

TrueConf SDK for Windows

TrueConf CallX

2.0

© 2016-2019. TrueConf LLC. All rights reserved.



# Contents

- 1 Namespace Index** **1**
  - 1.1 Namespace List 1
  
- 2 Hierarchical Index** **3**
  - 2.1 Class Hierarchy 3
  
- 3 Class Index** **5**
  - 3.1 Class List 5
  
- 4 File Index** **7**
  - 4.1 File List 7
  
- 5 Namespace Documentation** **9**
  - 5.1 TrueConf\_CallXLib Module Reference 9
    - 5.1.1 Function Documentation 9
      - 5.1.1.1 importlib() 9
  
- 6 Class Documentation** **11**
  - 6.1 TrueConf\_CallXLib::\_ITrueConfCallXEvents Interface Reference 11
    - 6.1.1 Member Function Documentation 13
      - 6.1.1.1 OnAbookUpdate() 13
      - 6.1.1.2 OnAppUpdateAvailable() 13
      - 6.1.1.3 OnAudioCapturerRmsLevelUpdated() 13
      - 6.1.1.4 OnBroadcastPictureStateChanged() 14
      - 6.1.1.5 OnCallHistoryUpdated() 14
      - 6.1.1.6 OnChangeVideoMatrixReport() 14

---

6.1.1.7	OnChatMessageSent()	14
6.1.1.8	OnCmdAddToAbook()	15
6.1.1.9	OnCmdAddToGroup()	15
6.1.1.10	OnCmdBlock()	15
6.1.1.11	OnCmdChatClear()	16
6.1.1.12	OnCmdCreateGroup()	16
6.1.1.13	OnCmdRemoveFromAbook()	16
6.1.1.14	OnCmdRemoveFromGroup()	16
6.1.1.15	OnCmdRemoveGroup()	17
6.1.1.16	OnCmdRenameGroup()	17
6.1.1.17	OnCmdRenameInAbook()	17
6.1.1.18	OnCmdUnblock()	18
6.1.1.19	OnCommandReceived()	18
6.1.1.20	OnCommandSent()	18
6.1.1.21	OnConferenceCreated()	19
6.1.1.22	OnConferenceDeleted()	19
6.1.1.23	OnContactBlocked()	19
6.1.1.24	OnContactDeleted()	19
6.1.1.25	OnContactUnblocked()	19
6.1.1.26	OnDetailInfo()	20
6.1.1.27	OnDeviceModesDone()	20
6.1.1.28	OnFileAccepted()	20
6.1.1.29	OnFileConferenceSent()	20
6.1.1.30	OnFileRejected()	20
6.1.1.31	OnFileSent()	21
6.1.1.32	OnFileTransferAvailable()	21
6.1.1.33	OnGroupChatMessageSent()	21
6.1.1.34	OnGroupsUpdate()	22
6.1.1.35	OnHangUpPressed()	22
6.1.1.36	OnHardwareChanged()	22

---

6.1.1.37	OnIncomingChatMessage()	22
6.1.1.38	OnIncomingGroupChatMessage()	23
6.1.1.39	OnIncomingRequestToPodiumAnswered()	23
6.1.1.40	OnInviteReceived()	23
6.1.1.41	OnInviteRequestSent()	23
6.1.1.42	OnInviteSent()	23
6.1.1.43	OnJabraHangUpPressed()	24
6.1.1.44	OnJabraHookOffPressed()	24
6.1.1.45	OnLogin()	24
6.1.1.46	OnLogout()	24
6.1.1.47	OnOffHookPressed()	24
6.1.1.48	OnReceiversInfoUpdated()	24
6.1.1.49	OnRecordRequest()	24
6.1.1.50	OnRecordRequestReply()	25
6.1.1.51	OnRejectReceived()	25
6.1.1.52	OnRejectSent()	25
6.1.1.53	OnRemarkCountDown()	25
6.1.1.54	OnRequestInviteReceived()	25
6.1.1.55	OnRestrictionsChanged()	25
6.1.1.56	OnRoleChanged()	25
6.1.1.57	OnSelfSSInfoUpdate()	26
6.1.1.58	OnServerConnected()	26
6.1.1.59	OnServerDisconnected()	26
6.1.1.60	OnSettingsChanged()	26
6.1.1.61	OnSlideShowInfoUpdate()	26
6.1.1.62	OnSlideShowStart()	26
6.1.1.63	OnSlideShowStop()	27
6.1.1.64	OnStart()	27
6.1.1.65	OnStopCalling()	27
6.1.1.66	OnTestAudioCapturerStateUpdated()	27

---

6.1.1.67	OnToneDial()	27
6.1.1.68	OnUpdateAvatar()	28
6.1.1.69	OnUpdateCameraInfo()	28
6.1.1.70	OnUpdateParticipantList()	28
6.1.1.71	OnVideoMatrixChanged()	28
6.1.1.72	OnXAfterStart()	28
6.1.1.73	OnXChangeState()	28
6.1.1.74	OnXCommandExecution()	29
6.1.1.75	OnXError()	29
6.1.1.76	OnXFileReceive()	29
6.1.1.77	OnXFileReceiveProgress()	30
6.1.1.78	OnXFileRequestReceived()	30
6.1.1.79	OnXFileSend()	30
6.1.1.80	OnXFileSendError()	30
6.1.1.81	OnXFileStatusChange()	30
6.1.1.82	OnXLogin()	31
6.1.1.83	OnXLoginError()	31
6.1.1.84	OnXNotify()	32
6.1.1.85	OnXStartFail()	32
6.1.1.86	OnXTerminate()	32
6.2	ITrueConfCallX Interface Reference	32
6.2.1	Member Function Documentation	36
6.2.1.1	accept()	36
6.2.1.2	acceptPeer()	36
6.2.1.3	acceptPodiumInvite()	36
6.2.1.4	acceptPodiumRequest()	37
6.2.1.5	acceptPodiumRequest2()	37
6.2.1.6	activateLicense()	37
6.2.1.7	addContactToAbook()	37
6.2.1.8	addToAbook()	38

---

6.2.1.9	<a href="#">addToGroup()</a>	38
6.2.1.10	<a href="#">allowRecord()</a>	38
6.2.1.11	<a href="#">allowRecord2()</a>	39
6.2.1.12	<a href="#">appUpdate()</a>	39
6.2.1.13	<a href="#">block()</a>	39
6.2.1.14	<a href="#">call()</a>	40
6.2.1.15	<a href="#">changeVideoMatrix()</a>	40
6.2.1.16	<a href="#">chatClear()</a>	40
6.2.1.17	<a href="#">clearSlides()</a>	41
6.2.1.18	<a href="#">connectToServer()</a>	41
6.2.1.19	<a href="#">createConference()</a>	41
6.2.1.20	<a href="#">createGroup()</a>	42
6.2.1.21	<a href="#">extendUidTtl()</a>	42
6.2.1.22	<a href="#">firstSlide()</a>	42
6.2.1.23	<a href="#">forbidRecord()</a>	42
6.2.1.24	<a href="#">forbidRecord2()</a>	42
6.2.1.25	<a href="#">getAbook()</a>	43
6.2.1.26	<a href="#">getAppState()</a>	43
6.2.1.27	<a href="#">getAudioMute()</a>	43
6.2.1.28	<a href="#">getBanList()</a>	43
6.2.1.29	<a href="#">getCallHistory()</a>	43
6.2.1.30	<a href="#">getChatLastMessages()</a>	44
6.2.1.31	<a href="#">getContactDetails()</a>	44
6.2.1.32	<a href="#">getDisplayNameById()</a>	44
6.2.1.33	<a href="#">getGroups()</a>	45
6.2.1.34	<a href="#">getHardware()</a>	45
6.2.1.35	<a href="#">getHardwareKey()</a>	45
6.2.1.36	<a href="#">getIdListRequestInvite()</a>	45
6.2.1.37	<a href="#">getLicenseType()</a>	45
6.2.1.38	<a href="#">getListOfChats()</a>	46

---

---

6.2.1.39	<code>getMaxConfTitleLength()</code>	46
6.2.1.40	<code>getMicMute()</code>	46
6.2.1.41	<code>getModes()</code>	46
6.2.1.42	<code>getMonitorsInfo()</code>	46
6.2.1.43	<code>getParticipantsList()</code>	46
6.2.1.44	<code>getSelfieBroadcast()</code>	47
6.2.1.45	<code>getSettings()</code>	47
6.2.1.46	<code>getSlideShowInfo()</code>	47
6.2.1.47	<code>getSystemInfo()</code>	47
6.2.1.48	<code>getVideoMatrix()</code>	47
6.2.1.49	<code>getVideoMute()</code>	47
6.2.1.50	<code>gotoPodium()</code>	47
6.2.1.51	<code>hangUp()</code>	48
6.2.1.52	<code>hangUp2()</code>	48
6.2.1.53	<code>inviteToConference()</code>	48
6.2.1.54	<code>inviteToPodium()</code>	49
6.2.1.55	<code>lastSlide()</code>	49
6.2.1.56	<code>leavePodium()</code>	49
6.2.1.57	<code>login()</code>	49
6.2.1.58	<code>logout()</code>	50
6.2.1.59	<code>moveSlideDown()</code>	50
6.2.1.60	<code>moveSlideUp()</code>	50
6.2.1.61	<code>nextSlide()</code>	50
6.2.1.62	<code>prevSlide()</code>	50
6.2.1.63	<code>ptzDown()</code>	51
6.2.1.64	<code>ptzLeft()</code>	51
6.2.1.65	<code>ptzRight()</code>	51
6.2.1.66	<code>ptzStop()</code>	51
6.2.1.67	<code>ptzUp()</code>	51
6.2.1.68	<code>ptzZoomIn()</code>	51



---

6.2.1.69	ptzZoomOut()	51
6.2.1.70	reject()	52
6.2.1.71	rejectInvitationToPodium()	52
6.2.1.72	rejectPeer()	52
6.2.1.73	rejectRequestToPodium()	52
6.2.1.74	rejectRequestToPodium2()	52
6.2.1.75	removeContactFromAbook()	53
6.2.1.76	removeFromAbook()	53
6.2.1.77	removeFromGroup()	53
6.2.1.78	removeFromPodium()	54
6.2.1.79	removeGroup()	54
6.2.1.80	removePeer()	54
6.2.1.81	renameGroup()	55
6.2.1.82	renameInAbook()	55
6.2.1.83	sendCommand()	56
6.2.1.84	sendGroupMessage()	56
6.2.1.85	sendMessage()	56
6.2.1.86	setAudioMute()	57
6.2.1.87	setHardware()	57
6.2.1.88	setLastCallsViewed()	57
6.2.1.89	setMicMute()	57
6.2.1.90	setModes()	57
6.2.1.91	setSelfieBroadcast()	58
6.2.1.92	setSettings()	58
6.2.1.93	setVideoMute()	58
6.2.1.94	showSlide()	58
6.2.1.95	shutdown()	58
6.2.1.96	shutdown2()	58
6.2.1.97	startAudioRemark()	59
6.2.1.98	startScreenCapture()	59

---

6.2.1.99 startSlideShow()	59
6.2.1.100 stopScreenCapture()	59
6.2.1.101 stopSlideShow()	60
6.2.1.102 switchVideoStream()	60
6.2.1.103 takePicture()	60
6.2.1.104 testAudioCapterStart()	60
6.2.1.105 testAudioCapterStop()	61
6.2.1.106 testAudioRenderer()	61
6.2.1.107 toneDial()	61
6.2.1.108 turnRemoteCamera()	61
6.2.1.109 turnRemoteMic()	62
6.2.1.110 unblock()	62
6.2.1.111 XAddSlide()	62
6.2.1.112 XAddSlidesDialog()	63
6.2.1.113 XCheckForEvent()	63
6.2.1.114 XDeselectCamera()	63
6.2.1.115 XDeselectMic()	64
6.2.1.116 XDeselectSpeaker()	64
6.2.1.117 XEncryptedLogin()	64
6.2.1.118 XExecuteCommand()	64
6.2.1.119 XFileAccept()	65
6.2.1.120 XFileDelete()	65
6.2.1.121 XFileReject()	65
6.2.1.122 XFileSend()	66
6.2.1.123 XFileSendAbort()	66
6.2.1.124 XFileSendToConference()	66
6.2.1.125 XFileStatsInfo()	67
6.2.1.126 XFileTransferAvailable()	67
6.2.1.127 XForceRestart()	67
6.2.1.128 XGetCallXProcessId()	67

