BSTR HintText,[in, defaultvalue("")] BSTR IconPath,[in, defaultvalue(-1)] VARIANT_BOOL Enabled,[in, defaultvalue(pluginContactTypeAll)] TPluginContactType ContactType,[in, defaultvalue(0)] VARIANT_BOOL MultipleContacts,[out, retval] IPluginMenuItem **pMenuItem)	<b>IClient</b>
<b>Focus</b> ()	<b>IClient</b>
<b>IsRunning</b>	<b>IClient</b>
<b>Minimize</b> ()	<b>IClient</b>
<b>OpenAddContactDialog</b> ([in, defaultvalue("")] BSTR Username)	<b>IClient</b>
<b>OpenAuthorizationDialog</b> ([in] BSTR Username)	<b>IClient</b>
<b>OpenBlockedUsersDialog</b> ()	<b>IClient</b>
<b>OpenCallHistoryTab</b> ()	<b>IClient</b>
<b>OpenConferenceDialog</b> ()	<b>IClient</b>
<b>OpenContactsTab</b> ()	<b>IClient</b>
<b>OpenDialog</b> ([in] BSTR Name,[in, defaultvalue("")] BSTR Param1,[in, defaultvalue("")] BSTR Param2)	<b>IClient</b>
<b>OpenDialpadTab</b> ()	<b>IClient</b>
<b>OpenFileTransferDialog</b> ([in] BSTR User,[in] BSTR Folder)	<b>IClient</b>
<b>OpenGettingStartedWizard</b> ()	<b>IClient</b>
<b>OpenImportContactsWizard</b> ()	<b>IClient</b>
<b>OpenLiveTab</b> ()	<b>IClient</b>
<b>OpenMessageDialog</b> ([in] BSTR Username,[in,	

defaultvalue("")] BSTR Text)	<b>IClient</b>
<b>OpenOptionsDialog</b> ([in] BSTR Page)	<b>IClient</b>
<b>OpenProfileDialog</b> ()	<b>IClient</b>
<b>OpenSearchDialog</b> ()	<b>IClient</b>
<b>OpenSendContactsDialog</b> ([in, defaultvalue("")] BSTR Username)	<b>IClient</b>
<b>OpenSmsDialog</b> ([in] BSTR SmsId)	<b>IClient</b>
<b>OpenUserInfoDialog</b> ([in] BSTR Username)	<b>IClient</b>
<b>OpenVideoTestDialog</b> ()	<b>IClient</b>
<b>Shutdown</b> ()	<b>IClient</b>
<b>Start</b> ([in, defaultvalue(0)] VARIANT_BOOL Minimized,[in, defaultvalue(0)] VARIANT_BOOL Nosplash)	<b>IClient</b>
<b>Wallpaper</b>	<b>IClient</b>

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ICommand Member List

---

This is the complete list of members for **ICommand**, including all inherited members.

<b>Blocking</b>	<b>ICommand</b>
<b>Command</b>	<b>ICommand</b>
<b>Expected</b>	<b>ICommand</b>
<b>Id</b>	<b>ICommand</b>
<b>Reply</b>	<b>ICommand</b>
<b>Timeout</b>	<b>ICommand</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IConference Member List

---

This is the complete list of members for **IConference**, including all inherited members.

<b>ActiveCalls</b>	<b>IConference</b>
<b>Calls</b>	<b>IConference</b>
<b>Finish()</b>	<b>IConference</b>
<b>Hold()</b>	<b>IConference</b>
<b>Id</b>	<b>IConference</b>
<b>Resume()</b>	<b>IConference</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



# IConferenceCollection Member List

---

This is the complete list of members for **IConferenceCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IConferenceCollection</b>
<b>Add</b> ([in] IConference *pItem)	<b>IConferenceCollection</b>
<b>Count</b>	<b>IConferenceCollection</b>
<b>Item</b>	<b>IConferenceCollection</b>
<b>Remove</b> ([in] long Index)	<b>IConferenceCollection</b>
<b>RemoveAll</b> (void)	<b>IConferenceCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IConversion Member List

This is the complete list of members for **IConversion**, including all inherited members.

<b>AttachmentStatusToText</b> ([in] TAttachmentStatus status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>BuddyStatusToText</b> ([in] TBuddyStatus status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>CallFailureReasonToText</b> ([in] TCallFailureReason reason,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>CallStatusToText</b> ([in] TCallStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>CallTypeToText</b> ([in] TCallType CallType,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>CallVideoSendStatusToText</b> ([in] TCallVideoSendStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>CallVideoStatusToText</b> ([in] TCallVideoStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>ChatLeaveReasonToText</b> ([in] TChatLeaveReason reason,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>ChatMessageStatusToText</b> ([in] TChatMessageStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>ChatMessageTypeToText</b> ([in] TChatMessageType Type,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>ChatStatusToText</b> ([in] TChatStatus status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>ConnectionStatusToText</b> ([in] TConnectionStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>GroupTypeToText</b> ([in] TGroupType Type,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>Language</b>	<b>IConversion</b>
<b>OnlineStatusToText</b> ([in] TOnlineStatus status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>SmsMessageStatusToText</b> ([in] TSmsMessageStatus	<b>IConversion</b>

status,[out, retval] BSTR *pVal)	
<b>SmsMessageTypeToText</b> ([in] TSmsMessageType type, [out, retval] BSTR *pVal)	<b>IConversion</b>
<b>SmsTargetStatusToText</b> ([in] TSmsTargetStatus status, [out, retval] BSTR *pVal)	<b>IConversion</b>
<b>TextToAttachmentStatus</b> ([in] BSTR Text,[out, retval] TAttachmentStatus *pVal)	<b>IConversion</b>
<b>TextToBuddyStatus</b> ([in] BSTR Text,[out, retval] TBuddyStatus *pVal)	<b>IConversion</b>
<b>TextToCallStatus</b> ([in] BSTR Text,[out, retval] TCallStatus *pVal)	<b>IConversion</b>
<b>TextToCallType</b> ([in] BSTR Text,[out, retval] TCallType *pVal)	<b>IConversion</b>
<b>TextToChatMessageStatus</b> ([in] BSTR Text,[out, retval] TChatMessageStatus *pVal)	<b>IConversion</b>
<b>TextToChatMessageType</b> ([in] BSTR Text,[out, retval] TChatMessageType *pVal)	<b>IConversion</b>
<b>TextToConnectionStatus</b> ([in] BSTR Text,[out, retval] TConnectionStatus *pVal)	<b>IConversion</b>
<b>TextToGroupType</b> ([in] BSTR Text,[out, retval] TGroupType *pVal)	<b>IConversion</b>
<b>TextToOnlineStatus</b> ([in] BSTR Text,[out, retval] TOnlineStatus *pVal)	<b>IConversion</b>
<b>TextToUserSex</b> ([in] BSTR Text,[out, retval] TUserSex *pVal)	<b>IConversion</b>
<b>TextToUserStatus</b> ([in] BSTR Text,[out, retval] TUserStatus *pVal)	<b>IConversion</b>
<b>TextToVoicemailStatus</b> ([in] BSTR Text,[out, retval] TVoicemailStatus *pVal)	<b>IConversion</b>
<b>UserSexToText</b> ([in] TUserSex Sex,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>UserStatusToText</b> ([in] TUserStatus Status,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>VoicemailFailureReasonToText</b> ([in] TVoicemailFailureReason code,[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>VoicemailStatusToText</b> ([in] TVoicemailStatus status,	

[out, retval] BSTR *pVal)	<b>IConversion</b>
<b>VoicemailTypeToText</b> ([in] TVoicemailType type,[out, retval] BSTR *pVal)	<b>IConversion</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IFileTransfer Member List

---

This is the complete list of members for **IFileTransfer**, including all inherited members.

<b>BytesPerSecond</b>	<b>IFileTransfer</b>
<b>BytesTransferred</b>	<b>IFileTransfer</b>
<b>FailureReason</b>	<b>IFileTransfer</b>
<b>FileName</b>	<b>IFileTransfer</b>
<b>FilePath</b>	<b>IFileTransfer</b>
<b>FinishTime</b>	<b>IFileTransfer</b>
<b>Id</b>	<b>IFileTransfer</b>
<b>PartnerDisplayName</b>	<b>IFileTransfer</b>
<b>PartnerHandle</b>	<b>IFileTransfer</b>
<b>StartTime</b>	<b>IFileTransfer</b>
<b>Status</b>	<b>IFileTransfer</b>
<b>Type</b>	<b>IFileTransfer</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



## IFileTransferCollection Member List

---

This is the complete list of members for **IFileTransferCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IFileTransferCollection</b>
<b>Add</b> ([in] IFileTransfer *pItem)	<b>IFileTransferCollection</b>
<b>Count</b>	<b>IFileTransferCollection</b>
<b>Item</b>	<b>IFileTransferCollection</b>
<b>Remove</b> ([in] long Index)	<b>IFileTransferCollection</b>
<b>RemoveAll</b> (void)	<b>IFileTransferCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IGroup Member List

---

This is the complete list of members for **IGroup**, including all inherited members.

<b>Accept()</b>	<b>IGroup</b>
<b>AddUser</b> ([in] BSTR Username)	<b>IGroup</b>
<b>CustomGroupId</b>	<b>IGroup</b>
<b>Decline()</b>	<b>IGroup</b>
<b>DisplayName</b>	<b>IGroup</b>
<b>Id</b>	<b>IGroup</b>
<b>IsExpanded</b>	<b>IGroup</b>
<b>IsVisible</b>	<b>IGroup</b>
<b>OnlineUsers</b>	<b>IGroup</b>
<b>RemoveUser</b> ([in] BSTR Username)	<b>IGroup</b>
<b>Share</b> ([in, defaultvalue("")] BSTR MessageText)	<b>IGroup</b>
<b>Type</b>	<b>IGroup</b>
<b>Users</b>	<b>IGroup</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IGroupCollection Member List

---

This is the complete list of members for **IGroupCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IGroupCollection</b>
<b>Add</b> ([in] IGroup *pGroup)	<b>IGroupCollection</b>
<b>Count</b>	<b>IGroupCollection</b>
<b>Item</b>	<b>IGroupCollection</b>
<b>Remove</b> ([in] long Index)	<b>IGroupCollection</b>
<b>RemoveAll</b> (void)	<b>IGroupCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IParticipant Member List

---

This is the complete list of members for **IParticipant**, including all inherited members.