6.2.1.129 XGetCameraByIndex()	67
6.2.1.130 XGetCameraCount()	68
6.2.1.131 XGetCameraList()	68
6.2.1.132 XGetFileDownloadPath()	68
6.2.1.133 XGetFileHashById()	69
6.2.1.134 XGetFileIdByHash()	69
6.2.1.135 XGetFileSendStatus()	69
6.2.1.136 XGetFileTransferList()	69
6.2.1.137 XGetHandle()	70
6.2.1.138 XGetMicByIndex()	70
6.2.1.139 XGetMicCount()	70
6.2.1.140 XGetMicList()	71
6.2.1.141 XGetSpeakerByIndex()	71
6.2.1.142 XGetSpeakerCount()	71
6.2.1.143 XGetSpeakerList()	72
6.2.1.144 XHangUpForAll()	72
6.2.1.145 XPrintPdf()	72
6.2.1.146 XRemotePrintPdf()	72
6.2.1.147 XRemoveSlide()	73
6.2.1.148 XRestart()	73
6.2.1.149 XSelectCamera()	73
6.2.1.150 XSelectMicByIndex()	74
6.2.1.151 XSelectSpeakerByIndex()	74
6.2.1.152 XSetBackgroundImage()	74
6.2.1.153 XSetCameraByIndex()	75
6.2.1.154 XStartPictureBroadcast()	75
6.2.1.155 XStopPictureBroadcast()	75
6.2.2 Property Documentation	76
6.2.2.1 CallXPath	76
6.2.2.2 Camera	76
6.2.2.3 Microphone	76
6.2.2.4 Speaker	76
6.2.2.5 XCommandLine	76
6.3 TrueConf_CallXLib::TrueConfCallX Class Reference	76
<b>7 File Documentation</b>	<b>77</b>
7.1 TrueConf_CallX.IDL File Reference	77
<b>Index</b>	<b>79</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[TrueConf\\_CallXLib](#) . . . . . 9



# Chapter 2

## Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

TrueConf_CallXLib::_ITrueConfCallXEvents . . . . .	11
IDispatch	
ITrueConfCallX . . . . .	32
TrueConf_CallXLib::TrueConfCallX . . . . .	76





# Chapter 3

## Class Index

### 3.1 Class List

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

<a href="#">TrueConf_CallXLib::_ITrueConfCallXEvents</a> . . . . .	11
<a href="#">ITrueConfCallX</a> . . . . .	32
<a href="#">TrueConf_CallXLib::TrueConfCallX</a> . . . . .	76



# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">TrueConf_CallX.IDL</a> . . . . .	77
--	----



## Chapter 5

# Namespace Documentation

### 5.1 TrueConf\_CallXLib Module Reference

#### Classes

- interface [\\_ITrueConfCallXEvents](#)
- class [TrueConfCallX](#)

#### Functions

- [importlib](#) ("stdole2.tlb")

#### 5.1.1 Function Documentation

##### 5.1.1.1 importlib()

```
TrueConf_CallXLib::importlib (
    "stdole2.tlb" )
```



# Chapter 6

## Class Documentation

### 6.1 TrueConf\_CallXLib::\_ITrueConfCallXEvents Interface Reference

```
import "TrueConf_CallX.IDL";
```

#### Public Member Functions

- void [OnXNotify](#) ([in] BSTR data)
- void [OnXAfterStart](#) ()
- void [OnXChangeState](#) ([in] LONG prevState, [in] LONG newState)
- void [OnXTerminate](#) ()
- void [OnXStartFail](#) ()
- void [OnAbookUpdate](#) ([in] BSTR eventDetails)
- void [OnAppUpdateAvailable](#) ([in] BSTR eventDetails)
- void [OnChangeVideoMatrixReport](#) ([in] BSTR eventDetails)
- void [OnConferenceCreated](#) ([in] BSTR eventDetails)
- void [OnConferenceDeleted](#) ([in] BSTR eventDetails)
- void [OnContactBlocked](#) ([in] BSTR eventDetails)
- void [OnContactDeleted](#) ([in] BSTR eventDetails)
- void [OnContactUnblocked](#) ([in] BSTR eventDetails)
- void [OnHardwareChanged](#) ([in] BSTR eventDetails)
- void [OnDetailInfo](#) ([in] BSTR eventDetails)
- void [OnDeviceModesDone](#) ([in] BSTR eventDetails)
- void [OnInviteReceived](#) ([in] BSTR eventDetails)
- void [OnIncomingRequestToPodiumAnswered](#) ([in] BSTR eventDetails)
- void [OnInviteRequestSent](#) ([in] BSTR eventDetails)
- void [OnInviteSent](#) ([in] BSTR eventDetails)
- void [OnLogin](#) ([in] BSTR eventDetails)
- void [OnLogout](#) ([in] BSTR eventDetails)
- void [OnReceiversInfoUpdated](#) ([in] BSTR eventDetails)
- void [OnRecordRequest](#) ([in] BSTR eventDetails)
- void [OnRecordRequestReply](#) ([in] BSTR eventDetails)
- void [OnRejectReceived](#) ([in] BSTR eventDetails)
- void [OnRejectSent](#) ([in] BSTR eventDetails)
- void [OnRemarkCountDown](#) ([in] BSTR eventDetails)
- void [OnRequestInviteReceived](#) ([in] BSTR eventDetails)
- void [OnRoleChanged](#) ([in] BSTR eventDetails)

- void [OnSelfSSInfoUpdate](#) ([in] BSTR eventDetails)
- void [OnServerConnected](#) ([in] BSTR eventDetails)
- void [OnServerDisconnected](#) ([in] BSTR eventDetails)
- void [OnSettingsChanged](#) ([in] BSTR eventDetails)
- void [OnSlideShowStart](#) ([in] BSTR eventDetails)
- void [OnSlideShowStop](#) ([in] BSTR eventDetails)
- void [OnStopCalling](#) ([in] BSTR eventDetails)
- void [OnUpdateAvatar](#) ([in] BSTR eventDetails)
- void [OnUpdateCameraInfo](#) ([in] BSTR eventDetails)
- void [OnUpdateParticipantList](#) ([in] BSTR eventDetails)
- void [OnRestrictionsChanged](#) ([in] BSTR eventDetails)
- void [OnVideoMatrixChanged](#) ([in] BSTR eventDetails)
- void [OnOffHookPressed](#) ([in] BSTR eventDetails)
- void [OnHangUpPressed](#) ([in] BSTR eventDetails)
- void [OnJabraHookOffPressed](#) ()
- void [OnJabraHangUpPressed](#) ()
- void [OnXCommandExecution](#) ([in] BSTR cmdName, [in] BSTR allData)
- void [OnSlideShowInfoUpdate](#) ([in] BSTR eventDetails)
- void [OnStart](#) ()
- void [OnXError](#) ([in] LONG errorCode, [in] BSTR errorMsg)
- void [OnXLogin](#) ()
- void [OnXLoginError](#) ([in] LONG errorCode)
- HRESULT [OnXFileRequestReceived](#) ([in] LONG fileId, [in] BSTR fileName, [in] BSTR fileCaption, [in] BSTR peerId)
- HRESULT [OnXFileStatusChange](#) ([in] LONG fileId, [in] LONG fileStatus, [in] LONG directionType)
- HRESULT [OnXFileSendError](#) ([in] LONG error\_code, [in] LONG fileId, [in] BSTR filePath, [in] BSTR fileCaption)
- HRESULT [OnXFileReceiveProgress](#) ([in] LONG percent, [in] LONG fileId, [in] BSTR fileName, [in] BSTR fileCaption)
- HRESULT [OnXFileReceive](#) ([in] BSTR peerId, [in] LONG fileId, [in] BSTR fileName, BSTR fileCaption)
- HRESULT [OnXFileSend](#) ([in] LONG fileId, [in] BSTR filePath, [in] BSTR fileCaption)
- HRESULT [OnCommandReceived](#) ([in] BSTR peerId, [in] BSTR command)
- void [OnBroadcastPictureStateChanged](#) ([in] BSTR filename)
- void [OnCallHistoryUpdated](#) ([in] BSTR jsonCallHistory)
- void [OnCmdAddToAbook](#) ([in] BSTR peerId, [in] BSTR peerDn)
- void [OnCmdAddToGroup](#) ([in] LONG groupId, [in] BSTR peerId)
- void [OnCmdBlock](#) ([in] BSTR peerId)
- void [OnCmdChatClear](#) ([in] BSTR chatId)
- void [OnCmdCreateGroup](#) ([in] BSTR groupName)
- void [OnCmdRemoveFromAbook](#) ([in] BSTR peerId)
- void [OnCmdRemoveFromGroup](#) ([in] LONG groupId, [in] BSTR peerId)
- void [OnCmdRemoveGroup](#) ([in] LONG groupId)
- void [OnCmdRenameGroup](#) ([in] LONG groupId, [in] BSTR groupName)
- void [OnCmdRenameInAbook](#) ([in] BSTR peerId, [in] BSTR peerDn)
- void [OnCmdUnblock](#) ([in] BSTR peerId)
- void [OnCommandSent](#) ([in] BSTR peerId, [in] BSTR command)
- void [OnFileAccepted](#) ([in] LONG fileId)
- void [OnFileConferenceSent](#) ([in] LONG fileId, [in] BSTR fileName)
- void [OnFileRejected](#) ([in] LONG fileId)
- void [OnFileSent](#) ([in] LONG fileId, [in] BSTR fileName, [in] BSTR peerId)
- void [OnFileTransferAvailable](#) ([in] VARIANT\_BOOL available)
- void [OnGroupsUpdate](#) ([in] BSTR jsonGroupList)
- void [OnIncomingChatMessage](#) ([in] BSTR peerId, [in] BSTR peerDn, [in] BSTR message, [in] ULONGLONG time)



- void [OnIncomingGroupChatMessage](#) ([in] BSTR peerId, [in] BSTR peerDn, [in] BSTR message, [in] ULONGLONG time)
- void [OnGroupChatMessageSent](#) ([in] BSTR message)
- void [OnChatMessageSent](#) ([in] BSTR peerId, [in] BSTR message)
- void [OnTestAudioCatcherStateUpdated](#) ([in] VARIANT\_BOOL started)
- void [OnAudioCatcherRmsLevelUpdated](#) ([in] DOUBLE lvl)
- void [OnToneDial](#) ([in] BSTR symbol)

## 6.1.1 Member Function Documentation

### 6.1.1.1 OnAbookUpdate()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnAbookUpdate (
    [in] BSTR eventDetails )
```

Address book update notification

#### Parameters

<i>eventDetails</i>	- event details in JSON format
---------------------	--------------------------------

### 6.1.1.2 OnAppUpdateAvailable()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnAppUpdateAvailable (
    [in] BSTR eventDetails )
```

Applicaton update available notification

### 6.1.1.3 OnAudioCatcherRmsLevelUpdated()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnAudioCatcherRmsLevelUpdated (
    [in] DOUBLE lvl )
```

Microphone test volume level notification

#### Parameters

<i>lvl</i>	- volume level
------------	----------------

#### 6.1.1.4 OnBroadcastPictureStateChanged()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnBroadcastPictureStateChanged (
    [in] BSTR filename )
```

Notification about video replacing picture send status change

##### Parameters

<i>picture</i>	filename without path, empty when not sending
----------------	---

#### 6.1.1.5 OnCallHistoryUpdated()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCallHistoryUpdated (
    [in] BSTR jsonCallHistory )
```

Call history change notification

##### Parameters

<i>jsonCallHistory</i>	- string with call history in JSON format. Contains only calls later than in lastView parameter
------------------------	---

#### 6.1.1.6 OnChangeVideoMatrixReport()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnChangeVideoMatrixReport (
    [in] BSTR eventDetails )
```

Video matrix change notification

#### 6.1.1.7 OnChatMessageSent()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnChatMessageSent (
    [in] BSTR peerId,
    [in] BSTR message )
```

Message sent to user notification

##### Parameters

<i>peerId</i>	- TrueConf ID
<i>message</i>	- message text

## 6.1.1.8 OnCmdAddToAbook()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdAddToAbook (
    [in] BSTR peerId,
    [in] BSTR peerDn )
```

**Deprecated** Notification about outgoing add to address book command

## Parameters

<i>peerId</i>	- user's TrueConf ID
<i>peerDn</i>	- user's display name

## 6.1.1.9 OnCmdAddToGroup()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdAddToGroup (
    [in] LONG groupId,
    [in] BSTR peerId )
```

**Deprecated** Notification about outgoing add to address book's group command

## Parameters

<i>groupId</i>	- group ID
<i>peerId</i>	- TrueConf ID

## 6.1.1.10 OnCmdBlock()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdBlock (
    [in] BSTR peerId )
```

**Deprecated** Notification about outgoing ban command

## Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

#### 6.1.1.11 OnCmdChatClear()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdChatClear (
    [in] BSTR chatId )
```

**Deprecated** Chat history clear notification

##### Parameters

<i>chat</i> ↔ <i>Id</i>	- chat ID
----------------------------	-----------

#### 6.1.1.12 OnCmdCreateGroup()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdCreateGroup (
    [in] BSTR groupName )
```

**Deprecated** Create address book group outgoing notification

##### Parameters

<i>groupName</i>	- group name
------------------	--------------

#### 6.1.1.13 OnCmdRemoveFromAbook()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdRemoveFromAbook (
    [in] BSTR peerId )
```

**Deprecated** Notification about outgoing remove from address book command

##### Parameters

<i>peer</i> ↔ <i>Id</i>	- TrueConf ID
----------------------------	---------------

#### 6.1.1.14 OnCmdRemoveFromGroup()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdRemoveFromGroup (
```

```
[in] LONG groupId,
[in] BSTR peerId )
```

**Deprecated** Notification about outgoing remove from address book's group command

#### Parameters

<i>peerId</i>	- TrueConf ID
<i>groupId</i>	- group ID

#### 6.1.1.15 OnCmdRemoveGroup()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdRemoveGroup (
    [in] LONG groupId )
```

**Deprecated** Notification about outgoing remove address book's group command

#### Parameters

<i>groupId</i>	- group ID
----------------	------------

#### 6.1.1.16 OnCmdRenameGroup()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdRenameGroup (
    [in] LONG groupId,
    [in] BSTR groupName )
```

**Deprecated** Notification about outgoing address book's group rename command

#### Parameters

<i>groupId</i>	- group ID
<i>groupName</i>	- new group name

#### 6.1.1.17 OnCmdRenameInAbook()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdRenameInAbook (
```

```
[in] BSTR peerId,
[in] BSTR peerDn )
```

**Deprecated** Notification about outgoing user rename in address book command

#### Parameters

<i>peerId</i>	- TrueConf ID
<i>peerDn</i>	- new user's display name

#### 6.1.1.18 OnCmdUnblock()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCmdUnblock (
[in] BSTR peerId )
```

**Deprecated** Notification about outgoing unban command

#### Parameters

<i>peer↔ Id</i>	- TrueConf ID
---------------------	---------------

#### 6.1.1.19 OnCommandReceived()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnCommandReceived (
[in] BSTR peerId,
[in] BSTR command )
```

On remote command receiving

#### Parameters

<i>peerId</i>	- sender TrueConf ID
<i>command</i>	- commang string

#### 6.1.1.20 OnCommandSent()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnCommandSent (
[in] BSTR peerId,
[in] BSTR command )
```

Outgoing notification about successful command sending

**Parameters**

<i>peerId</i>	- TrueConf ID
<i>command</i>	- command string

**6.1.1.21 OnConferenceCreated()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnConferenceCreated (
    [in] BSTR eventDetails )
```

Call start notification

**6.1.1.22 OnConferenceDeleted()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnConferenceDeleted (
    [in] BSTR eventDetails )
```

Call end notification

**6.1.1.23 OnContactBlocked()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnContactBlocked (
    [in] BSTR eventDetails )
```

Ban notification

**6.1.1.24 OnContactDeleted()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnContactDeleted (
    [in] BSTR eventDetails )
```

Delete from address book notification

**6.1.1.25 OnContactUnblocked()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnContactUnblocked (
    [in] BSTR eventDetails )
```

Unban notification

**6.1.1.26 OnDetailInfo()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnDetailInfo (
    [in] BSTR eventDetails )
```

User detailed info arrival notification

**6.1.1.27 OnDeviceModesDone()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnDeviceModesDone (
    [in] BSTR eventDetails )
```

Video pin and mode information arrival notification

**6.1.1.28 OnFileAccepted()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnFileAccepted (
    [in] LONG fileId )
```

File accepted notification

**Parameters**

<i>fileId</i>	- file ID
---------------	-----------

**6.1.1.29 OnFileConferenceSent()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnFileConferenceSent (
    [in] LONG fileId,
    [in] BSTR fileName )
```

File sent to current conference notification

**Parameters**

<i>fileId</i>	- file ID
<i>fileName</i>	- file Name

**6.1.1.30 OnFileRejected()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnFileRejected (
    [in] LONG fileId )
```

File rejected notification



## Parameters

<i>fileId</i>	- file ID
---------------	-----------

## 6.1.1.31 OnFileSent()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnFileSent (
    [in] LONG fileId,
    [in] BSTR fileName,
    [in] BSTR peerId )
```

File sent to user notification

## Parameters

<i>fileId</i>	- file ID
<i>fileName</i>	- file Name
<i>peerId</i>	- TrueConf ID

## 6.1.1.32 OnFileTransferAvailable()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnFileTransferAvailable (
    [in] VARIANT_BOOL available )
```

File transfer availability notification

## Parameters

<i>available</i>	- true if file transfer is available, false otherwise
------------------	---

## 6.1.1.33 OnGroupChatMessageSent()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnGroupChatMessageSent (
    [in] BSTR message )
```

Message sent to group chat notification

## Parameters

<i>message</i>	- message text
----------------	----------------

**6.1.1.34 OnGroupsUpdate()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnGroupsUpdate (
    [in] BSTR jsonGroupList )
```

User group change notification

**Parameters**

<i>jsonGroupList</i>	- string with detailed group information, including members, in JSON format
----------------------	---

**6.1.1.35 OnHangUpPressed()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnHangUpPressed (
    [in] BSTR eventDetails )
```

Hang up button pressed notification

**6.1.1.36 OnHardwareChanged()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnHardwareChanged (
    [in] BSTR eventDetails )
```

Hardware change notification

**6.1.1.37 OnIncomingChatMessage()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnIncomingChatMessage (
    [in] BSTR peerId,
    [in] BSTR peerDn,
    [in] BSTR message,
    [in] ULONGLONG time )
```

Incoming chat notificaton

**Parameters**

<i>peerId</i>	- sender's TrueConf ID
<i>peerDn</i>	- sender's display name
<i>message</i>	- message
<i>time</i>	- message time in unix-time format

**6.1.1.38 OnIncomingGroupChatMessage()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnIncomingGroupChatMessage (
    [in] BSTR peerId,
    [in] BSTR peerDn,
    [in] BSTR message,
    [in] ULONGLONG time )
```

Incoming group chat notificaton

**Parameters**

<i>peerId</i>	- sender's TrueConf ID
<i>peerDn</i>	- sender's display name
<i>message</i>	- message text
<i>time</i>	- message time in unix-time format

**6.1.1.39 OnIncomingRequestToPodiumAnswered()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnIncomingRequestToPodiumAnswered (
    [in] BSTR eventDetails )
```

Podium request reply notification

**6.1.1.40 OnInviteReceived()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnInviteReceived (
    [in] BSTR eventDetails )
```

Invite notification

**6.1.1.41 OnInviteRequestSent()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnInviteRequestSent (
    [in] BSTR eventDetails )
```

Invite request sent notification

**6.1.1.42 OnInviteSent()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnInviteSent (
    [in] BSTR eventDetails )
```

Invite sent notificaton

#### 6.1.1.43 OnJabraHangUpPressed()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnJabraHangUpPressed ( )
```

Hang up on external hardware device

#### 6.1.1.44 OnJabraHookOffPressed()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnJabraHookOffPressed ( )
```

Off hook on external hardware device

#### 6.1.1.45 OnLogin()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnLogin (
    [in] BSTR eventDetails )
```

Login notification

#### 6.1.1.46 OnLogout()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnLogout (
    [in] BSTR eventDetails )
```

Logout notification

#### 6.1.1.47 OnOffHookPressed()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnOffHookPressed (
    [in] BSTR eventDetails )
```

Call button pressed notification

#### 6.1.1.48 OnReceiversInfoUpdated()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnReceiversInfoUpdated (
    [in] BSTR eventDetails )
```

Receiver information update notification

#### 6.1.1.49 OnRecordRequest()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRecordRequest (
    [in] BSTR eventDetails )
```

Record request notification

**6.1.1.50 OnRecordRequestReply()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRecordRequestReply (
    [in] BSTR eventDetails )
```

Record request reply notificaton

**6.1.1.51 OnRejectReceived()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRejectReceived (
    [in] BSTR eventDetails )
```

Reject notification

**6.1.1.52 OnRejectSent()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRejectSent (
    [in] BSTR eventDetails )
```

Reject sent notificaton

**6.1.1.53 OnRemarkCountDown()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRemarkCountDown (
    [in] BSTR eventDetails )
```

Remark countdown notification

**6.1.1.54 OnRequestInviteReceived()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRequestInviteReceived (
    [in] BSTR eventDetails )
```

Request to join to your group conference notification

**6.1.1.55 OnRestrictionsChanged()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRestrictionsChanged (
    [in] BSTR eventDetails )
```

Restrictions update notification

**6.1.1.56 OnRoleChanged()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnRoleChanged (
    [in] BSTR eventDetails )
```

Role change in conference user list notification

#### 6.1.1.57 OnSelfSSInfoUpdate()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnSelfSSInfoUpdate (
    [in] BSTR eventDetails )
```

Slideshow information update notificaton

#### 6.1.1.58 OnServerConnected()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnServerConnected (
    [in] BSTR eventDetails )
```

Connect to server notification

#### 6.1.1.59 OnServerDisconnected()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnServerDisconnected (
    [in] BSTR eventDetails )
```

Disconnect from server notification

#### 6.1.1.60 OnSettingsChanged()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnSettingsChanged (
    [in] BSTR eventDetails )
```

Settings change notificaton

#### 6.1.1.61 OnSlideShowInfoUpdate()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnSlideShowInfoUpdate (
    [in] BSTR eventDetails )
```

Slideshowo parameter change

##### Parameters

<i>eventDetails</i>	- full notificatoin test in JSON.
---------------------	-----------------------------------

#### 6.1.1.62 OnSlideShowStart()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnSlideShowStart (
    [in] BSTR eventDetails )
```

Slideshow start notification

#### 6.1.1.63 OnSlideShowStop()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnSlideShowStop (
    [in] BSTR eventDetails )
```

Slideshow end notificaton

#### 6.1.1.64 OnStart()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnStart ( )
```

Initialization start notification

#### 6.1.1.65 OnStopCalling()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnStopCalling (
    [in] BSTR eventDetails )
```

End of P2P call

#### 6.1.1.66 OnTestAudioCapturerStateUpdated()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnTestAudioCapturerStateUpdated (
    [in] VARIANT_BOOL started )
```

Microphone test progress notification

##### Parameters

<i>started</i>	- true when test started, false when ended
----------------	--

#### 6.1.1.67 OnToneDial()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnToneDial (
    [in] BSTR symbol )
```

In-call DTMF tone sent notificaion

##### Parameters

<i>symbol</i>	- sent DTMF tone
---------------	------------------

**6.1.1.68 OnUpdateAvatar()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnUpdateAvatar (
    [in] BSTR eventDetails )
```

Avatar update notification

**6.1.1.69 OnUpdateCameraInfo()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnUpdateCameraInfo (
    [in] BSTR eventDetails )
```

Camera format change notification

**6.1.1.70 OnUpdateParticipantList()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnUpdateParticipantList (
    [in] BSTR eventDetails )
```

Conference participant list change notification

**6.1.1.71 OnVideoMatrixChanged()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnVideoMatrixChanged (
    [in] BSTR eventDetails )
```

Video matrix change notificaton

**6.1.1.72 OnXAfterStart()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXAfterStart ( )
```

Nonitifacton about full CallX initialization

**6.1.1.73 OnXChangeState()**

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXChangeState (
    [in] LONG prevState,
    [in] LONG newState )
```

State change notification

**Parameters**

<i>prevState</i>	- previous state
<i>newState</i>	- new state



prevState and newState could have the following values

- 0 none, no server connection and application does not try to connect to server
- 1 connect, application tries to connect to server
- 2 login, login required
- 3 normal, application connected and logged in
- 4 wait, incoming or outgoing call establishment in progress
- 5 conference, application is in call
- 6 close, call is terminating

#### 6.1.1.74 OnXCommandExecution()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXCommandExecution (
    [in] BSTR cmdName,
    [in] BSTR allData )
```

Command execution reply

##### Parameters

<i>cmdName</i>	- command name
<i>allData</i>	- command reply in JSON format

#### 6.1.1.75 OnXError()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXError (
    [in] LONG errorCode,
    [in] BSTR errorMsg )
```

Error notification

##### Parameters

<i>errorCode</i>	- integer error code
<i>errorMsg</i>	- error text

#### 6.1.1.76 OnXFileReceive()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileReceive (
    [in] BSTR peerId,
```

```
[in] LONG fileId,
[in] BSTR fileName,
BSTR fileCaption )
```

File receiving complete notification

#### 6.1.1.77 OnXFileReceiveProgress()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileReceiveProgress (
    [in] LONG percent,
    [in] LONG fileId,
    [in] BSTR fileName,
    [in] BSTR fileCaption )
```

File receiving progress notification

#### 6.1.1.78 OnXFileRequestReceived()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileRequestReceived (
    [in] LONG fileId,
    [in] BSTR fileName,
    [in] BSTR fileCaption,
    [in] BSTR peerId )
```

On incoming file transfer request received

#### 6.1.1.79 OnXFileSend()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileSend (
    [in] LONG fileId,
    [in] BSTR filePath,
    [in] BSTR fileCaption )
```

File sending complete notification

#### 6.1.1.80 OnXFileSendError()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileSendError (
    [in] LONG error_code,
    [in] LONG fileId,
    [in] BSTR filePath,
    [in] BSTR fileCaption )
```

On file send error

#### 6.1.1.81 OnXFileStatusChange()

```
HRESULT TrueConf_CallXLib::_ITrueConfCallXEvents::OnXFileStatusChange (
    [in] LONG fileId,
    [in] LONG fileStatus,
    [in] LONG directionType )
```

On file status change while file transfer

## Parameters

<i>fileId</i>	- unique file id
<i>directionType</i>	<ul style="list-style-type: none"> <li>• 1 - Download</li> <li>• 2 - Upload</li> </ul>
<i>fileStatus</i>	<ul style="list-style-type: none"> <li>• 1 - StartError</li> <li>• 2 - Started</li> <li>• 3 - Complete</li> <li>• 4 - UndefinedError</li> <li>• 5 - Pending</li> </ul>

## 6.1.1.82 OnXLogin()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXLogin ( )
```

Login confirmation

## 6.1.1.83 OnXLoginError()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXLoginError (
    [in] LONG errorCode )
```

Logon error notificaton

## Parameters

<i>errorCode</i>	<p>- error code</p> <ul style="list-style-type: none"> <li>• 1 - answer on CheckUserLoginStatus_Method, if current CID is already authorized at TransportRouter</li> <li>• 2 - answer on CheckUserLoginStatus_Method, if current CID is not authorized at TransportRouter - can try to login</li> <li>• 3 - incorrect password or other problems with DB</li> <li>• 4 - client shouldn't show error to user (example: incorrect AutoLoginKey)</li> <li>• 5 - license restriction of online users reached, server cannot login you</li> <li>• 6 - user exist, but he is disabled to use this server</li> <li>• 7 - client should retry login after timeout (value in container or default), due to server busy or other server problems</li> <li>• 8 - user cannot login using this client app (should use other type of client app)</li> </ul>
------------------	--

#### 6.1.1.84 OnXNotify()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXNotify (
    [in] BSTR data )
```

Component event handler

##### Parameters

<i>data</i>	- event data in JSON
-------------	----------------------

#### 6.1.1.85 OnXStartFail()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXStartFail ( )
```

Failed start notification

Possible reasons

- wrong CallXPath;
- application stopped functioning before full start (OnXAfterStart).