<b>CallStatus</b>	<b>IParticipant</b>
<b>CallType</b>	<b>IParticipant</b>
<b>DisplayName</b>	<b>IParticipant</b>
<b>Handle</b>	<b>IParticipant</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



## IParticipantCollection Member List

---

This is the complete list of members for **IParticipantCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IParticipantCollection</b>
<b>Add</b> ([in] IParticipant *pParticipant)	<b>IParticipantCollection</b>
<b>Count</b>	<b>IParticipantCollection</b>
<b>Item</b>	<b>IParticipantCollection</b>
<b>Remove</b> ([in] long Index)	<b>IParticipantCollection</b>
<b>RemoveAll</b> (void)	<b>IParticipantCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IPluginEvent Member List

---

This is the complete list of members for **IPluginEvent**, including all inherited members.

**Delete**(void) **IPluginEvent**

**Id** **IPluginEvent**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IPluginMenuItem Member List

---

This is the complete list of members for **IPluginMenuItem**, including all inherited members.

<b>Caption</b>	<b>IPluginMenuItem</b>
<b>Delete(void)</b>	<b>IPluginMenuItem</b>
<b>Enabled</b>	<b>IPluginMenuItem</b>
<b>Hint</b>	<b>IPluginMenuItem</b>
<b>Id</b>	<b>IPluginMenuItem</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IProfile Member List

This is the complete list of members for **IProfile**, including all inherited members.

About	IProfile
Balance	IProfile
BalanceCurrency	IProfile
BalanceToText	IProfile
Birthday	IProfile
CallApplyCF	IProfile
CallForwardRules	IProfile
CallNoAnswerTimeout	IProfile
CallSendToVM	IProfile
City	IProfile
Country	IProfile
FullName	IProfile
Homepage	IProfile
IPCountry	IProfile
Languages	IProfile
MoodText	IProfile
PhoneHome	IProfile
PhoneMobile	IProfile
PhoneOffice	IProfile
Province	IProfile
RichMoodText	IProfile
Sex	IProfile
Timezone	IProfile
ValidatedSmsNumbers	IProfile

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



## ISettings Member List

This is the complete list of members for **ISettings**, including all inherited members.

<b>AEC</b>	<b>ISettings</b>
<b>AGC</b>	<b>ISettings</b>
<b>AudioIn</b>	<b>ISettings</b>
<b>AudioOut</b>	<b>ISettings</b>
<b>AutoAway</b>	<b>ISettings</b>
<b>Avatar</b>	<b>ISettings</b>
<b>Language</b>	<b>ISettings</b>
<b>LoadAvatarFromFile</b> ([in] BSTR Filename,[in, defaultvalue("1")] BSTR AvatarId)	<b>ISettings</b>
<b>PCSpeaker</b>	<b>ISettings</b>
<b>ResetIdleTimer</b> (void)	<b>ISettings</b>
<b>Ringer</b>	<b>ISettings</b>
<b>RingTone</b>	<b>ISettings</b>
<b>RingToneStatus</b>	<b>ISettings</b>
<b>SaveAvatarToFile</b> ([in] BSTR Filename,[in, defaultvalue("1")] BSTR AvatarId)	<b>ISettings</b>
<b>VideoIn</b>	<b>ISettings</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# ISkype Member List

This is the complete list of members for **ISkype**, including all inherited members.

<b>ActiveCalls</b>	<b>ISkype</b>
<b>ActiveChats</b>	<b>ISkype</b>
<b>ActiveFileTransfers</b>	<b>ISkype</b>
<b>ApiSecurityContextEnabled</b>	<b>ISkype</b>
<b>ApiWrapperVersion</b>	<b>ISkype</b>
<b>Application</b>	<b>ISkype</b>
<b>AsyncSearchUsers</b> ([in] BSTR Target,[out, retval] LONG *pCookie)	<b>ISkype</b>
<b>Attach</b> ([in, defaultvalue(5)] LONG Protocol,[in, defaultvalue(-1)] VARIANT_BOOL Wait)	<b>ISkype</b>
<b>AttachmentStatus</b>	<b>ISkype</b>
<b>BookmarkedChats</b>	<b>ISkype</b>
<b>Cache</b>	<b>ISkype</b>
<b>Call</b>	<b>ISkype</b>
<b>Calls</b>	<b>ISkype</b>
<b>ChangeUserStatus</b> ([in] TUserStatus newVal)	<b>ISkype</b>
<b>Chat</b>	<b>ISkype</b>
<b>Chats</b>	<b>ISkype</b>
<b>ClearCallHistory</b> ([in, defaultvalue("ALL")] BSTR Username,[in, defaultvalue(chsAllCalls)] TCallHistory type)	<b>ISkype</b>
<b>ClearChatHistory</b> ()	<b>ISkype</b>
<b>ClearVoicemailHistory</b> ()	<b>ISkype</b>
<b>Client</b>	<b>ISkype</b>
<b>Command</b>	<b>ISkype</b>
<b>CommandId</b>	<b>ISkype</b>
<b>Conference</b>	<b>ISkype</b>
<b>Conferences</b>	<b>ISkype</b>
<b>ConnectionStatus</b>	<b>ISkype</b>

<b>Convert</b>	<b>ISkype</b>
<b>CreateChatMultiple</b> ([in] IUserCollection *pMembers,[out, retval] IChat **pChat)	<b>ISkype</b>
<b>CreateChatUsingBlob</b> ([in] BSTR Blob,[out, retval] IChat **pChat)	<b>ISkype</b>
<b>CreateChatWith</b> ([in] BSTR Username,[out, retval] IChat **pVal)	<b>ISkype</b>
<b>CreateGroup</b> ([in] BSTR GroupName,[out, retval] IGroup **pGroup)	<b>ISkype</b>
<b>CreateSms</b> ([in] TSmsMessageType MessageType,[in] BSTR TargetNumbers,[out, retval] ISmsMessage **pMessage)	<b>ISkype</b>
<b>CurrentUser</b>	<b>ISkype</b>
<b>CurrentUserHandle</b>	<b>ISkype</b>
<b>CurrentUserProfile</b>	<b>ISkype</b>
<b>CurrentUserStatus</b>	<b>ISkype</b>
<b>CustomGroups</b>	<b>ISkype</b>
<b>DeleteGroup</b> ([in] LONG GroupId)	<b>ISkype</b>
<b>EnableApiSecurityContext</b> ([in] TApiSecurityContext Context)	<b>ISkype</b>
<b>FileTransfers</b>	<b>ISkype</b>
<b>FindChatUsingBlob</b> ([in] BSTR Blob,[out, retval] IChat **pChat)	<b>ISkype</b>
<b>FocusedContacts</b>	<b>ISkype</b>
<b>FriendlyName</b>	<b>ISkype</b>
<b>Friends</b>	<b>ISkype</b>
<b>Greeting</b>	<b>ISkype</b>
<b>Groups</b>	<b>ISkype</b>
<b>HardwiredGroups</b>	<b>ISkype</b>
<b>Message</b>	<b>ISkype</b>
<b>Messages</b>	<b>ISkype</b>
<b>MissedCalls</b>	<b>ISkype</b>
<b>MissedChats</b>	<b>ISkype</b>
<b>MissedMessages</b>	<b>ISkype</b>
<b>MissedSmss</b>	<b>ISkype</b>

<b>MissedVoicemails</b>	<b>ISkype</b>
<b>Mute</b>	<b>ISkype</b>
<b>PlaceCall</b> ([in] BSTR Target,[in, defaultvalue("")] BSTR Target2,[in, defaultvalue("")] BSTR Target3,[in, defaultvalue("")] BSTR Target4,[out, retval] ICall **pCall)	<b>ISkype</b>
<b>PredictiveDialerCountry</b>	<b>ISkype</b>
<b>Privilege</b>	<b>ISkype</b>
<b>Profile</b>	<b>ISkype</b>
<b>Property</b>	<b>ISkype</b>
<b>Protocol</b>	<b>ISkype</b>
<b>RecentChats</b>	<b>ISkype</b>
<b>ResetCache()</b>	<b>ISkype</b>
<b>SearchForUsers</b> ([in] BSTR Target,[out, retval] IUserCollection **pVal)	<b>ISkype</b>
<b>SendCommand</b> ([in] ICommand *pCommand)	<b>ISkype</b>
<b>SendMessage</b> ([in] BSTR Username,[in] BSTR Text,[out, retval] IChatMessage **pMessage)	<b>ISkype</b>
<b>SendSms</b> ([in] BSTR TargetNumbers,[in] BSTR MessageText,[in, defaultvalue("")] BSTR ReplyToNumber,[out, retval] ISmsMessage **pMessage)	<b>ISkype</b>
<b>SendVoicemail</b> ([in] BSTR Username,[out, retval] IVoicemail **pVoicemail)	<b>ISkype</b>
<b>Settings</b>	<b>ISkype</b>
<b>SilentMode</b>	<b>ISkype</b>
<b>Smss</b>	<b>ISkype</b>
<b>Timeout</b>	<b>ISkype</b>
<b>User</b>	<b>ISkype</b>
<b>UsersWaitingAuthorization</b>	<b>ISkype</b>
<b>Variable</b>	<b>ISkype</b>
<b>Version</b>	<b>ISkype</b>
<b>Voicemail</b>	<b>ISkype</b>
<b>Voicemails</b>	<b>ISkype</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# ISmsChunk Member List

---

This is the complete list of members for **ISmsChunk**, including all inherited members.

<b>CharactersLeft</b>	<b>ISmsChunk</b>
<b>Id</b>	<b>ISmsChunk</b>
<b>Message</b>	<b>ISmsChunk</b>
<b>Text</b>	<b>ISmsChunk</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



## ISmsChunkCollection Member List

---

This is the complete list of members for **ISmsChunkCollection**, including all inherited members.

<b>_NewEnum</b>	<b>ISmsChunkCollection</b>
<b>Add</b> ([in] ISmsChunk *pItem)	<b>ISmsChunkCollection</b>
<b>Count</b>	<b>ISmsChunkCollection</b>
<b>Item</b>	<b>ISmsChunkCollection</b>
<b>Remove</b> ([in] long Index)	<b>ISmsChunkCollection</b>
<b>RemoveAll</b> (void)	<b>ISmsChunkCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ISmsMessage Member List

---

This is the complete list of members for **ISmsMessage**, including all inherited members.

<b>Body</b>	<b>ISmsMessage</b>
<b>Chunks</b>	<b>ISmsMessage</b>
<b>Delete()</b>	<b>ISmsMessage</b>
<b>FailureReason</b>	<b>ISmsMessage</b>
<b>Id</b>	<b>ISmsMessage</b>
<b>IsFailedUnseen</b>	<b>ISmsMessage</b>
<b>Price</b>	<b>ISmsMessage</b>
<b>PriceCurrency</b>	<b>ISmsMessage</b>
<b>PricePrecision</b>	<b>ISmsMessage</b>
<b>ReplyToNumber</b>	<b>ISmsMessage</b>
<b>Seen</b>	<b>ISmsMessage</b>
<b>Send()</b>	<b>ISmsMessage</b>
<b>Status</b>	<b>ISmsMessage</b>
<b>TargetNumbers</b>	<b>ISmsMessage</b>
<b>Targets</b>	<b>ISmsMessage</b>
<b>Timestamp</b>	<b>ISmsMessage</b>
<b>Type</b>	<b>ISmsMessage</b>

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## ISmsMessageCollection Member List

---

This is the complete list of members for **ISmsMessageCollection**, including all inherited members.

<b>_NewEnum</b>	<b>ISmsMessageCollection</b>
<b>Add</b> ([in] ISmsMessage *pItem)	<b>ISmsMessageCollection</b>
<b>Count</b>	<b>ISmsMessageCollection</b>
<b>Item</b>	<b>ISmsMessageCollection</b>
<b>Remove</b> ([in] long Index)	<b>ISmsMessageCollection</b>
<b>RemoveAll</b> (void)	<b>ISmsMessageCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# ISmsTarget Member List

---

This is the complete list of members for **ISmsTarget**, including all inherited members.

<b>Message</b>	<b>ISmsTarget</b>
<b>Number</b>	<b>ISmsTarget</b>
<b>Status</b>	<b>ISmsTarget</b>

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		



## ISmsTargetCollection Member List

---

This is the complete list of members for **ISmsTargetCollection**, including all inherited members.

<b>_NewEnum</b>	<b>ISmsTargetCollection</b>
<b>Add</b> ([in] ISmsTarget *pItem)	<b>ISmsTargetCollection</b>
<b>Count</b>	<b>ISmsTargetCollection</b>
<b>Item</b>	<b>ISmsTargetCollection</b>
<b>Remove</b> ([in] long Index)	<b>ISmsTargetCollection</b>
<b>RemoveAll</b> (void)	<b>ISmsTargetCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IUser Member List

This is the complete list of members for **IUser**, including all inherited members.

About	IUser
Aliases	IUser
Birthday	IUser
BuddyStatus	IUser
CanLeaveVoicemail	IUser
City	IUser
Country	IUser
CountryCode	IUser
DisplayName	IUser
FullName	IUser
Handle	IUser
HasCallEquipment	IUser
Homepage	IUser
IsAuthorized	IUser
IsBlocked	IUser
IsCallForwardActive	IUser
IsSkypeOutContact	IUser
IsVideoCapable	IUser
IsVoicemailCapable	IUser
Language	IUser
LanguageCode	IUser
LastOnline	IUser
MoodText	IUser
NumberOfAuthBuddies	IUser
OnlineStatus	IUser
PhoneHome	IUser
PhoneMobile	IUser
PhoneOffice	IUser

<b>Province</b>	<b>IUser</b>
<b>ReceivedAuthRequest</b>	<b>IUser</b>
<b>RichMoodText</b>	<b>IUser</b>
<b>Sex</b>	<b>IUser</b>
<b>SpeedDial</b>	<b>IUser</b>
<b>Timezone</b>	<b>IUser</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IUserCollection Member List

---

This is the complete list of members for **IUserCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IUserCollection</b>
<b>Add</b> ([in] IUser *pUser)	<b>IUserCollection</b>
<b>Count</b>	<b>IUserCollection</b>
<b>Item</b>	<b>IUserCollection</b>
<b>Remove</b> ([in] long Index)	<b>IUserCollection</b>
<b>RemoveAll</b> (void)	<b>IUserCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

## IVoicemail Member List

---

This is the complete list of members for **IVoicemail**, including all inherited members.

<b>AllowedDuration</b>	<b>IVoicemail</b>
<b>Delete()</b>	<b>IVoicemail</b>
<b>Download()</b>	<b>IVoicemail</b>
<b>Duration</b>	<b>IVoicemail</b>
<b>FailureReason</b>	<b>IVoicemail</b>
<b>Id</b>	<b>IVoicemail</b>
<b>Open()</b>	<b>IVoicemail</b>
<b>PartnerDisplayName</b>	<b>IVoicemail</b>
<b>PartnerHandle</b>	<b>IVoicemail</b>
<b>SetUnplayed()</b>	<b>IVoicemail</b>
<b>StartPlayback()</b>	<b>IVoicemail</b>
<b>StartPlaybackInCall()</b>	<b>IVoicemail</b>
<b>StartRecording()</b>	<b>IVoicemail</b>
<b>Status</b>	<b>IVoicemail</b>
<b>StopPlayback()</b>	<b>IVoicemail</b>
<b>StopRecording()</b>	<b>IVoicemail</b>
<b>Timestamp</b>	<b>IVoicemail</b>
<b>Type</b>	<b>IVoicemail</b>
<b>Upload()</b>	<b>IVoicemail</b>



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		

# IVoicemailCollection Member List

---

This is the complete list of members for **IVoicemailCollection**, including all inherited members.

<b>_NewEnum</b>	<b>IVoicemailCollection</b>
<b>Add</b> ([in] IVoicemail *pItem)	<b>IVoicemailCollection</b>
<b>Count</b>	<b>IVoicemailCollection</b>
<b>Item</b>	<b>IVoicemailCollection</b>
<b>Remove</b> ([in] long Index)	<b>IVoicemailCollection</b>
<b>RemoveAll</b> (void)	<b>IVoicemailCollection</b>

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- a -

- [Accept\(\)](#) : [IGroup](#)
  - [AcceptAdd\(\)](#) : [IChat](#)
  - [Add\(\)](#) : [IParticipantCollection](#) , [IConferenceCollection](#) , [ICallChannelCollection](#) , [IVoicemailCollection](#) , [ICallCollection](#) , [IApplicationStreamCollection](#) , [IGroupCollection](#) , [IUserCollection](#) , [IChatMessageCollection](#) , [ISmsChunkCollection](#) , [ISmsTargetCollection](#) , [IChatCollection](#) , [ISmsMessageCollection](#) , [IFileTransferCollection](#)
  - [AddMembers\(\)](#) : [IChat](#)
  - [AddUser\(\)](#) : [IGroup](#)
  - [Answer\(\)](#) : [ICall](#)
  - [ApplicationConnecting\(\)](#) : [\\_ISkypeEvents](#)
  - [ApplicationDatagram\(\)](#) : [\\_ISkypeEvents](#)
  - [ApplicationReceiving\(\)](#) : [\\_ISkypeEvents](#)
  - [ApplicationSending\(\)](#) : [\\_ISkypeEvents](#)
  - [ApplicationStreams\(\)](#) : [\\_ISkypeEvents](#)
  - [AsyncSearchUsers\(\)](#) : [ISkype](#)
  - [AsyncSearchUsersFinished\(\)](#) : [\\_ISkypeEvents](#)
  - [Attach\(\)](#) : [ISkype](#)
  - [AttachmentStatus\(\)](#) : [\\_ISkypeEvents](#)
  - [AttachmentStatusToText\(\)](#) : [IConversion](#)
  - [AutoAway\(\)](#) : [\\_ISkypeEvents](#)
-

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>			<a href="#">Modules</a>			<a href="#">Classes</a>												
<a href="#">Files</a>		<a href="#">Examples</a>																		
<a href="#">Class List</a>		<a href="#">Class Members</a>																		
<a href="#">All</a>		<a href="#">Functions</a>			<a href="#">Properties</a>															
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- \_ -

- [\\_NewEnum](#) : [IUserCollection](#) , [ICallChannelCollection](#) , [IFileTransferCollection](#) , [ISmsMessageCollection](#) , [ISmsTargetCollection](#) , [ISmsChunkCollection](#) , [IGroupCollection](#) , [IApplicationStreamCollection](#) , [IVoicemailCollection](#) , [IConferenceCollection](#) , [IChatCollection](#) , [IChatMemberCollection](#) , [IChatMessageCollection](#) , [ICallCollection](#) , [IParticipantCollection](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">-</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- a -

- About : [IUser](#) , [IProfile](#)
- Accept() : [IGroup](#)
- AcceptAdd() : [IChat](#)
- ActiveCalls : [IConference](#) , [ISkype](#)
- ActiveChats : [ISkype](#)
- ActiveFileTransfers : [ISkype](#)
- ActiveMembers : [IChat](#)
- ActivityTimestamp : [IChat](#)
- Add() : [IConferenceCollection](#) , [IvoicemailCollection](#) , [IApplicationStreamCollection](#) , [IGroupCollection](#) , [ISmsChunkCollection](#) , [ISmsTargetCollection](#) , [IUserCollection](#) , [ISmsMessageCollection](#) , [IFileTransferCollection](#) , [IParticipantCollection](#) , [ICallChannelCollection](#) , [ICallCollection](#) , [IChatMessageCollection](#) , [IChatCollection](#)
- Adder : [IChat](#)
- AddMembers() : [IChat](#)
- AddUser() : [IGroup](#)
- AEC : [ISettings](#)
- AGC : [ISettings](#)
- AlertString : [IChat](#)
- Aliases : [IUser](#)
- AllowedDuration : [Ivoicemail](#)