#### 6.1.1.86 OnXTerminate()

```
void TrueConf_CallXLib::_ITrueConfCallXEvents::OnXTerminate ( )
```

Termination notification in this state application could be restarted with XRestart or XRestartForce commands

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

- [TrueConf\\_CallX.IDL](#)

## 6.2 ITrueConfCallX Interface Reference

```
import "TrueConf_CallX.IDL";
```

Inherits IDispatch.

## Public Member Functions

- HRESULT [XExecuteCommand](#) ([in] BSTR cmd)
- HRESULT [XRestart](#) ()
- HRESULT [XSelectCamera](#) ([in] BSTR cameraName)
- HRESULT [XGetCameraList](#) ([out, retval] BSTR \*cameras)
- HRESULT [XDeselectCamera](#) ()
- HRESULT [XGetMicList](#) ([out, retval] BSTR \*microphones)
- HRESULT [XGetSpeakerList](#) ([out, retval] BSTR \*speakers)
- HRESULT [XDeselectMic](#) ()
- HRESULT [XDeselectSpeaker](#) ()
- HRESULT [XHangUpForAll](#) ()
- HRESULT [XSetBackgroundImage](#) ([in] BSTR imgPath)
- HRESULT [XCheckForEvent](#) ([in] BSTR notificationData, [in] BSTR eventName, [out, retval] VARIANT\_BOOL \*eventDetected)
- HRESULT [XStartPictureBroadcast](#) ([in] BSTR imgPath)
- HRESULT [XStopPictureBroadcast](#) ()
- HRESULT [accept](#) ()
- HRESULT [acceptPeer](#) ([in] BSTR peerId)
- HRESULT [acceptPodiumInvite](#) ()
- HRESULT [acceptPodiumRequest](#) ()
- HRESULT [activateLicense](#) ([in] BSTR licenseKey)
- HRESULT [addContactToAbook](#) ([in] BSTR callId, [in] BSTR displayName)
- HRESULT [allowRecord](#) ()
- HRESULT [appUpdate](#) ()
- HRESULT [call](#) ([in] BSTR peerId)
- HRESULT [changeVideoMatrix](#) ([in] BSTR jsonMatrix)
- HRESULT [connectToServer](#) ([in] BSTR serverStr)
- HRESULT [createConference](#) ([in] BSTR confName, [in] BSTR type, [in] VARIANT\_BOOL autoAccept, [in] BSTR users)
- HRESULT [forbidRecord](#) ()
- HRESULT [extendUidTtl](#) ()
- HRESULT [getAbook](#) ()
- HRESULT [getAppState](#) ()
- HRESULT [getAudioMute](#) ()
- HRESULT [getSelfieBroadcast](#) ()
- HRESULT [getDisplayNameById](#) ([in] BSTR peerId)
- HRESULT [getParticipantsList](#) ([out, retval] BSTR \*returnJson)
- HRESULT [getContactDetails](#) ([in] BSTR peerId)
- HRESULT [getHardware](#) ()
- HRESULT [getHardwareKey](#) ()
- HRESULT [getMaxConfTitleLength](#) ()
- HRESULT [getMicMute](#) ()
- HRESULT [getMonitorsInfo](#) ()
- HRESULT [getIdListRequestInvite](#) ()
- HRESULT [getModes](#) ()
- HRESULT [getSettings](#) ()
- HRESULT [getSystemInfo](#) ()
- HRESULT [getVideoMatrix](#) ()
- HRESULT [getVideoMute](#) ()
- HRESULT [getLicenseType](#) ()
- HRESULT [gotoPodium](#) ()
- HRESULT [hangUp](#) ()
- HRESULT [hangUp2](#) ([in] VARIANT\_BOOL forAll)
- HRESULT [inviteToConference](#) ([in] BSTR peerId)

- HRESULT [inviteToPodium](#) ([in] BSTR peerId)
- HRESULT [removeFromPodium](#) ([in] BSTR peerId)
- HRESULT [removePeer](#) ([in] BSTR peerId)
- HRESULT [leavePodium](#) ()
- HRESULT [login](#) ([in] BSTR login, [in] BSTR password)
- HRESULT [logout](#) ()
- HRESULT [ptzRight](#) ()
- HRESULT [ptzLeft](#) ()
- HRESULT [ptzUp](#) ()
- HRESULT [ptzDown](#) ()
- HRESULT [ptzZoomIn](#) ()
- HRESULT [ptzZoomOut](#) ()
- HRESULT [reject](#) ()
- HRESULT [rejectPeer](#) ([in] BSTR peerId)
- HRESULT [rejectInvitationToPodium](#) ()
- HRESULT [rejectRequestToPodium](#) ()
- HRESULT [removeContactFromAbook](#) ([in] BSTR callId)
- HRESULT [setAudioMute](#) ([in] VARIANT\_BOOL mute)
- HRESULT [setSelfieBroadcast](#) ([in] BSTR paramsInJSON)
- HRESULT [setHardware](#) ([in] BSTR hardwareInJson)
- HRESULT [setMicMute](#) ([in] VARIANT\_BOOL mute)
- HRESULT [setModes](#) ([in] BSTR pin, [in] BSTR mode)
- HRESULT [setSettings](#) ([in] BSTR settingsInJson)
- HRESULT [setVideoMute](#) ([in] VARIANT\_BOOL mute)
- HRESULT [shutdown](#) ()
- HRESULT [shutdown2](#) ([in] VARIANT\_BOOL forAll)
- HRESULT [startScreenCapture](#) ([in] BSTR id)
- HRESULT [startAudioRemark](#) ()
- HRESULT [stopScreenCapture](#) ()
- HRESULT [XSetCameraByIndex](#) ([in] LONG cameraIdx)
- HRESULT [XGetCameraByIndex](#) ([in] LONG cameraIdx, [out, retval] BSTR \*cameraName)
- HRESULT [XGetCameraCount](#) ([out, retval] LONG \*camerasCount)
- HRESULT [XSelectMicByIndex](#) ([in] LONG microphoneIdx)
- HRESULT [XSelectSpeakerByIndex](#) ([in] LONG speakerIdx)
- HRESULT [XGetMicByIndex](#) ([in] LONG microphoneIdx, [out, retval] BSTR \*microphoneName)
- HRESULT [XGetSpeakerByIndex](#) ([in] LONG speakerIdx, [out, retval] BSTR \*speakerName)
- HRESULT [XGetMicCount](#) ([out, retval] LONG \*microphonesCount)
- HRESULT [XGetSpeakerCount](#) ([out, retval] LONG \*speakersCount)
- HRESULT [XAddSlide](#) ([in] BSTR imgPath)
- HRESULT [XRemoveSlide](#) ([in] LONG imgIdx)
- HRESULT [startSlideShow](#) ([in] BSTR slideShowName)
- HRESULT [showSlide](#) ([in] LONG slideIdx)
- HRESULT [nextSlide](#) ()
- HRESULT [prevSlide](#) ()
- HRESULT [firstSlide](#) ()
- HRESULT [lastSlide](#) ()
- HRESULT [moveSlideUp](#) ([in] LONG slideIdx)
- HRESULT [moveSlideDown](#) ([in] LONG slideIdx)
- HRESULT [stopSlideShow](#) ()
- HRESULT [getSlideShowInfo](#) ()
- HRESULT [switchVideoStream](#) ([in] BSTR peerId, [in] VARIANT\_BOOL mainCamera)
- HRESULT [XAddSlidesDialog](#) ([out, retval] ULONG \*errorCode)
- HRESULT [XForceRestart](#) ()
- HRESULT [XGetHandle](#) ([out, retval] ULONG \*handle)
- HRESULT [takePicture](#) ([in] BSTR outputFileName)

- HRESULT [ptzStop](#) ()
- HRESULT [XFileSend](#) ([in] BSTR peerId, [in] BSTR filePath, [in] BSTR fileCaption, [out, retval] LONG \*resId)
- HRESULT [XFileAccept](#) ([in] LONG fileId, [out, retval] VARIANT\_BOOL \*success)
- HRESULT [XFileTransferAvailable](#) ([out, retval] VARIANT\_BOOL \*available)
- HRESULT [XFileStatsInfo](#) ([in] LONG fileId, [out] LONG \*size, [out] LONG \*downloaded, [out] LONG \*uploaded, [out] LONG \*u\_speed, [out] LONG \*d\_speed, [out] LONG \*peers, [out, retval] VARIANT\_BOOL \*success)
- HRESULT [XFileSendAbort](#) ([in] LONG fileId, [out, retval] VARIANT\_BOOL \*success)
- HRESULT [XFileReject](#) ([in] LONG fileId, [out, retval] VARIANT\_BOOL \*success)
- HRESULT [XGetFileSendStatus](#) ([in] LONG id, [out, retval] LONG \*result)
- HRESULT [XFileDelete](#) ([in] LONG fileId, [out, retval] VARIANT\_BOOL \*success)
- HRESULT [XGetFileHashById](#) ([in] LONG fileId, [out, retval] BSTR \*hash)
- HRESULT [XGetFileIdByHash](#) ([in] BSTR hash, [out, retval] LONG \*fileId)
- HRESULT [XGetFileDownloadPath](#) ([in] LONG fileId, [out, retval] BSTR \*filePath)
- HRESULT [XGetFileTransferList](#) ([out, retval] BSTR \*files)
- HRESULT [sendCommand](#) ([in] BSTR peerId, [in] BSTR cmd)
- HRESULT [XRemotePrintPdf](#) ([in] BSTR peerId, [in] LONG fileId, [out, retval] LONG \*retCode)
- HRESULT [XPrintPdf](#) ([in] BSTR filePath, [out, retval] LONG \*retCode)
- HRESULT [XGetCallXProcessId](#) ([out, retval] LONG \*pid)
- HRESULT [XFileSendToConference](#) ([in] BSTR filePath, [in] BSTR fileCaption, [out, retval] LONG \*resId)
- HRESULT [addToAbook](#) ([in] BSTR peerId, [in] BSTR displayName, [out, retval] BSTR \*result)
- HRESULT [renameInAbook](#) ([in] BSTR peerId, [in] BSTR newDisplayName, [out, retval] BSTR \*result)
- HRESULT [removeFromAbook](#) ([in] BSTR peerId, [out, retval] BSTR \*result)
- HRESULT [block](#) ([in] BSTR peerId, [out, retval] BSTR \*result)
- HRESULT [unblock](#) ([in] BSTR peerId, [out, retval] BSTR \*result)
- HRESULT [getBanList](#) ([out, retval] BSTR \*result)
- HRESULT [createGroup](#) ([in] BSTR name, [out, retval] BSTR \*result)
- HRESULT [removeGroup](#) ([in] LONG groupId, [out, retval] BSTR \*result)
- HRESULT [renameGroup](#) ([in] LONG groupId, [in] BSTR newGroupName, [out, retval] BSTR \*result)
- HRESULT [addToGroup](#) ([in] LONG groupId, [in] BSTR peerId, [out, retval] BSTR \*result)
- HRESULT [removeFromGroup](#) ([in] LONG groupId, [in] BSTR peerId, [out, retval] BSTR \*result)
- HRESULT [getGroups](#) ([out, retval] BSTR \*result)
- HRESULT [testAudioRenderer](#) ([out, retval] BSTR \*result)
- HRESULT [testAudioCatcherStart](#) ([out, retval] BSTR \*result)
- HRESULT [testAudioCatcherStop](#) ([out, retval] BSTR \*result)
- HRESULT [toneDial](#) ([in] BSTR symbol, [out, retval] BSTR \*result)
- HRESULT [sendMessage](#) ([in] BSTR peerId, [in] BSTR message, [out, retval] BSTR \*result)
- HRESULT [sendGroupMessage](#) ([in] BSTR message, [out, retval] BSTR \*result)
- HRESULT [chatClear](#) ([in] BSTR chatId, [out, retval] BSTR \*result)
- HRESULT [getListOfChats](#) ([out, retval] BSTR \*result)
- HRESULT [getChatLastMessages](#) ([in] BSTR chatId, [in] LONG count, [out, retval] BSTR \*result)
- HRESULT [getCallHistory](#) ([in] LONG count, [out, retval] BSTR \*result)
- HRESULT [setLastCallsViewed](#) ([out, retval] BSTR \*result)
- HRESULT [XEncryptedLogin](#) ([in] BSTR login, [in] BSTR password, [in] VARIANT\_BOOL encryptPassword)
- HRESULT [clearSlides](#) ()
- HRESULT [acceptPodiumRequest2](#) ([in] BSTR peerId)
- HRESULT [allowRecord2](#) ([in] BSTR peerId)
- HRESULT [forbidRecord2](#) ([in] BSTR peerId)
- HRESULT [rejectRequestToPodium2](#) ([in] BSTR peerId)
- HRESULT [turnRemoteCamera](#) ([in] BSTR peerId, [in] VARIANT\_BOOL on)
- HRESULT [turnRemoteMic](#) ([in] BSTR peerId, [in] VARIANT\_BOOL on)