- Answer() : **ICall**
- ApiSecurityContextEnabled : **ISkype**
- ApiWrapperVersion : **ISkype**
- Applicants : **IChat**
- Application : **ISkype**
- ApplicationConnecting() : **\_ISkypeEvents**
- ApplicationDatagram() : **\_ISkypeEvents**
- ApplicationName : **IApplicationStream**
- ApplicationReceiving() : **\_ISkypeEvents**
- ApplicationSending() : **\_ISkypeEvents**
- ApplicationStreams() : **\_ISkypeEvents**
- AsyncSearchUsers() : **ISkype**
- AsyncSearchUsersFinished() : **\_ISkypeEvents**
- Attach() : **ISkype**
- AttachmentStatus : **ISkype** , **\_ISkypeEvents**
- AttachmentStatusToText() : **IConversion**
- AudioIn : **ISettings**
- AudioOut : **ISettings**
- AutoAway : **ISettings** , **\_ISkypeEvents**
- Avatar : **ISettings**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- b -

- Balance : [IProfile](#)
- BalanceCurrency : [IProfile](#)
- BalanceToText : [IProfile](#)
- Birthday : [IUser](#) , [IProfile](#)
- Blob : [IChat](#)
- Blocking : [ICommand](#)
- Body : [IChatMessage](#) , [ISmsMessage](#)
- Bookmark() : [IChat](#)
- Bookmarked : [IChat](#)
- BookmarkedChats : [ISkype](#)
- BuddyStatus : [IUser](#)
- BuddyStatusToText() : [IConversion](#)
- ButtonPressed() : [IClient](#)
- ButtonReleased() : [IClient](#)
- BytesPerSecond : [IFileTransfer](#)
- BytesTransferred : [IFileTransfer](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- C -

- Cache : [ISkype](#)
- Call : [ISkype](#) , [ICallChannel](#)
- CallApplyCF : [IProfile](#)
- CallDtmfReceived() : [\\_ISkypeEvents](#)
- CallFailureReasonToText() : [IConversion](#)
- CallForwardRules : [IProfile](#)
- CallHistory() : [\\_ISkypeEvents](#)
- CallInputStatusChanged() : [\\_ISkypeEvents](#)
- CallNoAnswerTimeout : [IProfile](#)
- Calls : [IConference](#) , [ISkype](#)
- CallSeenStatusChanged() : [\\_ISkypeEvents](#)
- CallSendToVM : [IProfile](#)
- CallStatus() : [\\_ISkypeEvents](#) , [IParticipant](#)
- CallStatusToText() : [IConversion](#)
- CallTransferStatusChanged() : [\\_ISkypeEvents](#)
- CallType : [IParticipant](#)
- CallTypeToText() : [IConversion](#)
- CallVideoReceiveStatusChanged() : [\\_ISkypeEvents](#)
- CallVideoSendStatusChanged() : [\\_ISkypeEvents](#)
- CallVideoSendStatusToText() : [IConversion](#)
- CallVideoStatusChanged() : [\\_ISkypeEvents](#)
- CallVideoStatusToText() : [IConversion](#)
- CanLeaveVoicemail : [IUser](#)

- CanSetRoleTo : **IChatMember**
- CanTransfer : **ICall**
- Caption : **IPluginMenuItem**
- CaptureMicDevice : **ICall**
- ChangeUserStatus() : **ISkype**
- Channels : **ICallChannelManager** ,  
**\_ICallChannelManagerEvents**
- ChannelType : **ICallChannelManager**
- CharactersLeft : **ISmsChunk**
- Chat : **IChatMessage** , **IChatMember** , **ISkype**
- ChatLeaveReasonToText() : **IConversion**
- ChatMemberRoleChanged() : **\_ISkypeEvents**
- ChatMembersChanged() : **\_ISkypeEvents**
- ChatMessageStatusToText() : **IConversion**
- ChatMessageTypeToText() : **IConversion**
- ChatName : **IChatMessage**
- Chats : **ISkype**
- ChatStatusToText() : **IConversion**
- Chunks : **ISmsMessage**
- City : **IUser** , **IProfile**
- ClearCallHistory() : **ISkype**
- ClearChatHistory() : **ISkype**
- ClearRecentMessages() : **IChat**
- ClearVoicemailHistory() : **ISkype**
- Client : **ISkype**
- Command : **ICommand** , **ISkype** , **\_ISkypeEvents**
- CommandId : **ISkype**
- Conference : **ISkype**
- ConferenceId : **ICall**
- Conferences : **ISkype**
- Connect() : **ICallChannelManager** , **IApplication**
- ConnectableUsers : **IApplication**
- ConnectingUsers : **IApplication**
- ConnectionStatus : **ISkype** , **\_ISkypeEvents**
- ConnectionStatusToText() : **IConversion**
- ContactsFocused() : **\_ISkypeEvents**
- Convert : **ISkype**
- Count : **IChatMessageCollection** , **IChatCollection** ,  
**IFileTransferCollection** , **ICallCollection** ,

**IParticipantCollection , IUserCollection ,  
ICallChannelCollection , IGroupCollection ,  
ISmsTargetCollection , IChatMemberCollection ,  
IVoicemailCollection , IApplicationStreamCollection ,  
ISmsChunkCollection , ISmsMessageCollection ,  
IConferenceCollection**

- Country : **IUser , IProfile**
- CountryCode : **IUser**
- Create() : **IApplication**
- CreateApplication() : **ICallChannelManager**
- CreateChatMultiple() : **ISkype**
- CreateChatUsingBlob() : **ISkype**
- CreateChatWith() : **ISkype**
- Created : **ICallChannelManager , \_ICallChannelManagerEvents**
- CreateEvent() : **IClient**
- CreateGroup() : **ISkype**
- CreateMenuItem() : **IClient**
- CreateSms() : **ISkype**
- CurrentUser : **ISkype**
- CurrentUserHandle : **ISkype**
- CurrentUserProfile : **ISkype**
- CurrentUserStatus : **ISkype**
- CustomGroupId : **IGroup**
- CustomGroups : **ISkype**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- d -

- [DataLength](#) : [IApplicationStream](#)
- [Decline\(\)](#) : [IGroup](#)
- [Delete\(\)](#) : [IApplication](#) , [ISmsMessage](#) , [IVoicemail](#) , [IPluginMenuItem](#) , [IPluginEvent](#)
- [DeleteGroup\(\)](#) : [ISkype](#)
- [Description](#) : [IChat](#)
- [DialogPartner](#) : [IChat](#)
- [Disband\(\)](#) : [IChat](#)
- [Disconnect\(\)](#) : [IApplicationStream](#) , [ICallChannelManager](#)
- [DisplayName](#) : [IGroup](#) , [IParticipant](#) , [IUser](#)
- [Download\(\)](#) : [IVoicemail](#)
- [DTMF](#) : [ICall](#)
- [Duration](#) : [ICall](#) , [IVoicemail](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- e -

- EditedBy : [IChatMessage](#)
- EditedTimestamp : [IChatMessage](#)
- EnableApiSecurityContext() : [ISkype](#)
- Enabled : [IPluginMenuItem](#)
- EnterPassword() : [IChat](#)
- Error() : [\\_ISkypeEvents](#)
- Expected : [ICommand](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- f -

- [FailureReason](#) : [ICall](#) , [IVoicemail](#) , [IFileTransfer](#) , [ISmsMessage](#)
- [FileName](#) : [IFileTransfer](#)
- [FilePath](#) : [IFileTransfer](#)
- [FileTransfers](#) : [ISkype](#)
- [FileTransferStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [FindChatUsingBlob\(\)](#) : [ISkype](#)
- [Finish\(\)](#) : [ICall](#) , [IConference](#)
- [FinishTime](#) : [IFileTransfer](#)
- [Focus\(\)](#) : [IClient](#)
- [FocusedContacts](#) : [ISkype](#)
- [Forward\(\)](#) : [ICall](#)
- [ForwardedBy](#) : [ICall](#)
- [FriendlyName](#) : [ISkype](#) , [IChat](#)
- [Friends](#) : [ISkype](#)
- [FromDisplayName](#) : [IChatMessage](#)
- [FromHandle](#) : [IChatMessage](#)
- [FullName](#) : [IProfile](#) , [IUser](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>		<a href="#">Modules</a>		<a href="#">Classes</a>																
<a href="#">Files</a>		<a href="#">Examples</a>																				
<a href="#">Class List</a>		<a href="#">Class Members</a>																				
<a href="#">All</a>		<a href="#">Functions</a>		<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- g -

- Greeting : [ISkype](#)
- GroupDeleted() : [\\_ISkypeEvents](#)
- GroupExpanded() : [\\_ISkypeEvents](#)
- Groups : [ISkype](#)
- GroupTypeToText() : [IConversion](#)
- GroupUsers() : [\\_ISkypeEvents](#)
- GroupVisible() : [\\_ISkypeEvents](#)
- GuideLines : [IChat](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- h -

- Handle : [IUser](#) , [IParticipant](#) , [IApplicationStream](#) , [IChatMember](#)
- HardwiredGroups : [ISkype](#)
- HasCallEquipment : [IUser](#)
- Hint : [IPluginMenuItem](#)
- Hold() : [ICall](#) , [IConference](#)
- Homepage : [IProfile](#) , [IUser](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- i -

- [Id](#) : [ICommand](#) , [ICall](#) , [IChatMember](#) , [ISmsChunk](#) , [ISmsMessage](#) , [IConference](#) , [IPluginMenuItem](#) , [IPluginEvent](#) , [IChatMessage](#) , [IVoicemail](#) , [IFileTransfer](#) , [IGroup](#)
- [InputDevice](#) : [ICall](#)
- [InputStatus](#) : [ICall](#)
- [IPCountry](#) : [IProfile](#)
- [IsActive](#) : [IChatMember](#)
- [IsAuthorized](#) : [IUser](#)
- [IsBlocked](#) : [IUser](#)
- [IsCallForwardActive](#) : [IUser](#)
- [IsEditable](#) : [IChatMessage](#)
- [IsExpanded](#) : [IGroup](#)
- [IsFailedUnseen](#) : [ISmsMessage](#)
- [IsRunning](#) : [IClient](#)
- [IsSkypeOutContact](#) : [IUser](#)
- [IsVideoCapable](#) : [IUser](#)
- [IsVisible](#) : [IGroup](#)
- [IsVoicemailCapable](#) : [IUser](#)
- [Item](#) : [ICallChannelCollection](#) , [IChatMessageCollection](#) , [IChatCollection](#) , [ISmsTargetCollection](#) , [IParticipantCollection](#) , [IVoicemailCollection](#) , [IChatMemberCollection](#) , [IConferenceCollection](#) , [IApplicationStreamCollection](#) , [ICallCollection](#) , [IUserCollection](#) , [ISmsChunkCollection](#) ,

**IGroupCollection , IFileTransferCollection ,  
ISmsMessageCollection**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- j -

- [Join\(\)](#) : [ICall](#) , [IChat](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- k -

- [Kick\(\)](#) : [IChat](#)
- [KickBan\(\)](#) : [IChat](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- l -

- Language : [IUser](#) , [IConversion](#) , [ISettings](#)
- LanguageCode : [IUser](#)
- Languages : [IProfile](#)
- LastOnline : [IUser](#)
- Leave() : [IChat](#)
- LeaveReason : [IChatMessage](#)
- LoadAvatarFromFile() : [ISettings](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- m -

- Manager : [ICallChannel](#)
- MemberObjects : [IChat](#)
- Members : [IChat](#)
- Message : [ISmsChunk](#) , [ISmsTarget](#) , [ISkype](#) , [\\_ICallChannelManagerEvents](#)
- MessageHistory() : [\\_ISkypeEvents](#)
- Messages : [IChat](#) , [ISkype](#)
- MessageStatus() : [\\_ISkypeEvents](#)
- Minimize() : [IClient](#)
- MissedCalls : [ISkype](#)
- MissedChats : [ISkype](#)
- MissedMessages : [ISkype](#)
- MissedSmss : [ISkype](#)
- MissedVoicemails : [ISkype](#)
- MoodText : [IUser](#) , [IProfile](#)
- Mute() : [\\_ISkypeEvents](#) , [ISkype](#)
- MyRole : [IChat](#)
- MyStatus : [IChat](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<b><a href="#">Class Members</a></b>																					
<b><a href="#">All</a></b>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<b><a href="#">n</a></b>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- n -

- Name : [IChat](#) , [IApplication](#) , [ICallChannelManager](#)
- Number : [ISmsTarget](#)
- NumberOfAuthBuddies : [IUser](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- o -

- [OnlineStatus](#) : [IUser](#) , [\\_ISkypeEvents](#)
- [OnlineStatusToText\(\)](#) : [IConversion](#)
- [OnlineUsers](#) : [IGroup](#)
- [Open\(\)](#) : [IVoicemail](#)
- [OpenAddContactDialog\(\)](#) : [IClient](#)
- [OpenAuthorizationDialog\(\)](#) : [IClient](#)
- [OpenBlockedUsersDialog\(\)](#) : [IClient](#)
- [OpenCallHistoryTab\(\)](#) : [IClient](#)
- [OpenConferenceDialog\(\)](#) : [IClient](#)
- [OpenContactsTab\(\)](#) : [IClient](#)
- [OpenDialog\(\)](#) : [IClient](#)
- [OpenDialpadTab\(\)](#) : [IClient](#)
- [OpenFileTransferDialog\(\)](#) : [IClient](#)
- [OpenGettingStartedWizard\(\)](#) : [IClient](#)
- [OpenImportContactsWizard\(\)](#) : [IClient](#)
- [OpenLiveTab\(\)](#) : [IClient](#)
- [OpenMessageDialog\(\)](#) : [IClient](#)
- [OpenOptionsDialog\(\)](#) : [IClient](#)
- [OpenProfileDialog\(\)](#) : [IClient](#)
- [OpenSearchDialog\(\)](#) : [IClient](#)
- [OpenSendContactsDialog\(\)](#) : [IClient](#)
- [OpenSmsDialog\(\)](#) : [IClient](#)
- [OpenUserInfoDialog\(\)](#) : [IClient](#)



- OpenVideoTestDialog() : **IClient**
- OpenWindow() : **IChat**
- Options : **IChatMessage** , **IChat**
- OutputDevice : **ICall**

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- p -

- Participants : [ICall](#)
- PartnerDisplayName : [ICall](#) , [IFileTransfer](#) , [IVoicemail](#)
- PartnerHandle : [IFileTransfer](#) , [ICall](#) , [IVoicemail](#) , [IApplicationStream](#)
- PasswordHint : [IChat](#)
- PCSpeaker : [ISettings](#)
- PhoneHome : [IUser](#) , [IProfile](#)
- PhoneMobile : [IUser](#) , [IProfile](#)
- PhoneOffice : [IProfile](#) , [IUser](#)
- PlaceCall() : [ISkype](#)
- PluginEventClicked() : [\\_ISkypeEvents](#)
- PluginMenuItemClicked() : [\\_ISkypeEvents](#)
- Posters : [IChat](#)
- PredictiveDialerCountry : [ISkype](#)
- Price : [ISmsMessage](#)
- PriceCurrency : [ISmsMessage](#)
- PricePrecision : [ISmsMessage](#)
- Privilege : [ISkype](#)
- Profile : [ISkype](#)
- Property : [ISkype](#)
- Protocol : [ISkype](#)
- Province : [IUser](#) , [IProfile](#)
- PstnNumber : [ICall](#)

- PstnStatus : **ICall**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- r -

- Rate : [ICall](#)
- RateCurrency : [ICall](#)
- RatePrecision : [ICall](#)
- Read() : [IApplicationStream](#)
- ReceivedAuthRequest : [IUser](#)
- ReceivedStreams : [IApplication](#)
- RecentChats : [ISkype](#)
- RecentMessages : [IChat](#)
- RedirectToVoicemail() : [ICall](#)
- Remove() : [IVoicemailCollection](#) , [IApplicationStreamCollection](#) , [IGroupCollection](#) , [ISmsChunkCollection](#) , [IUserCollection](#) , [ISmsTargetCollection](#) , [ISmsMessageCollection](#) , [IParticipantCollection](#) , [IFileTransferCollection](#) , [ICallChannelCollection](#) , [ICallCollection](#) , [IChatMessageCollection](#) , [IChatCollection](#) , [IConferenceCollection](#)
- RemoveAll() : [ISmsMessageCollection](#) , [IGroupCollection](#) , [ISmsChunkCollection](#) , [IChatMessageCollection](#) , [ISmsTargetCollection](#) , [IFileTransferCollection](#) , [ICallChannelCollection](#) , [IApplicationStreamCollection](#) , [IUserCollection](#) , [IParticipantCollection](#) , [ICallCollection](#) , [IChatCollection](#) , [IConferenceCollection](#) , [IVoicemailCollection](#)

- RemoveUser() : **IGroup**
- Reply() : **\_ISkypeEvents** , **ICommand**
- ReplyToNumber : **ISmsMessage**
- ResetCache() : **ISkype**
- ResetIdleTimer() : **ISettings**
- Resume() : **IConference** , **ICall**
- RichMoodText : **IUser** , **IProfile**
- Ringer : **ISettings**
- RingTone : **ISettings**
- RingToneStatus : **ISettings**
- Role : **IChatMessage** , **IChatMember**

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- s -

- [SaveAvatarToFile\(\)](#) : [ISettings](#)
- [SearchForUsers\(\)](#) : [ISkype](#)
- [Seen](#) : [IChatMessage](#) , [ISmsMessage](#) , [ICall](#)
- [Send\(\)](#) : [ISmsMessage](#)
- [SendCommand\(\)](#) : [ISkype](#)
- [SendDatagram\(\)](#) : [IApplicationStream](#) , [IApplication](#)
- [Sender](#) : [IChatMessage](#)
- [SendingStreams](#) : [IApplication](#)
- [SendMessage\(\)](#) : [IChat](#) , [ISkype](#)
- [SendSms\(\)](#) : [ISkype](#)
- [SendTextMessage\(\)](#) : [ICallChannel](#)
- [SendVoicemail\(\)](#) : [ISkype](#)
- [SetPassword\(\)](#) : [IChat](#)
- [Settings](#) : [ISkype](#)
- [SetUnplayed\(\)](#) : [IVoicemail](#)
- [Sex](#) : [IUser](#) , [IProfile](#)
- [Share\(\)](#) : [IGroup](#)
- [Shutdown\(\)](#) : [IClient](#)
- [SilentMode](#) : [ISkype](#)
- [SilentModeStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [SmsMessageStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [SmsMessageStatusToText\(\)](#) : [IConversion](#)
- [SmsMessageTypeToText\(\)](#) : [IConversion](#)

- Smss : **ISkype**
- SmsTargetStatusChanged() : **\_ISkypeEvents**
- SmsTargetStatusToText() : **IConversion**
- SpeedDial : **IUser**
- Start() : **IClient**
- StartPlayback() : **IVoicemail**
- StartPlaybackInCall() : **IVoicemail**
- StartRecording() : **IVoicemail**
- StartTime : **IFileTransfer**
- StartVideoReceive() : **ICall**
- StartVideoSend() : **ICall**
- Status : **IFileTransfer** , **ISmsMessage** , **ICall** , **ISmsTarget** , **IVoicemail** , **IChatMessage** , **IChat**
- StopPlayback() : **IVoicemail**
- StopRecording() : **IVoicemail**
- StopVideoReceive() : **ICall**
- StopVideoSend() : **ICall**
- Stream : **ICallChannel**
- Streams : **IApplication**
- Subject : **ICall**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- t -

- [TargetIdentity](#) : [ICall](#)
- [TargetNumbers](#) : [ISmsMessage](#)
- [Targets](#) : [ISmsMessage](#)
- [Text](#) : [ISmsChunk](#) , [ICallChannelMessage](#)
- [TextToAttachmentStatus\(\)](#) : [IConversion](#)
- [TextToBuddyStatus\(\)](#) : [IConversion](#)
- [TextToCallStatus\(\)](#) : [IConversion](#)
- [TextToCallType\(\)](#) : [IConversion](#)
- [TextToChatMessageStatus\(\)](#) : [IConversion](#)
- [TextToChatMessageType\(\)](#) : [IConversion](#)
- [TextToConnectionStatus\(\)](#) : [IConversion](#)
- [TextToGroupType\(\)](#) : [IConversion](#)
- [TextToOnlineStatus\(\)](#) : [IConversion](#)
- [TextToUserSex\(\)](#) : [IConversion](#)
- [TextToUserStatus\(\)](#) : [IConversion](#)
- [TextToVoicemailStatus\(\)](#) : [IConversion](#)
- [Timeout](#) : [ICommand](#) , [ISkype](#)
- [Timestamp](#) : [IVoicemail](#) , [ICall](#) , [IChatMessage](#) , [IChat](#) , [ISmsMessage](#)
- [Timezone](#) : [IUser](#) , [IProfile](#)
- [Topic](#) : [IChat](#)
- [TopicXML](#) : [IChat](#)
- [Transfer\(\)](#) : [ICall](#)