## Properties

- BSTR [CallXPath](#) [get, set]
- BSTR [Camera](#) [get, set]
- BSTR [Microphone](#) [get, set]
- BSTR [Speaker](#) [get, set]
- BSTR [XCommandLine](#) [get, set]

## 6.2.1 Member Function Documentation

### 6.2.1.1 `accept()`

```
HRESULT ITrueConfCallX::accept ( )
```

Accept incoming call. Could be executed only when incoming call or group conference invite is in process, ignored otherwise

#### See also

[acceptPeer](#)

### 6.2.1.2 `acceptPeer()`

```
HRESULT ITrueConfCallX::acceptPeer (
    [in] BSTR peerId )
```

Allow user to join your group conference. Could be executed only when incoming request for joining conference is in process, ignored otherwise. Moderator rights for current conference are required.

#### Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

#### See also

[accept](#)

### 6.2.1.3 `acceptPodiumInvite()`

```
HRESULT ITrueConfCallX::acceptPodiumInvite ( )
```

Accept invitation to take podium. Could be executed only when incoming request for taking podium is in process, ignored otherwise.



## 6.2.1.4 acceptPodiumRequest()

```
HRESULT ITrueConfCallX::acceptPodiumRequest ( )
```

**Deprecated** Allow user to take podium. Could be executed only when incoming request from user for taking podium is in process, ignored otherwise. Moderator rights for current conference are required.

## 6.2.1.5 acceptPodiumRequest2()

```
HRESULT ITrueConfCallX::acceptPodiumRequest2 (
    [in] BSTR peerId )
```

Allow user to take podium. Could be executed only when incoming request from user for taking podium is in process, ignored otherwise. Moderator rights for current conference are required.

## Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

## 6.2.1.6 activateLicense()

```
HRESULT ITrueConfCallX::activateLicense (
    [in] BSTR licenseKey )
```

Activate license key

## Parameters

<i>licenseKey</i>	- license key
-------------------	---------------

## 6.2.1.7 addContactToAbook()

```
HRESULT ITrueConfCallX::addContactToAbook (
    [in] BSTR callId,
    [in] BSTR displayName )
```

Add contact to user address book

## Parameters

<i>callId</i>	- TrueConf ID of user to add
<i>displayName</i>	- Display Name of added user

### 6.2.1.8 addToAbook()

```
HRESULT ITrueConfCallX::addToAbook (
    [in] BSTR peerId,
    [in] BSTR displayName,
    [out, retval] BSTR * result )
```

Add user to address book

#### Parameters

<i>peerId</i>	- TrueConf ID
<i>displayName</i>	- user display name

#### Returns

json string with return result

### 6.2.1.9 addToGroup()

```
HRESULT ITrueConfCallX::addToGroup (
    [in] LONG groupId,
    [in] BSTR peerId,
    [out, retval] BSTR * result )
```

Add user to group

#### Parameters

<i>groupId</i>	- group ID
<i>peerId</i>	- TrueConf ID

#### Returns

json string with return result

### 6.2.1.10 allowRecord()

```
HRESULT ITrueConfCallX::allowRecord ( )
```

**Deprecated** Allow your video/audio stream recording Could be executed only when incoming request on stream recording is in process, ignored otherwise.

**See also**

denyRecord

**6.2.1.11 allowRecord2()**

```
HRESULT ITrueConfCallX::allowRecord2 (
    [in] BSTR peerId )
```

Allow your video/audio stream recording Could be executed only when incoming request on stream recording is in process, ignored otherwise.

**See also**

denyRecord

**Parameters**

<i>peerId</i>	- TrueConf ID
---------------	---------------

**6.2.1.12 appUpdate()**

```
HRESULT ITrueConfCallX::appUpdate ( )
```

Start application update After successful update conference will be restarted

**6.2.1.13 block()**

```
HRESULT ITrueConfCallX::block (
    [in] BSTR peerId,
    [out, retval] BSTR * result )
```

Ban user

**Parameters**

<i>peerId</i>	- TrueConf ID
---------------	---------------

**Returns**

json string with return result

### 6.2.1.14 call()

```
HRESULT ITrueConfCallX::call (
    [in] BSTR peerId )
```

Call user or join the conference by CID. This command starts p2p call or joins the conference. This command starts p2p call.

#### Parameters

<i>peerId</i>	- user TrueConf ID or alias, or other supported ID types (H323, SIP, RTSP) in correct format, or conference CID in correct format ([conference name]@[server name]#vcs).
---------------	--

#### See also

[reject](#)

### 6.2.1.15 changeVideoMatrix()

```
HRESULT ITrueConfCallX::changeVideoMatrix (
    [in] BSTR jsonMatrix )
```

Set videomatrix type and peer plasec Command is available in group conference, ignored otherwise

#### Parameters

<i>jsonMatrix</i>	- videomatrix data in JSON format , sample: {"matrixType":1,"participants":["VideoCaptureSlot","user@trueconf.com"]}
-------------------	---

### 6.2.1.16 chatClear()

```
HRESULT ITrueConfCallX::chatClear (
    [in] BSTR chatId,
    [out, retval] BSTR * result )
```

Clear chat history

#### Parameters

<i>chatId</i>	- chat ID, peerId of party for pesonal chats or group conference ID
---------------	---

#### Returns

json string with return result

## 6.2.1.17 clearSlides()

```
HRESULT ITrueConfCallX::clearSlides ( )
```

Clear slides

## 6.2.1.18 connectToServer()

```
HRESULT ITrueConfCallX::connectToServer (
    [in] BSTR serverStr )
```

Connect to server

## Parameters

<i>serverStr</i>	<p>- specify server IP address or DNS name in format server.name[:port], for example "video.company.com:4307"</p> <ul style="list-style-type: none"> <li>• port number is optional, default port number 4307 is used if omitted</li> <li>• use empty string to connect to TrueConf Online cloud             <ul style="list-style-type: none"> <li>– use .name to connect to other cloud</li> </ul> </li> </ul>
------------------	---

## See also

OnServerConnected  
OnServerDisconnected

## 6.2.1.19 createConference()

```
HRESULT ITrueConfCallX::createConference (
    [in] BSTR confName,
    [in] BSTR type,
    [in] VARIANT_BOOL autoAccept,
    [in] BSTR users )
```

Create conference with specific parametes

## Parameters

<i>confName</i>	- conference name
<i>type</i>	<p>- Conference type. One of</p> <ul style="list-style-type: none"> <li>• "symmetric"</li> <li>• "asymmetric"</li> <li>• "role"</li> </ul>

## Parameters

<i>autoAccept</i>	- allow users to join automatically
<i>users</i>	- user ID list to invite. TrueConf ID and H323,SIP, RTSP ID's could be used in special format this parameter can be empty string , or list, sepatated by ',' ("user1@my_server,user2@my_server")

## 6.2.1.20 createGroup()

```
HRESULT ITrueConfCallX::createGroup (
    [in] BSTR name,
    [out, retval] BSTR * result )
```

Create group

## Parameters

<i>name</i>	- group name
-------------	--------------

## Returns

json string with return result

## 6.2.1.21 extendUidTtl()

```
HRESULT ITrueConfCallX::extendUidTtl ( )
```

## 6.2.1.22 firstSlide()

```
HRESULT ITrueConfCallX::firstSlide ( )
```

Show first slide Command could be executed only in process of slideshow

## 6.2.1.23 forbidRecord()

```
HRESULT ITrueConfCallX::forbidRecord ( )
```

**Deprecated** Deny recording request of your stream Could be executed only when incoming request on stream recording is in process, ignored otherwise.

## 6.2.1.24 forbidRecord2()

```
HRESULT ITrueConfCallX::forbidRecord2 (
    [in] BSTR peerId )
```

Deny recording request of your stream Could be executed only when incoming request on stream recording is in process, ignored otherwise.

## Parameters

<i>peer</i> ↔ <i>Id</i>	- TrueConf ID
----------------------------	---------------

**6.2.1.25 getAbook()**

```
HRESULT ITrueConfCallX::getAbook ( )
```

Send address book request to server

**6.2.1.26 getAppState()**

```
HRESULT ITrueConfCallX::getAppState ( )
```

Get application state

**6.2.1.27 getAudioMute()**

```
HRESULT ITrueConfCallX::getAudioMute ( )
```

Get audio mute state

**6.2.1.28 getBanList()**

```
HRESULT ITrueConfCallX::getBanList (
    [out, retval] BSTR * result )
```

Get ban list

**Returns**

json string with return result

**6.2.1.29 getCallHistory()**

```
HRESULT ITrueConfCallX::getCallHistory (
    [in] LONG count,
    [out, retval] BSTR * result )
```

(TODO translate) get call history list

**Parameters**

<i>count</i>	- requested calls count, it should be more than one return json string that contains call history list
--------------	--

**See also**

OnCallHistoryUpdated

**6.2.1.30 getChatLastMessages()**

```
HRESULT ITrueConfCallX::getChatLastMessages (
    [in] BSTR chatId,
    [in] LONG count,
    [out, retval] BSTR * result )
```

Get latest chat messages

**Parameters**

<i>chat↔ Id</i>	- chat ID, peerId of party for pesonal chats or group conference ID
<i>count</i>	- message count

**Returns**

json string with return result and message list

**6.2.1.31 getContactDetails()**

```
HRESULT ITrueConfCallX::getContactDetails (
    [in] BSTR peerId )
```

Get detailed user information

**Parameters**

<i>peer↔ Id</i>	- TrueConf ID
---------------------	---------------

**6.2.1.32 getDisplayNameById()**

```
HRESULT ITrueConfCallX::getDisplayNameById (
    [in] BSTR peerId )
```



Get user display name from TrueConf ID.

#### Parameters

<i>peer</i> ↔ <i>Id</i>	- TrueConf ID
----------------------------	---------------

#### 6.2.1.33 getGroups()

```
HRESULT ITrueConfCallX::getGroups (
    [out, retval] BSTR * result )
```

Get group information

#### Returns

json string with return result

#### 6.2.1.34 getHardware()

```
HRESULT ITrueConfCallX::getHardware ( )
```

Get hardware list

#### 6.2.1.35 getHardwareKey()

```
HRESULT ITrueConfCallX::getHardwareKey ( )
```

Get unique hardware key for license generation

#### 6.2.1.36 getIdListRequestInvite()

```
HRESULT ITrueConfCallX::getIdListRequestInvite ( )
```

Get list of all users requesting to enter conference

#### 6.2.1.37 getLicenseType()

```
HRESULT ITrueConfCallX::getLicenseType ( )
```

Get current license information

**6.2.1.38 getListOfChats()**

```
HRESULT ITrueConfCallX::getListOfChats (
    [out, retval] BSTR * result )
```

Get chat list

**Returns**

json string with chat list

**6.2.1.39 getMaxConfTitleLength()**

```
HRESULT ITrueConfCallX::getMaxConfTitleLength ( )
```

Get maximum conference title length

**6.2.1.40 getMicMute()**

```
HRESULT ITrueConfCallX::getMicMute ( )
```

Get audio input mute information

**6.2.1.41 getModes()**

```
HRESULT ITrueConfCallX::getModes ( )
```

Get modes and pins for current camera device

**6.2.1.42 getMonitorsInfo()**

```
HRESULT ITrueConfCallX::getMonitorsInfo ( )
```

Get system screen information

**6.2.1.43 getParticipantsList()**

```
HRESULT ITrueConfCallX::getParticipantsList (
    [out, retval] BSTR * returnJson )
```

Get conference participants list

#### 6.2.1.44 getSelfieBroadcast()

HRESULT ITrueConfCallX::getSelfieBroadcast ( )

**Deprecated**

#### 6.2.1.45 getSettings()

HRESULT ITrueConfCallX::getSettings ( )

Get settings

#### 6.2.1.46 getSlideShowInfo()

HRESULT ITrueConfCallX::getSlideShowInfo ( )

Get slideshow information

#### 6.2.1.47 getSystemInfo()

HRESULT ITrueConfCallX::getSystemInfo ( )

Get system information

#### 6.2.1.48 getVideoMatrix()

HRESULT ITrueConfCallX::getVideoMatrix ( )

Get video matrix information

#### 6.2.1.49 getVideoMute()

HRESULT ITrueConfCallX::getVideoMute ( )

Get video mute state

#### 6.2.1.50 gotoPodium()

HRESULT ITrueConfCallX::gotoPodium ( )

Send request to take podium. Moderators take podium automatically

See also

onRoleEventOccurred

### 6.2.1.51 hangUp()

```
HRESULT ITrueConfCallX::hangUp ( )
```

Leave conference Could be executed only when incoming request on stream recording is in process, ignored otherwise.