- TransferActive : **ICall**
- TransferredBy : **ICall**
- TransferredTo : **ICall**
- TransferStatus : **ICall**
- Type : **IVoicemail** , **ISmsMessage** , **IChatMessage** , **IChat** , **ICall** , **ICallChannel** , **IGroup** , **IFileTransfer**

---

Copyright  2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- u -

- [UILanguageChanged\(\)](#) : [\\_ISkypeEvents](#)
- [Unbookmark\(\)](#) : [IChat](#)
- [Upload\(\)](#) : [IVoicemail](#)
- [User](#) : [ISkype](#)
- [UserAuthorizationRequestReceived\(\)](#) : [\\_ISkypeEvents](#)
- [UserMood\(\)](#) : [\\_ISkypeEvents](#)
- [Users](#) : [IGroup](#) , [IChatMessage](#)
- [UserSexToText\(\)](#) : [IConversion](#)
- [UserStatus\(\)](#) : [\\_ISkypeEvents](#)
- [UserStatusToText\(\)](#) : [IConversion](#)
- [UsersWaitingAuthorization](#) : [ISkype](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- v -

- ValidatedSmsNumbers : [IProfile](#)
- Variable : [ISkype](#)
- Version : [ISkype](#)
- VideoIn : [ISettings](#)
- VideoReceiveStatus : [ICall](#)
- VideoSendStatus : [ICall](#)
- VideoStatus : [ICall](#)
- VmAllowedDuration : [ICall](#)
- VmDuration : [ICall](#)
- Voicemail : [ISkype](#)
- VoicemailFailureReasonToText() : [IConversion](#)
- Voicemails : [ISkype](#)
- VoicemailStatus() : [\\_ISkypeEvents](#)
- VoicemailStatusToText() : [IConversion](#)
- VoicemailTypeToText() : [IConversion](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																			
<a href="#">Files</a>	<a href="#">Examples</a>																					
<a href="#">Class List</a>	<a href="#">Class Members</a>																					
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																				
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

Here is a list of all documented class members with links to the class documentation for each member:

- w -

- Wallpaper : [IClient](#)
- WallpaperChanged() : [\\_ISkypeEvents](#)
- Write() : [IApplicationStream](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<b><a href="#">Classes</a></b>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<b><a href="#">Class Members</a></b>																		
<a href="#">All</a>	<b><a href="#">Functions</a></b>	<a href="#">Properties</a>																	
<a href="#">a</a>	<b><a href="#">b</a></b>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- b -

- [Bookmark\(\)](#) : **[IChat](#)**
- [BuddyStatusToText\(\)](#) : **[IConversion](#)**
- [ButtonPressed\(\)](#) : **[IClient](#)**
- [ButtonReleased\(\)](#) : **[IClient](#)**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>
<a href="#">Files</a>	<a href="#">Examples</a>		
<a href="#">Class List</a>	<a href="#">Class Members</a>		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>
<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>
<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>
<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>
<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- c -

- [CallDtmfReceived\(\)](#) : [\\_ISkypeEvents](#)
- [CallFailureReasonToText\(\)](#) : [IConversion](#)
- [CallHistory\(\)](#) : [\\_ISkypeEvents](#)
- [CallInputStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallSeenStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallStatus\(\)](#) : [\\_ISkypeEvents](#)
- [CallStatusToText\(\)](#) : [IConversion](#)
- [CallTransferStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallTypeToText\(\)](#) : [IConversion](#)
- [CallVideoReceiveStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallVideoSendStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallVideoSendStatusToText\(\)](#) : [IConversion](#)
- [CallVideoStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [CallVideoStatusToText\(\)](#) : [IConversion](#)
- [ChangeUserStatus\(\)](#) : [ISkype](#)
- [Channels\(\)](#) : [\\_ICallChannelManagerEvents](#)
- [ChatLeaveReasonToText\(\)](#) : [IConversion](#)
- [ChatMemberRoleChanged\(\)](#) : [\\_ISkypeEvents](#)
- [ChatMembersChanged\(\)](#) : [\\_ISkypeEvents](#)
- [ChatMessageStatusToText\(\)](#) : [IConversion](#)
- [ChatMessageTypeToText\(\)](#) : [IConversion](#)
- [ChatStatusToText\(\)](#) : [IConversion](#)
- [ClearCallHistory\(\)](#) : [ISkype](#)
- [ClearChatHistory\(\)](#) : [ISkype](#)

- ClearRecentMessages() : **IChat**
- ClearVoicemailHistory() : **ISkype**
- Command() : **\_ISkypeEvents**
- Connect() : **IApplication , ICallChannelManager**
- ConnectionStatus() : **\_ISkypeEvents**
- ConnectionStatusToText() : **IConversion**
- ContactsFocused() : **\_ISkypeEvents**
- Create() : **IApplication**
- CreateApplication() : **ICallChannelManager**
- CreateChatMultiple() : **ISkype**
- CreateChatUsingBlob() : **ISkype**
- CreateChatWith() : **ISkype**
- Created() : **\_ICallChannelManagerEvents**
- CreateEvent() : **IClient**
- CreateGroup() : **ISkype**
- CreateMenuItem() : **IClient**
- CreateSms() : **ISkype**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>		<a href="#">Modules</a>		<a href="#">Classes</a>													
<a href="#">Files</a>		<a href="#">Examples</a>																	
<a href="#">Class List</a>		<a href="#">Class Members</a>																	
<a href="#">All</a>		<a href="#">Functions</a>		<a href="#">Properties</a>															
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- d -

- [Decline\(\)](#) : [IGroup](#)
- [Delete\(\)](#) : [IVoicemail](#) , [ISmsMessage](#) , [IPluginMenuItem](#) , [IApplication](#) , [IPluginEvent](#)
- [DeleteGroup\(\)](#) : [ISkype](#)
- [Disband\(\)](#) : [IChat](#)
- [Disconnect\(\)](#) : [IApplicationStream](#) , [ICallChannelManager](#)
- [Download\(\)](#) : [IVoicemail](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- e -

- [EnableApiSecurityContext\(\)](#) : **ISkype**
- [EnterPassword\(\)](#) : **IChat**
- [Error\(\)](#) : **\_ISkypeEvents**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>			<a href="#">Modules</a>			<a href="#">Classes</a>											
<a href="#">Files</a>		<a href="#">Examples</a>																	
<a href="#">Class List</a>		<a href="#">Class Members</a>																	
<a href="#">All</a>		<a href="#">Functions</a>			<a href="#">Properties</a>														
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- f -

- [FileTransferStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [FindChatUsingBlob\(\)](#) : [ISkype](#)
- [Finish\(\)](#) : [IConference](#) , [ICall](#)
- [Focus\(\)](#) : [IClient](#)
- [Forward\(\)](#) : [ICall](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- g -

- [GroupDeleted\(\)](#) : [\\_ISkypeEvents](#)
- [GroupExpanded\(\)](#) : [\\_ISkypeEvents](#)
- [GroupTypeToText\(\)](#) : [IConversion](#)
- [GroupUsers\(\)](#) : [\\_ISkypeEvents](#)
- [GroupVisible\(\)](#) : [\\_ISkypeEvents](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- h -

- `Hold()` : [ICall](#) , [IConference](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- j -

- [Join\(\)](#) : [ICall](#) , [IChat](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- k -

- Kick() : **IChat**
- KickBan() : **IChat**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- | -

- [Leave\(\)](#) : [IChat](#)
- [LoadAvatarFromFile\(\)](#) : [ISettings](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- m -

- [Message\(\)](#) : [\\_ICallChannelManagerEvents](#)
- [MessageHistory\(\)](#) : [\\_ISkypeEvents](#)
- [MessageStatus\(\)](#) : [\\_ISkypeEvents](#)
- [Minimize\(\)](#) : [IClient](#)
- [Mute\(\)](#) : [\\_ISkypeEvents](#)



# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>													
<b>Files</b>		<b>Examples</b>																	
Class List		<b>Class Members</b>																	
All		<b>Functions</b>		Properties															
a	b	c	d	e	f	g	h	j	k	l	m	<b>o</b>	p	r	s	t	u	v	w

- o -

- [OnlineStatus\(\)](#) : **[\\_ISkypeEvents](#)**
- [OnlineStatusToText\(\)](#) : **[IConversion](#)**
- [Open\(\)](#) : **[IVoicemail](#)**
- [OpenAddContactDialog\(\)](#) : **[IClient](#)**
- [OpenAuthorizationDialog\(\)](#) : **[IClient](#)**
- [OpenBlockedUsersDialog\(\)](#) : **[IClient](#)**
- [OpenCallHistoryTab\(\)](#) : **[IClient](#)**
- [OpenConferenceDialog\(\)](#) : **[IClient](#)**
- [OpenContactsTab\(\)](#) : **[IClient](#)**
- [OpenDialog\(\)](#) : **[IClient](#)**
- [OpenDialpadTab\(\)](#) : **[IClient](#)**
- [OpenFileTransferDialog\(\)](#) : **[IClient](#)**
- [OpenGettingStartedWizard\(\)](#) : **[IClient](#)**
- [OpenImportContactsWizard\(\)](#) : **[IClient](#)**
- [OpenLiveTab\(\)](#) : **[IClient](#)**
- [OpenMessageDialog\(\)](#) : **[IClient](#)**
- [OpenOptionsDialog\(\)](#) : **[IClient](#)**
- [OpenProfileDialog\(\)](#) : **[IClient](#)**
- [OpenSearchDialog\(\)](#) : **[IClient](#)**
- [OpenSendContactsDialog\(\)](#) : **[IClient](#)**
- [OpenSmsDialog\(\)](#) : **[IClient](#)**
- [OpenUserInfoDialog\(\)](#) : **[IClient](#)**
- [OpenVideoTestDialog\(\)](#) : **[IClient](#)**
- [OpenWindow\(\)](#) : **[IChat](#)**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- p -

- [PlaceCall\(\)](#) : [ISkype](#)
- [PluginEventClicked\(\)](#) : [\\_ISkypeEvents](#)
- [PluginMenuItemClicked\(\)](#) : [\\_ISkypeEvents](#)

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>													
<b>Files</b>		<b>Examples</b>																	
Class List		<b>Class Members</b>																	
All		<b>Functions</b>		Properties															
a	b	c	d	e	f	g	h	j	k	l	m	o	p	<b>r</b>	s	t	u	v	w

- r -

- Read() : **IApplicationStream**
- RedirectToVoicemail() : **ICall**
- Remove() : **IParticipantCollection** , **IConferenceCollection** , **ICallChannelCollection** , **IVoicemailCollection** , **ICallCollection** , **IApplicationStreamCollection** , **IGroupCollection** , **IUserCollection** , **IChatMessageCollection** , **ISmsChunkCollection** , **ISmsTargetCollection** , **IChatCollection** , **ISmsMessageCollection** , **IFileTransferCollection**
- RemoveAll() : **IChatCollection** , **IFileTransferCollection** , **IUserCollection** , **IParticipantCollection** , **ICallCollection** , **ICallChannelCollection** , **IApplicationStreamCollection** , **IGroupCollection** , **ISmsTargetCollection** , **IChatMessageCollection** , **IConferenceCollection** , **ISmsChunkCollection** , **ISmsMessageCollection** , **IVoicemailCollection**
- RemoveUser() : **IGroup**
- Reply() : **\_ISkypeEvents**
- ResetCache() : **ISkype**
- ResetIdleTimer() : **ISettings**
- Resume() : **ICall** , **IConference**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- S -

- [SaveAvatarToFile\(\)](#) : [ISettings](#)
- [SearchForUsers\(\)](#) : [ISkype](#)
- [Send\(\)](#) : [ISmsMessage](#)
- [SendCommand\(\)](#) : [ISkype](#)
- [SendDatagram\(\)](#) : [IApplicationStream](#) , [IApplication](#)
- [SendMessage\(\)](#) : [ISkype](#) , [IChat](#)
- [SendSms\(\)](#) : [ISkype](#)
- [SendTextMessage\(\)](#) : [ICallChannel](#)
- [SendVoicemail\(\)](#) : [ISkype](#)
- [SetPassword\(\)](#) : [IChat](#)
- [SetUnplayed\(\)](#) : [IVoicemail](#)
- [Share\(\)](#) : [IGroup](#)
- [Shutdown\(\)](#) : [IClient](#)
- [SilentModeStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [SmsMessageStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [SmsMessageStatusToText\(\)](#) : [IConversion](#)
- [SmsMessageTypeToText\(\)](#) : [IConversion](#)
- [SmsTargetStatusChanged\(\)](#) : [\\_ISkypeEvents](#)
- [SmsTargetStatusToText\(\)](#) : [IConversion](#)
- [Start\(\)](#) : [IClient](#)
- [StartPlayback\(\)](#) : [IVoicemail](#)
- [StartPlaybackInCall\(\)](#) : [IVoicemail](#)
- [StartRecording\(\)](#) : [IVoicemail](#)
- [StartVideoReceive\(\)](#) : [ICall](#)

- StartVideoSend() : **ICall**
- StopPlayback() : **IVoicemail**
- StopRecording() : **IVoicemail**
- StopVideoReceive() : **ICall**
- StopVideoSend() : **ICall**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- t -

- [TextToAttachmentStatus\(\)](#) : **ICConversion**
- [TextToBuddyStatus\(\)](#) : **ICConversion**
- [TextToCallStatus\(\)](#) : **ICConversion**
- [TextToCallType\(\)](#) : **ICConversion**
- [TextToChatMessageStatus\(\)](#) : **ICConversion**
- [TextToChatMessageType\(\)](#) : **ICConversion**
- [TextToConnectionStatus\(\)](#) : **ICConversion**
- [TextToGroupType\(\)](#) : **ICConversion**
- [TextToOnlineStatus\(\)](#) : **ICConversion**
- [TextToUserSex\(\)](#) : **ICConversion**
- [TextToUserStatus\(\)](#) : **ICConversion**
- [TextToVoicemailStatus\(\)](#) : **ICConversion**
- [Transfer\(\)](#) : **ICall**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

## - u -

- [UILanguageChanged\(\)](#) : [\\_ISkypeEvents](#)
- [Unbookmark\(\)](#) : [IChat](#)
- [Upload\(\)](#) : [IVoicemail](#)
- [UserAuthorizationRequestReceived\(\)](#) : [\\_ISkypeEvents](#)
- [UserMood\(\)](#) : [\\_ISkypeEvents](#)
- [UserSexToText\(\)](#) : [IConversion](#)
- [UserStatus\(\)](#) : [\\_ISkypeEvents](#)
- [UserStatusToText\(\)](#) : [IConversion](#)



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- v -

- VoicemailFailureReasonToText() : **ICConversion**
- VoicemailStatus() : **\_ISkypeEvents**
- VoicemailStatusToText() : **ICConversion**
- VoicemailTypeToText() : **ICConversion**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																
<a href="#">Files</a>	<a href="#">Examples</a>																		
<a href="#">Class List</a>	<a href="#">Class Members</a>																		
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																	
<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">j</a>	<a href="#">k</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- w -

- [WallpaperChanged\(\)](#) : [\\_ISkypeEvents](#)
- [Write\(\)](#) : [IApplicationStream](#)

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>														
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		<b>Functions</b>		<b>Properties</b>																
-	<b>a</b>	b	c	d	e	f	g	h	i	l	m	n	o	p	r	s	t	u	v	w

- a -

- About : **IUser** , **IProfile**
- ActiveCalls : **ISkype** , **IConference**
- ActiveChats : **ISkype**
- ActiveFileTransfers : **ISkype**
- ActiveMembers : **IChat**
- ActivityTimestamp : **IChat**
- Adder : **IChat**
- AEC : **ISettings**
- AGC : **ISettings**
- AlertString : **IChat**
- Aliases : **IUser**
- AllowedDuration : **IVoicemail**
- ApiSecurityContextEnabled : **ISkype**
- ApiWrapperVersion : **ISkype**
- Applicants : **IChat**
- Application : **ISkype**
- ApplicationName : **IApplicationStream**
- AttachmentStatus : **ISkype**
- AudioIn : **ISettings**
- AudioOut : **ISettings**
- AutoAway : **ISettings**
- Avatar : **ISettings**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- b -

- Balance : [IProfile](#)
- BalanceCurrency : [IProfile](#)
- BalanceToText : [IProfile](#)
- Birthday : [IUser](#) , [IProfile](#)
- Blob : [IChat](#)
- Blocking : [ICommand](#)
- Body : [IChatMessage](#) , [ISmsMessage](#)
- Bookmarked : [IChat](#)
- BookmarkedChats : [ISkype](#)
- BuddyStatus : [IUser](#)
- BytesPerSecond : [IFileTransfer](#)
- BytesTransferred : [IFileTransfer](#)

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>																																	
<b>Files</b>		<b>Examples</b>																																					
Class List		<b>Class Members</b>																																					
All		<b>Functions</b>		<b>Properties</b>																																			
_		a		<b>c</b>		d		e		f		g		h		i		l		m		n		o		p		r		s		t		u		v		w	

- c -

- Cache : **ISkype**
- Call : **ISkype** , **ICallChannel**
- CallApplyCF : **IProfile**
- CallForwardRules : **IProfile**
- CallNoAnswerTimeout : **IProfile**
- Calls : **ISkype** , **IConference**
- CallSendToVM : **IProfile**
- CallStatus : **IParticipant**
- CallType : **IParticipant**
- CanLeaveVoicemail : **IUser**
- CanSetRoleTo : **IChatMember**
- CanTransfer : **ICall**
- Caption : **IPluginMenuItem**
- CaptureMicDevice : **ICall**
- Channels : **ICallChannelManager**
- ChannelType : **ICallChannelManager**
- CharactersLeft : **ISmsChunk**
- Chat : **IChatMessage** , **IChatMember** , **ISkype**
- ChatName : **IChatMessage**
- Chats : **ISkype**
- Chunks : **ISmsMessage**
- City : **IUser** , **IProfile**
- Client : **ISkype**
- Command : **ICommand** , **ISkype**

- CommandId : **ISkype**
- Conference : **ISkype**
- ConferencId : **ICall**
- Conferences : **ISkype**
- ConnectableUsers : **IApplication**
- ConnectingUsers : **IApplication**
- ConnectionStatus : **ISkype**
- Convert : **ISkype**
- Count : **ICallCollection** , **ICallChannelCollection** , **IApplicationStreamCollection** , **IConferenceCollection** , **IChatMemberCollection** , **IVoicemailCollection** , **IParticipantCollection** , **IChatMessageCollection** , **IChatCollection** , **IUserCollection** , **ISmsTargetCollection** , **IGroupCollection** , **ISmsMessageCollection** , **IFileTransferCollection** , **ISmsChunkCollection**
- Country : **IUser** , **IProfile**
- CountryCode : **IUser**
- Created : **ICallChannelManager**
- CurrentUser : **ISkype**
- CurrentUserHandle : **ISkype**
- CurrentUserProfile : **ISkype**
- CurrentUserStatus : **ISkype**
- CustomGroupId : **IGroup**
- CustomGroups : **ISkype**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>		<a href="#">Modules</a>		<a href="#">Classes</a>														
<a href="#">Files</a>		<a href="#">Examples</a>																		
<a href="#">Class List</a>		<a href="#">Class Members</a>																		
<a href="#">All</a>		<a href="#">Functions</a>		<a href="#">Properties</a>																
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- d -

- DataLength : **IApplicationStream**
- Description : **IChat**
- DialogPartner : **IChat**
- DisplayName : **IUser** , **IParticipant** , **IGroup**
- DTMF : **ICall**
- Duration : **ICall** , **Voicemail**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- e -