See also

[hangUp2](#)  
[XHangUpForAll](#)

### 6.2.1.52 hangUp2()

```
HRESULT ITrueConfCallX::hangUp2 (
    [in] VARIANT_BOOL forAll )
```

Leave or terminate conference Could be executed only when incoming request on stream recording is in process, ignored otherwise.

Parameters

<i>forAll</i>	<p>- leave or terminate</p> <ul style="list-style-type: none"> <li>• true - terminate conference for all. Requires moderator rights.</li> <li>• false - leave conference</li> </ul>
---------------	---

See also

[hangUp](#)  
[XHangUpForAll](#)

### 6.2.1.53 inviteToConference()

```
HRESULT ITrueConfCallX::inviteToConference (
    [in] BSTR peerId )
```

Invite user to conference. Requires moderator rights. Could be executed only when incoming request on stream recording is in process, ignored otherwise.

Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

#### 6.2.1.54 inviteToPodium()

```
HRESULT ITrueConfCallX::inviteToPodium (
    [in] BSTR peerId )
```

Invite user to podium. Command is valid for role-based conference only. Requires moderator rights. Could be executed only when incoming request on stream recording is in process, ignored otherwise.

##### Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

#### 6.2.1.55 lastSlide()

```
HRESULT ITrueConfCallX::lastSlide ( )
```

Show last slide Command could be executed only in process of slideshow

#### 6.2.1.56 leavePodium()

```
HRESULT ITrueConfCallX::leavePodium ( )
```

Leave podium. Command is valid for role-based conference only. Could be executed only when incoming request on stream recording is in process, ignored otherwise.

#### 6.2.1.57 login()

```
HRESULT ITrueConfCallX::login (
    [in] BSTR login,
    [in] BSTR password )
```

Authorize on server

##### Parameters

<i>login</i>	- User ID
<i>password</i>	- Use password

##### See also

[logout](#)

### 6.2.1.58 logout()

```
HRESULT ITrueConfCallX::logout ( )
```

Deauthorize from server

See also

[login](#)

### 6.2.1.59 moveSlideDown()

```
HRESULT ITrueConfCallX::moveSlideDown (
    [in] LONG slideIdx )
```

Move slide down in list Command could be executed only in process of slideshow

Parameters

<i>slideIdx</i>	- slide index to move
-----------------	-----------------------

### 6.2.1.60 moveSlideUp()

```
HRESULT ITrueConfCallX::moveSlideUp (
    [in] LONG slideIdx )
```

Move slide up in list Command could be executed only in process of slideshow

Parameters

<i>slideIdx</i>	- slide index to move
-----------------	-----------------------

### 6.2.1.61 nextSlide()

```
HRESULT ITrueConfCallX::nextSlide ( )
```

Show next slide Command could be executed only in process of slideshow

### 6.2.1.62 prevSlide()

```
HRESULT ITrueConfCallX::prevSlide ( )
```

Show previous slide Command could be executed only in process of slideshow

**6.2.1.63 ptzDown()**

```
HRESULT ITrueConfCallX::ptzDown ( )
```

Turn camera down. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.64 ptzLeft()**

```
HRESULT ITrueConfCallX::ptzLeft ( )
```

Turn camera left. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.65 ptzRight()**

```
HRESULT ITrueConfCallX::ptzRight ( )
```

Turn camera right. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.66 ptzStop()**

```
HRESULT ITrueConfCallX::ptzStop ( )
```

Stop camera rotation and zoom.

**6.2.1.67 ptzUp()**

```
HRESULT ITrueConfCallX::ptzUp ( )
```

Turn camera up. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.68 ptzZoomIn()**

```
HRESULT ITrueConfCallX::ptzZoomIn ( )
```

Increase camera zoom. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.69 ptzZoomOut()**

```
HRESULT ITrueConfCallX::ptzZoomOut ( )
```

Decrease camera zoom. Use ptzStop command to stop the operation. PTZ camera control

**6.2.1.70 reject()**

```
HRESULT ITrueConfCallX::reject ( )
```

reject incoming call or conference invite Command could be executed only when request is in process/

See also

[accept](#)

**6.2.1.71 rejectInvitationToPodium()**

```
HRESULT ITrueConfCallX::rejectInvitationToPodium ( )
```

reject invitation to take podium

**6.2.1.72 rejectPeer()**

```
HRESULT ITrueConfCallX::rejectPeer (
    [in] BSTR peerId )
```

reject request to join conference

Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

**6.2.1.73 rejectRequestToPodium()**

```
HRESULT ITrueConfCallX::rejectRequestToPodium ( )
```

**Deprecated** reject request to take podium

**6.2.1.74 rejectRequestToPodium2()**

```
HRESULT ITrueConfCallX::rejectRequestToPodium2 (
    [in] BSTR peerId )
```

reject request to take podium



## Parameters

<i>peer↔ Id</i>	- TrueConf ID
---------------------	---------------

## 6.2.1.75 removeContactFromAbook()

```
HRESULT ITrueConfCallX::removeContactFromAbook (
    [in] BSTR callId )
```

Remove user from address book

## Parameters

<i>call↔ Id</i>	- User ID
---------------------	-----------

## 6.2.1.76 removeFromAbook()

```
HRESULT ITrueConfCallX::removeFromAbook (
    [in] BSTR peerId,
    [out, retval] BSTR * result )
```

Delete user from address book

## Parameters

<i>peer↔ Id</i>	- TrueConf ID
---------------------	---------------

## Returns

json string with return result

## 6.2.1.77 removeFromGroup()

```
HRESULT ITrueConfCallX::removeFromGroup (
    [in] LONG groupId,
    [in] BSTR peerId,
    [out, retval] BSTR * result )
```

Remove user from group

**Parameters**

<i>groupId</i>	- group ID
<i>peerId</i>	- TrueConf ID

**Returns**

json string with return result

**6.2.1.78 removeFromPodium()**

```
HRESULT ITrueConfCallX::removeFromPodium (
    [in] BSTR peerId )
```

Remove user from podium. Command is valid for role-based conference only. Requires moderator rights. Could be executed only when incoming request on stream recording is in process, ignored otherwise.

**Parameters**

<i>peerId</i>	- TrueConf ID
---------------	---------------

**6.2.1.79 removeGroup()**

```
HRESULT ITrueConfCallX::removeGroup (
    [in] LONG groupId,
    [out, retval] BSTR * result )
```

Delete group

**Parameters**

<i>groupId</i>	- group ID
----------------	------------

**Returns**

json string with return result

**6.2.1.80 removePeer()**

```
HRESULT ITrueConfCallX::removePeer (
    [in] BSTR peerId )
```

Remove user from conference. Requires moderator rights. Could be executed only when incoming request on stream recording is in process, ignored otherwise.

#### Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

#### 6.2.1.81 renameGroup()

```
HRESULT ITrueConfCallX::renameGroup (
    [in] LONG groupId,
    [in] BSTR newGroupName,
    [out, retval] BSTR * result )
```

Rename group

#### Parameters

<i>groupId</i>	- group ID
<i>newGroupName</i>	- new group name

#### Returns

json string with return result

#### 6.2.1.82 renameInAbook()

```
HRESULT ITrueConfCallX::renameInAbook (
    [in] BSTR peerId,
    [in] BSTR newDisplayName,
    [out, retval] BSTR * result )
```

Rename user in address book

#### Parameters

<i>peerId</i>	- TrueConf ID
<i>newDisplayName</i>	- new user's display name

#### Returns

json

**6.2.1.83 sendCommand()**

```
HRESULT ITrueConfCallX::sendCommand (
    [in] BSTR peerId,
    [in] BSTR cmd )
```

send command to a user, the user will receive onCommandReceived event

**Parameters**

<i>peerId</i>	- TrueConf ID
<i>cmd</i>	- Any any command for user

**6.2.1.84 sendGroupMessage()**

```
HRESULT ITrueConfCallX::sendGroupMessage (
    [in] BSTR message,
    [out, retval] BSTR * result )
```

Send message to current group conference

**Parameters**

<i>message</i>	- message
----------------	-----------

**Returns**

json string with return result

**6.2.1.85 sendMessage()**

```
HRESULT ITrueConfCallX::sendMessage (
    [in] BSTR peerId,
    [in] BSTR message,
    [out, retval] BSTR * result )
```

Send message to user

**Parameters**

<i>peerId</i>	- TrueConf ID
<i>message</i>	- message

**Returns**

json string with return result

**6.2.1.86 setAudioMute()**

```
HRESULT ITrueConfCallX::setAudioMute (
    [in] VARIANT_BOOL mute )
```

Mute/unmute audio output

**6.2.1.87 setHardware()**

```
HRESULT ITrueConfCallX::setHardware (
    [in] BSTR hardwareInJson )
```

Set current hardware

**6.2.1.88 setLastCallsViewed()**

```
HRESULT ITrueConfCallX::setLastCallsViewed (
    [out, retval] BSTR * result )
```

(TODO translate) mark all calls in call history as "viewed", update lastView parameter (see OnCallHistoryUpdated)  
return json string with the result of the command execution

**See also**

OnCallHistoryUpdated

**6.2.1.89 setMicMute()**

```
HRESULT ITrueConfCallX::setMicMute (
    [in] VARIANT_BOOL mute )
```

Select audio input mute state

**6.2.1.90 setModes()**

```
HRESULT ITrueConfCallX::setModes (
    [in] BSTR pin,
    [in] BSTR mode )
```

Set mode and pin of current camera device

### 6.2.1.91 setSelfieBroadcast()

```
HRESULT ITrueConfCallX::setSelfieBroadcast (
    [in] BSTR paramsInJSON )
```

#### Deprecated

### 6.2.1.92 setSettings()

```
HRESULT ITrueConfCallX::setSettings (
    [in] BSTR settingsInJson )
```

Set application settings

### 6.2.1.93 setVideoMute()

```
HRESULT ITrueConfCallX::setVideoMute (
    [in] VARIANT_BOOL mute )
```

Mute/unmute video

### 6.2.1.94 showSlide()

```
HRESULT ITrueConfCallX::showSlide (
    [in] LONG slideIdx )
```

Show slide by index Command could be executed only in process of slideshow

### 6.2.1.95 shutdown()

```
HRESULT ITrueConfCallX::shutdown ( )
```

Shutdown application

### 6.2.1.96 shutdown2()

```
HRESULT ITrueConfCallX::shutdown2 (
    [in] VARIANT_BOOL forAll )
```

Shutdown application with options

#### Parameters

<i>forAll</i>	- if true, terminates conference for all users
---------------	--

### 6.2.1.97 startAudioRemark()

```
HRESULT ITrueConfCallX::startAudioRemark ( )
```

Start remark. Available in role-based conference only. Fails if other user remark is in process. After positive response 5 seconds countdown starts onRemarkCountDown will be called each second

#### See also

onRemarkCountDown

### 6.2.1.98 startScreenCapture()

```
HRESULT ITrueConfCallX::startScreenCapture (
    [in] BSTR id )
```

Start screen or application sharing

#### Parameters

<i>id</i>	- screen or window handle
-----------	---------------------------

### 6.2.1.99 startSlideShow()

```
HRESULT ITrueConfCallX::startSlideShow (
    [in] BSTR slideShowName )
```

Starts slideshow. Slides should be uploaded previously

#### Parameters

<i>slideShowName</i>	- Slideshow name, could be empty
----------------------	----------------------------------

#### See also

[stopSlideShow](#)

### 6.2.1.100 stopScreenCapture()

```
HRESULT ITrueConfCallX::stopScreenCapture ( )
```

Stop screen or application sharing.

**6.2.1.101 stopSlideShow()**

```
HRESULT ITrueConfCallX::stopSlideShow ( )
```

Stop slideshow Command could be executed only in process of slideshow

See also

[startSlideShow](#)

**6.2.1.102 switchVideoStream()**

```
HRESULT ITrueConfCallX::switchVideoStream (
    [in] BSTR peerId,
    [in] VARIANT_BOOL mainCamera )
```

Request video/content stream switch

Parameters

<i>peerId</i>	- TrueConf ID
<i>mainCamera</i>	- content flag true - main stream false - content stream

**6.2.1.103 takePicture()**

```
HRESULT ITrueConfCallX::takePicture (
    [in] BSTR outputFileName )
```

take screen shot of the callx

Parameters

<i>outputFileName</i>	- file name for screen shot output file
-----------------------	---

**6.2.1.104 testAudioCapturerStart()**

```
HRESULT ITrueConfCallX::testAudioCapturerStart (
    [out, retval] BSTR * result )
```

Starts audio capture test

Returns

json string with return result



### 6.2.1.105 testAudioCapturerStop()

```
HRESULT ITrueConfCallX::testAudioCapturerStop (
    [out, retval] BSTR * result )
```

Stops audio capture test

#### Returns

json string with return result

### 6.2.1.106 testAudioRenderer()

```
HRESULT ITrueConfCallX::testAudioRenderer (
    [out, retval] BSTR * result )
```

Starts audio playback test

#### Returns

json string with return result

### 6.2.1.107 toneDial()

```
HRESULT ITrueConfCallX::toneDial (
    [in] BSTR symbol,
    [out, retval] BSTR * result )
```

Send DTMF dial tones in video call

#### Parameters

<i>symbol</i>	- tone to send
---------------	----------------

#### Returns

json string with return result

### 6.2.1.108 turnRemoteCamera()

```
HRESULT ITrueConfCallX::turnRemoteCamera (
    [in] BSTR peerId,
    [in] VARIANT_BOOL on )
```

Turn on/off conference participants camera

## Parameters

<i>peerId</i>	- TrueConf ID of user
<i>on</i>	- turn camera on/off

## 6.2.1.109 turnRemoteMic()

```
HRESULT ITrueConfCallX::turnRemoteMic (
    [in] BSTR peerId,
    [in] VARIANT_BOOL on )
```

Turn on/off conference participants mic

## Parameters

<i>peerId</i>	- TrueConf ID of user
<i>on</i>	- turn mic on/off

## 6.2.1.110 unblock()

```
HRESULT ITrueConfCallX::unblock (
    [in] BSTR peerId,
    [out, retval] BSTR * result )
```

Unban user

## Parameters

<i>peerId</i>	- TrueConf ID
---------------	---------------

## Returns

json string with return result

## 6.2.1.111 XAddSlide()

```
HRESULT ITrueConfCallX::XAddSlide (
    [in] BSTR imgPath )
```

Adds slide to slideshow

**Parameters**

<i>imgPath</i>	- full image path
----------------	-------------------

**6.2.1.112 XAddSlidesDialog()**

```
HRESULT ITrueConfCallX::XAddSlidesDialog (
    [out, retval] ULONG * errorCode )
```

Open slide select dialog for slideshow

**Returns**

returns 0 if ok or error code

**6.2.1.113 XCheckForEvent()**

```
HRESULT ITrueConfCallX::XCheckForEvent (
    [in] BSTR notificationData,
    [in] BSTR eventName,
    [out, retval] VARIANT_BOOL * eventDetected )
```

Check if event exists Helper function that can check which event was triggered in OnXNotify

**Parameters**

<i>notificationData</i>	- data passed in OnXNotify
<i>eventName</i>	- event name

**Returns**

returns true, if eventName exists in notificationData

**6.2.1.114 XDeselectCamera()**

```
HRESULT ITrueConfCallX::XDeselectCamera ( )
```

Do not use camera

**See also**

[XSelectCamera](#)  
[Camera](#)

**6.2.1.115 XDeselectMic()**

```
HRESULT ITrueConfCallX::XDeselectMic ( )
```

Do not use audio input (microphone)

See also

[Microphone](#)

**6.2.1.116 XDeselectSpeaker()**

```
HRESULT ITrueConfCallX::XDeselectSpeaker ( )
```

Do not use audio output (speaker)

See also

[Speaker](#)

**6.2.1.117 XEncryptedLogin()**

```
HRESULT ITrueConfCallX::XEncryptedLogin (
    [in] BSTR login,
    [in] BSTR password,
    [in] VARIANT_BOOL encryptPassword )
```

Authorize on server

Parameters

<i>login</i>	- User ID
<i>password</i>	- Use password
<i>encryptPassword</i>	- not used

See also

[logout](#)

**6.2.1.118 XExecuteCommand()**

```
HRESULT ITrueConfCallX::XExecuteCommand (
    [in] BSTR cmd )
```

Execute command in text format

## Parameters

<i>cmd</i>	- command text
------------	----------------

## 6.2.1.119 XFileAccept()

```
HRESULT ITrueConfCallX::XFileAccept (
    [in] LONG fileId,
    [out, retval] VARIANT_BOOL * success )
```

Start file transfer

## Parameters

<i>file↔ Id</i>	- File ID
---------------------	-----------

## 6.2.1.120 XFileDelete()

```
HRESULT ITrueConfCallX::XFileDelete (
    [in] LONG fileID,
    [out, retval] VARIANT_BOOL * success )
```

Delete file uploaded before

## Parameters

<i>fileID</i>	- file id , use 0 to delete all files
---------------	---------------------------------------

## 6.2.1.121 XFileReject()

```
HRESULT ITrueConfCallX::XFileReject (
    [in] LONG fileId,
    [out, retval] VARIANT_BOOL * success )
```

Reject file transfer request

## Parameters

<i>file↔ Id</i>	- file ID
---------------------	-----------

**6.2.1.122 XFileSend()**

```
HRESULT ITrueConfCallX::XFileSend (
    [in] BSTR peerId,
    [in] BSTR filePath,
    [in] BSTR fileCaption,
    [out, retval] LONG * resId )
```

Send file

**Parameters**

<i>peerId</i>	- TrueConf ID
<i>filePath</i>	- Full path and file name
<i>fileCaption</i>	- File caption

**Returns**

File ID

**6.2.1.123 XFileSendAbort()**

```
HRESULT ITrueConfCallX::XFileSendAbort (
    [in] LONG fileId,
    [out, retval] VARIANT_BOOL * success )
```

Cancel file transfer

**6.2.1.124 XFileSendToConference()**

```
HRESULT ITrueConfCallX::XFileSendToConference (
    [in] BSTR filePath,
    [in] BSTR fileCaption,
    [out, retval] LONG * resId )
```

Send file to all group conference participants

**Parameters**

<i>filePath</i>	- full file path
<i>fileCaption</i>	- file name

**Returns**

file ID

**6.2.1.125 XFileStatsInfo()**

```
HRESULT ITrueConfCallX::XFileStatsInfo (
    [in] LONG fileId,
    [out] LONG * size,
    [out] LONG * downloaded,
    [out] LONG * uploaded,
    [out] LONG * u_speed,
    [out] LONG * d_speed,
    [out] LONG * peers,
    [out, retval] VARIANT_BOOL * success )
```

Get file transfer progress and statistics

**6.2.1.126 XFileTransferAvailable()**

```
HRESULT ITrueConfCallX::XFileTransferAvailable (
    [out, retval] VARIANT_BOOL * available )
```

Check if file transfer is available

**6.2.1.127 XForceRestart()**

```
HRESULT ITrueConfCallX::XForceRestart ( )
```

Restart terminal options

**6.2.1.128 XGetCallXProcessId()**

```
HRESULT ITrueConfCallX::XGetCallXProcessId (
    [out, retval] LONG * pid )
```

Get CallX process ID

**6.2.1.129 XGetCameraByIndex()**

```
HRESULT ITrueConfCallX::XGetCameraByIndex (
    [in] LONG cameraIdx,
    [out, retval] BSTR * cameraName )
```

Returns camera name by index

## Parameters

<i>cameraIdx</i>	- camera index in list camera indexes start from 0.
------------------	---

## 6.2.1.130 XGetCameraCount()

```
HRESULT ITrueConfCallX::XGetCameraCount (
    [out, retval] LONG * camerasCount )
```

Returns number of cameras in system

## 6.2.1.131 XGetCameraList()

```
HRESULT ITrueConfCallX::XGetCameraList (
    [out, retval] BSTR * cameras )
```

Get camera list

## Returns

List of camera names that are present in system. Multilite text separated by "\r\n" newline symbols

## See also

[Camera](#)  
[XGetSpeakerList](#)  
[XGetMicList](#)  
[XSetCameraByIndex](#)  
[XGetCameraCount](#)

## 6.2.1.132 XGetFileDownloadPath()

```
HRESULT ITrueConfCallX::XGetFileDownloadPath (
    [in] LONG fileID,
    [out, retval] BSTR * filePath )
```

(TODO translate) get full path of downloaded file

## Parameters

<i>fileID</i>	- file id
---------------	-----------



**6.2.1.133 XGetFileHashById()**

```
HRESULT ITrueConfCallX::XGetFileHashById (
    [in] LONG fileID,
    [out, retval] BSTR * hash )
```

(TODO translate) get hash by file id

**6.2.1.134 XGetFileIdByHash()**

```
HRESULT ITrueConfCallX::XGetFileIdByHash (
    [in] BSTR hash,
    [out, retval] LONG * fileID )
```

(TODO translate) get file id by hash

**6.2.1.135 XGetFileSendStatus()**

```
HRESULT ITrueConfCallX::XGetFileSendStatus (
    [in] LONG id,
    [out, retval] LONG * result )
```

Get file status (if id != 0) or get common file transfer information (id == 0)

**Parameters**

<i>id</i>	- file id or 0 for common information
-----------	---------------------------------------

**Returns**

- file transfer status
  - 1 - start error
  - 2 - started
  - 3 - complete
  - 4 - undefined error while sending
  - 5 - pending
  - 0 - if id != 0 - id not found
  - 0 - if id == 0 - no files in queue
  - 255 -there are files in queue ( id == 0)
  - -1 - file transfer not available
  - -2 - unknown error

**6.2.1.136 XGetFileTransferList()**

```
HRESULT ITrueConfCallX::XGetFileTransferList (
    [out, retval] BSTR * files )
```

Get list of all files in FileTransfer

## Parameters

<i>files</i>	- function return string in JSON format Each list item contains the following information id - file id isForSent - file transfer direction: true - sending, false - receiving fileName - file name, name without path while receiving, full path while sending caption - file caption peerId - TrueConf ID of sender (while receiving) downloadDir - folder for file download (while receiving) hash - file hash
--------------	---

Result JSON string sample (first item-sended file, second item-received file): {"files":[{"id":1,"isForSent":true,"fileName":"my\_file.txt","caption":"My sended File","peerId":"","downloadDir":"","hash":""}, {"id":2,"isForSent":false,"fileName":"my\_file.txt","caption":"My received file","peerId":"user@trueconf.com","downloadDir":"","hash":""}]}

## 6.2.1.137 XGetHandle()

```
HRESULT ITrueConfCallX::XGetHandle (
    [out, retval] ULONG * handle )
```

Get CallX window handle

## Returns

returns window handle, positive value, can be used for screen shot making

## 6.2.1.138 XGetMicByIndex()

```
HRESULT ITrueConfCallX::XGetMicByIndex (
    [in] LONG microphoneIdx,
    [out, retval] BSTR * microphoneName )
```

Get audio input name by index

## Parameters

<i>microphoneIdx</i>	- audio input device index in list
----------------------	------------------------------------

## See also

[Microphone](#)

## 6.2.1.139 XGetMicCount()

```
HRESULT ITrueConfCallX::XGetMicCount (
    [out, retval] LONG * microphonesCount )
```

Get audio input devices count in system

See also

[Microphone](#)

#### 6.2.1.140 XGetMicList()

```
HRESULT ITrueConfCallX::XGetMicList (
    [out, retval] BSTR * microphones )
```

Get audio input (microphone) device list

Returns

List of audio input device names that are present in system. Multilite text separated by "\r\n" newline symbols

See also

[Microphone](#)  
[XGetSpeakerList](#)  
[XGetCameraList](#)

#### 6.2.1.141 XGetSpeakerByIndex()

```
HRESULT ITrueConfCallX::XGetSpeakerByIndex (
    [in] LONG speakerIdx,
    [out, retval] BSTR * speakerName )
```

Get audio output name by index

Parameters

<i>speakerIdx</i>	- audio output device index in list
-------------------	-------------------------------------

See also

[Speaker](#)

#### 6.2.1.142 XGetSpeakerCount()

```
HRESULT ITrueConfCallX::XGetSpeakerCount (
    [out, retval] LONG * speakersCount )
```

Get audio output devices count in system

**See also**[Speaker](#)**6.2.1.143 XGetSpeakerList()**

```
HRESULT ITrueConfCallX::XGetSpeakerList (
    [out, retval] BSTR * speakers )
```

Get audio output input (speaker) device list

**Returns**

List of audio output device names that are present in system. Multilite text separated by "\r\n" newline symbols

**See also**

[Speaker](#)  
[XGetCameraList](#)  
[XGetMicList](#)

**6.2.1.144 XHangUpForAll()**

```
HRESULT ITrueConfCallX::XHangUpForAll ( )
```

Terminate conference for all participants. Moderator rights for current conference are required

**See also**

[hangUp](#)  
[hangUp2](#)

**6.2.1.145 XPrintPdf()**

```
HRESULT ITrueConfCallX::XPrintPdf (
    [in] BSTR filePath,
    [out, retval] LONG * retCode )
```

Print local file

**6.2.1.146 XRemotePrintPdf()**

```
HRESULT ITrueConfCallX::XRemotePrintPdf (
    [in] BSTR peerId,
    [in] LONG fileId,
    [out, retval] LONG * retCode )
```

Print file on conference party side, file should be previously uploaded by party.