- EditedBy : **IChatMessage**
- EditedTimestamp : **IChatMessage**
- Enabled : **IPluginMenuItem**
- Expected : **ICommand**



# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>			<b>Modules</b>			<b>Classes</b>												
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		Functions			<b>Properties</b>															
-	a	b	c	d	e	<b>f</b>	g	h	i	l	m	n	o	p	r	s	t	u	v	w

- f -

- FailureReason : **ICall** , **IVoicemail** , **IFileTransfer** , **ISmsMessage**
- FileName : **IFileTransfer**
- FilePath : **IFileTransfer**
- FileTransfers : **ISkype**
- FinishTime : **IFileTransfer**
- FocusedContacts : **ISkype**
- ForwardedBy : **ICall**
- FriendlyName : **IChat** , **ISkype**
- Friends : **ISkype**
- FromDisplayName : **IChatMessage**
- FromHandle : **IChatMessage**
- FullName : **IProfile** , **IUser**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- g -

- Greeting : **ISkype**
- Groups : **ISkype**
- GuideLines : **IChat**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>		<a href="#">Modules</a>		<a href="#">Classes</a>														
<a href="#">Files</a>		<a href="#">Examples</a>																		
<a href="#">Class List</a>		<a href="#">Class Members</a>																		
<a href="#">All</a>		<a href="#">Functions</a>		<a href="#">Properties</a>																
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- h -

- Handle : [IUser](#) , [IParticipant](#) , [IApplicationStream](#) , [IChatMember](#)
- HardwiredGroups : [ISkype](#)
- HasCallEquipment : [IUser](#)
- Hint : [IPluginMenuItem](#)
- Homepage : [IUser](#) , [IProfile](#)

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>														
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		<b>Functions</b>		<b>Properties</b>																
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- i -

- Id : **ICommand** , **ICall** , **IChatMember** , **ISmsChunk** , **ISmsMessage** , **IConference** , **IPluginMenuItem** , **IPluginEvent** , **IChatMessage** , **Voicemail** , **IFileTransfer** , **IGroup**
- InputDevice : **ICall**
- InputStatus : **ICall**
- IPCountry : **IProfile**
- IsActive : **IChatMember**
- IsAuthorized : **IUser**
- IsBlocked : **IUser**
- IsCallForwardActive : **IUser**
- IsEditable : **IChatMessage**
- IsExpanded : **IGroup**
- IsFailedUnseen : **ISmsMessage**
- IsRunning : **IClient**
- IsSkypeOutContact : **IUser**
- IsVideoCapable : **IUser**
- IsVisible : **IGroup**
- IsVoicemailCapable : **IUser**
- Item : **ICallChannelCollection** , **IChatMessageCollection** , **IChatCollection** , **ISmsTargetCollection** , **IParticipantCollection** , **VoicemailCollection** , **IChatMemberCollection** , **IConferenceCollection** , **IApplicationStreamCollection** , **ICallCollection** , **IUserCollection** , **ISmsChunkCollection** , **IGroupCollection** , **IFileTransferCollection** ,

## ISmsMessageCollection

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- l -

- Language : **IUser** , **ICConversion** , **ISettings**
- LanguageCode : **IUser**
- Languages : **IProfile**
- LastOnline : **IUser**
- LeaveReason : **IChatMessage**

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>			<b>Modules</b>			<b>Classes</b>												
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		Functions			<b>Properties</b>															
-	a	b	c	d	e	f	g	h	i	l	<b>m</b>	n	o	p	r	s	t	u	v	w

- m -

- Manager : **ICallChannel**
- MemberObjects : **IChat**
- Members : **IChat**
- Message : **ISmsChunk** , **ISmsTarget** , **ISkype**
- Messages : **ISkype** , **IChat**
- MissedCalls : **ISkype**
- MissedChats : **ISkype**
- MissedMessages : **ISkype**
- MissedSmss : **ISkype**
- MissedVoicemails : **ISkype**
- MoodText : **IProfile** , **IUser**
- Mute : **ISkype**
- MyRole : **IChat**
- MyStatus : **IChat**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- n -

- Name : **IChat** , **IApplication** , **ICallChannelManager**
- Number : **ISmsTarget**
- NumberOfAuthBuddies : **IUser**

---

Copyright © 2006 Skype Limited. All rights reserved.



# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- 0 -

- OnlineStatus : **IUser**
- OnlineUsers : **IGroup**
- Options : **IChat** , **IChatMessage**
- OutputDevice : **ICall**

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>		<b>Modules</b>		<b>Classes</b>														
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		<b>Functions</b>		<b>Properties</b>																
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<b><a href="#">p</a></b>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- p -

- Participants : **ICall**
- PartnerDisplayName : **ICall** , **IFileTransfer** , **IVoicemail**
- PartnerHandle : **IFileTransfer** , **ICall** , **IVoicemail** , **IApplicationStream**
- PasswordHint : **IChat**
- PCSpeaker : **ISettings**
- PhoneHome : **IUser** , **IProfile**
- PhoneMobile : **IUser** , **IProfile**
- PhoneOffice : **IProfile** , **IUser**
- Posters : **IChat**
- PredictiveDialerCountry : **ISkype**
- Price : **ISmsMessage**
- PriceCurrency : **ISmsMessage**
- PricePrecision : **ISmsMessage**
- Privilege : **ISkype**
- Profile : **ISkype**
- Property : **ISkype**
- Protocol : **ISkype**
- Province : **IProfile** , **IUser**
- PstnNumber : **ICall**
- PstnStatus : **ICall**

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>			<b>Modules</b>			<b>Classes</b>												
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		Functions			<b>Properties</b>															
-	a	b	c	d	e	f	g	h	i	l	m	n	o	p	<b>r</b>	s	t	u	v	w

- r -

- Rate : **ICall**
- RateCurrency : **ICall**
- RatePrecision : **ICall**
- ReceivedAuthRequest : **IUser**
- ReceivedStreams : **IApplication**
- RecentChats : **ISkype**
- RecentMessages : **IChat**
- Reply : **ICommand**
- ReplyToNumber : **ISmsMessage**
- RichMoodText : **IUser** , **IProfile**
- Ringer : **ISettings**
- RingTone : **ISettings**
- RingToneStatus : **ISettings**
- Role : **IChatMessage** , **IChatMember**

# Skype4COM 1.0.36.0

<b>Main Page</b>		<b>Related Pages</b>			<b>Modules</b>			<b>Classes</b>												
<b>Files</b>		<b>Examples</b>																		
Class List		<b>Class Members</b>																		
All		Functions			<b>Properties</b>															
-	a	b	c	d	e	f	g	h	i	l	m	n	o	p	r	<b>s</b>	t	u	v	w

- s -

- Seen : **ICall** , **IChatMessage** , **ISmsMessage**
- Sender : **IChatMessage**
- SendingStreams : **IApplication**
- Settings : **ISkype**
- Sex : **IProfile** , **IUser**
- SilentMode : **ISkype**
- Smss : **ISkype**
- SpeedDial : **IUser**
- StartTime : **IFileTransfer**
- Status : **IChat** , **ISmsTarget** , **IVoicemail** , **IFileTransfer** , **ISmsMessage** , **IChatMessage** , **ICall**
- Stream : **ICallChannel**
- Streams : **IApplication**
- Subject : **ICall**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>		<a href="#">Related Pages</a>		<a href="#">Modules</a>		<a href="#">Classes</a>														
<a href="#">Files</a>		<a href="#">Examples</a>																		
<a href="#">Class List</a>		<a href="#">Class Members</a>																		
<a href="#">All</a>		<a href="#">Functions</a>		<a href="#">Properties</a>																
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- t -

- TargetIdentity : [ICall](#)
- TargetNumbers : [ISmsMessage](#)
- Targets : [ISmsMessage](#)
- Text : [ISmsChunk](#) , [ICallChannelMessage](#)
- Timeout : [ICommand](#) , [ISkype](#)
- Timestamp : [ICall](#) , [IChatMessage](#) , [IChat](#) , [IVoicemail](#) , [ISmsMessage](#)
- Timezone : [IUser](#) , [IProfile](#)
- Topic : [IChat](#)
- TopicXML : [IChat](#)
- TransferActive : [ICall](#)
- TransferredBy : [ICall](#)
- TransferredTo : [ICall](#)
- TransferStatus : [ICall](#)
- Type : [IVoicemail](#) , [IChat](#) , [ICallChannel](#) , [IGroup](#) , [ICall](#) , [IFileTransfer](#) , [IChatMessage](#) , [ISmsMessage](#)

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- u -

- User : **ISkype**
- Users : **IChatMessage** , **IGroup**
- UsersWaitingAuthorization : **ISkype**

---

Copyright © 2006 Skype Limited. All rights reserved.

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- v -

- ValidatedSmsNumbers : **IProfile**
- Variable : **ISkype**
- Version : **ISkype**
- VideoIn : **ISettings**
- VideoReceiveStatus : **ICall**
- VideoSendStatus : **ICall**
- VideoStatus : **ICall**
- VmAllowedDuration : **ICall**
- VmDuration : **ICall**
- Voicemail : **ISkype**
- Voicemails : **ISkype**

# Skype4COM 1.0.36.0

<a href="#">Main Page</a>	<a href="#">Related Pages</a>	<a href="#">Modules</a>	<a href="#">Classes</a>																	
<a href="#">Files</a>	<a href="#">Examples</a>																			
<a href="#">Class List</a>	<a href="#">Class Members</a>																			
<a href="#">All</a>	<a href="#">Functions</a>	<a href="#">Properties</a>																		
<a href="#">_</a>	<a href="#">a</a>	<a href="#">b</a>	<a href="#">c</a>	<a href="#">d</a>	<a href="#">e</a>	<a href="#">f</a>	<a href="#">g</a>	<a href="#">h</a>	<a href="#">i</a>	<a href="#">l</a>	<a href="#">m</a>	<a href="#">n</a>	<a href="#">o</a>	<a href="#">p</a>	<a href="#">r</a>	<a href="#">s</a>	<a href="#">t</a>	<a href="#">u</a>	<a href="#">v</a>	<a href="#">w</a>

- w -

- Wallpaper : [IClient](#)