## Parameters

<i>peer</i> ↔ <i>Id</i>	- TrueConf ID
----------------------------	---------------

## 6.2.1.147 XRemoveSlide()

```
HRESULT ITrueConfCallX::XRemoveSlide (
    [in] LONG imgIdx )
```

Deletes slide from slideshow

## Parameters

<i>imgIdx</i>	- slide index in list
---------------	-----------------------

## 6.2.1.148 XRestart()

```
HRESULT ITrueConfCallX::XRestart ( )
```

Restarts application window. Sends shutdown command and starts new instance without waiting for answer

## See also

XRestartForce

## 6.2.1.149 XSelectCamera()

```
HRESULT ITrueConfCallX::XSelectCamera (
    [in] BSTR cameraName )
```

Select camera by name

## Parameters

<i>cameraName</i>	- Camera name
-------------------	---------------

## See also

[Camera](#)  
[XDeselectCamera](#)

**6.2.1.150 XSelectMicByIndex()**

```
HRESULT ITrueConfCallX::XSelectMicByIndex (
    [in] LONG microphoneIdx )
```

Select audio input device by index

**Parameters**

<i>micIdx</i>	- audio input device index in list
---------------	------------------------------------

- audio input will be set to none if audio input with such index does not exist
- indexes start from 0.

**See also**

[Microphone](#)

**6.2.1.151 XSelectSpeakerByIndex()**

```
HRESULT ITrueConfCallX::XSelectSpeakerByIndex (
    [in] LONG speakerIdx )
```

Select audio output device by index

**Parameters**

<i>speakerIdx</i>	- audio output device index in list <ul style="list-style-type: none"> <li>• audio output will be set to none if audio output with such index does not exist</li> <li>• indexes start from 0.</li> </ul>
-------------------	--

**See also**

[Speaker](#)

**6.2.1.152 XSetBackgroundImage()**

```
HRESULT ITrueConfCallX::XSetBackgroundImage (
    [in] BSTR imgPath )
```

Set background image for main component window

## Parameters

<i>imgPath</i>	- full path to image file image must have resolution: 3840x2160 pixels
----------------	--

## 6.2.1.153 XSetCameraByIndex()

```
HRESULT ITrueConfCallX::XSetCameraByIndex (
    [in] LONG cameraIdx )
```

Select camera by index

## Parameters

<i>cameraIdx</i>	- camera index in list
------------------	------------------------

- camera will be set to none if camera with such index does not exist
- cameraIdx >= 0.

## 6.2.1.154 XStartPictureBroadcast()

```
HRESULT ITrueConfCallX::XStartPictureBroadcast (
    [in] BSTR imgPath )
```

Initiates picture broadcast in place of video from camera.

## Parameters

<i>imgPath</i>	- Full path of image file Image file must be in a directory 'Disk:/.../Id/Img.jpg' where 'id' - catalog with integer name - some file ID
----------------	--

## See also

[XStopPictureBroadcast](#)

## 6.2.1.155 XStopPictureBroadcast()

```
HRESULT ITrueConfCallX::XStopPictureBroadcast ( )
```

Stop picture broadcast, send video from camera

## See also

[XStartBroadcastPicture](#)

## 6.2.2 Property Documentation

### 6.2.2.1 CallXPath

BSTR ITrueConfCallX::CallXPath [get], [set]

Full path to executable used by TrueConf CallX. Initialized from registry by default. Used when executable is started.

### 6.2.2.2 Camera

BSTR ITrueConfCallX::Camera [get], [set]

Get current camera name

See also

- [XSelectCamera](#)
- [XDeselectCamera](#)
- [XGetCameraList](#)

### 6.2.2.3 Microphone

BSTR ITrueConfCallX::Microphone [get], [set]

Get audio input (microphone) device name

### 6.2.2.4 Speaker

BSTR ITrueConfCallX::Speaker [get], [set]

Get audio output (speaker) device name

### 6.2.2.5 XCommandLine

BSTR ITrueConfCallX::XCommandLine [get], [set]

Set additional application command line options

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

- [TrueConf\\_CallX.IDL](#)

## 6.3 TrueConf\_CallXLib::TrueConfCallX Class Reference

```
import "TrueConf_CallX.IDL";
```

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

- [TrueConf\\_CallX.IDL](#)



# Chapter 7

## File Documentation

### 7.1 TrueConf\_CallX.IDL File Reference

```
#include "olect1.h"
```

#### Classes

- interface [ITrueConfCallX](#)
- interface [TrueConf\\_CallXLib::\\_ITrueConfCallXEvents](#)
- class [TrueConf\\_CallXLib::TrueConfCallX](#)

#### Modules

- library [TrueConf\\_CallXLib](#)

#### Functions

- [TrueConf\\_CallXLib::importlib](#) ("stdole2.tlb")



# Index

- accept
  - [ITrueConfCallX, 36](#)
- acceptPeer
  - [ITrueConfCallX, 36](#)
- acceptPodiumInvite
  - [ITrueConfCallX, 36](#)
- acceptPodiumRequest
  - [ITrueConfCallX, 36](#)
- acceptPodiumRequest2
  - [ITrueConfCallX, 37](#)
- activateLicense
  - [ITrueConfCallX, 37](#)
- addContactToAbook
  - [ITrueConfCallX, 37](#)
- addToAbook
  - [ITrueConfCallX, 38](#)
- addToGroup
  - [ITrueConfCallX, 38](#)
- allowRecord
  - [ITrueConfCallX, 38](#)
- allowRecord2
  - [ITrueConfCallX, 39](#)
- appUpdate
  - [ITrueConfCallX, 39](#)
- block
  - [ITrueConfCallX, 39](#)
- call
  - [ITrueConfCallX, 39](#)
- CallXPath
  - [ITrueConfCallX, 76](#)
- Camera
  - [ITrueConfCallX, 76](#)
- changeVideoMatrix
  - [ITrueConfCallX, 40](#)
- chatClear
  - [ITrueConfCallX, 40](#)
- clearSlides
  - [ITrueConfCallX, 41](#)
- connectToServer
  - [ITrueConfCallX, 41](#)
- createConference
  - [ITrueConfCallX, 41](#)
- createGroup
  - [ITrueConfCallX, 42](#)
- extendUidTtl
  - [ITrueConfCallX, 42](#)
- firstSlide
  - [ITrueConfCallX, 42](#)
- forbidRecord
  - [ITrueConfCallX, 42](#)
- forbidRecord2
  - [ITrueConfCallX, 42](#)
- getAbook
  - [ITrueConfCallX, 43](#)
- getAppState
  - [ITrueConfCallX, 43](#)
- getAudioMute
  - [ITrueConfCallX, 43](#)
- getBanList
  - [ITrueConfCallX, 43](#)
- getCallHistory
  - [ITrueConfCallX, 43](#)
- getChatLastMessages
  - [ITrueConfCallX, 44](#)
- getContactDetails
  - [ITrueConfCallX, 44](#)
- getDisplayNameById
  - [ITrueConfCallX, 44](#)
- getGroups
  - [ITrueConfCallX, 45](#)
- getHardware
  - [ITrueConfCallX, 45](#)
- getHardwareKey
  - [ITrueConfCallX, 45](#)
- getIdListRequestInvite
  - [ITrueConfCallX, 45](#)
- getLicenseType
  - [ITrueConfCallX, 45](#)
- getListOfChats
  - [ITrueConfCallX, 45](#)
- getMaxConfTitleLength
  - [ITrueConfCallX, 46](#)
- getMicMute
  - [ITrueConfCallX, 46](#)
- getModes
  - [ITrueConfCallX, 46](#)
- getMonitorsInfo
  - [ITrueConfCallX, 46](#)
- getParticipantsList
  - [ITrueConfCallX, 46](#)
- getSelfieBroadcast
  - [ITrueConfCallX, 46](#)
- getSettings
  - [ITrueConfCallX, 47](#)
- getSlideShowInfo
  - [ITrueConfCallX, 47](#)

- getSystemInfo
  - ITrueConfCallX, 47
- getVideoMatrix
  - ITrueConfCallX, 47
- getVideoMute
  - ITrueConfCallX, 47
- gotoPodium
  - ITrueConfCallX, 47
- hangUp
  - ITrueConfCallX, 47
- hangUp2
  - ITrueConfCallX, 48
- ITrueConfCallX, 32
  - accept, 36
  - acceptPeer, 36
  - acceptPodiumInvite, 36
  - acceptPodiumRequest, 36
  - acceptPodiumRequest2, 37
  - activateLicense, 37
  - addContactToAbook, 37
  - addToAbook, 38
  - addToGroup, 38
  - allowRecord, 38
  - allowRecord2, 39
  - appUpdate, 39
  - block, 39
  - call, 39
  - CallXPath, 76
  - Camera, 76
  - changeVideoMatrix, 40
  - chatClear, 40
  - clearSlides, 41
  - connectToServer, 41
  - createConference, 41
  - createGroup, 42
  - extendUidTtl, 42
  - firstSlide, 42
  - forbidRecord, 42
  - forbidRecord2, 42
  - getAbook, 43
  - getAppState, 43
  - getAudioMute, 43
  - getBanList, 43
  - getCallHistory, 43
  - getChatLastMessages, 44
  - getContactDetails, 44
  - getDisplayNameById, 44
  - getGroups, 45
  - getHardware, 45
  - getHardwareKey, 45
  - getIdListRequestInvite, 45
  - getLicenseType, 45
  - getListOfChats, 45
  - getMaxConfTitleLength, 46
  - getMicMute, 46
  - getModes, 46
  - getMonitorsInfo, 46
  - getParticipantsList, 46
  - getSelfieBroadcast, 46
  - getSettings, 47
  - getSlideShowInfo, 47
  - getSystemInfo, 47
  - getVideoMatrix, 47
  - getVideoMute, 47
  - gotoPodium, 47
  - hangUp, 47
  - hangUp2, 48
  - inviteToConference, 48
  - inviteToPodium, 49
  - lastSlide, 49
  - leavePodium, 49
  - login, 49
  - logout, 49
  - Microphone, 76
  - moveSlideDown, 50
  - moveSlideUp, 50
  - nextSlide, 50
  - prevSlide, 50
  - ptzDown, 50
  - ptzLeft, 51
  - ptzRight, 51
  - ptzStop, 51
  - ptzUp, 51
  - ptzZoomIn, 51
  - ptzZoomOut, 51
  - reject, 51
  - rejectInvitationToPodium, 52
  - rejectPeer, 52
  - rejectRequestToPodium, 52
  - rejectRequestToPodium2, 52
  - removeContactFromAbook, 53
  - removeFromAbook, 53
  - removeFromGroup, 53
  - removeFromPodium, 54
  - removeGroup, 54
  - removePeer, 54
  - renameGroup, 55
  - renameInAbook, 55
  - sendCommand, 55
  - sendGroupMessage, 56
  - sendMessage, 56
  - setAudioMute, 57
  - setHardware, 57
  - setLastCallsViewed, 57
  - setMicMute, 57
  - setModes, 57
  - setSelfieBroadcast, 57
  - setSettings, 58
  - setVideoMute, 58
  - showSlide, 58
  - shutdown, 58
  - shutdown2, 58
  - Speaker, 76
  - startAudioRemark, 59
  - startScreenCapture, 59

- startSlideShow, [59](#)
- stopScreenCapture, [59](#)
- stopSlideShow, [59](#)
- switchVideoStream, [60](#)
- takePicture, [60](#)
- testAudioCapterStart, [60](#)
- testAudioCapterStop, [60](#)
- testAudioRenderer, [61](#)
- toneDial, [61](#)
- turnRemoteCamera, [61](#)
- turnRemoteMic, [62](#)
- unblock, [62](#)
- XAddSlide, [62](#)
- XAddSlidesDialog, [63](#)
- XCheckForEvent, [63](#)
- XCommandLine, [76](#)
- XDeselectCamera, [63](#)
- XDeselectMic, [63](#)
- XDeselectSpeaker, [64](#)
- XEncryptedLogin, [64](#)
- XExecuteCommand, [64](#)
- XFileAccept, [65](#)
- XFileDelete, [65](#)
- XFileReject, [65](#)
- XFileSend, [66](#)
- XFileSendAbort, [66](#)
- XFileSendToConference, [66](#)
- XFileStatsInfo, [67](#)
- XFileTransferAvailable, [67](#)
- XForceRestart, [67](#)
- XGetCallXProcessId, [67](#)
- XGetCameraByIndex, [67](#)
- XGetCameraCount, [68](#)
- XGetCameraList, [68](#)
- XGetFileDownloadPath, [68](#)
- XGetFileHashById, [68](#)
- XGetFileIdByHash, [69](#)
- XGetFileSendStatus, [69](#)
- XGetFileTransferList, [69](#)
- XGetHandle, [70](#)
- XGetMicByIndex, [70](#)
- XGetMicCount, [70](#)
- XGetMicList, [71](#)
- XGetSpeakerByIndex, [71](#)
- XGetSpeakerCount, [71](#)
- XGetSpeakerList, [72](#)
- XHangUpForAll, [72](#)
- XPrintPdf, [72](#)
- XRemotePrintPdf, [72](#)
- XRemoveSlide, [73](#)
- XRestart, [73](#)
- XSelectCamera, [73](#)
- XSelectMicByIndex, [73](#)
- XSelectSpeakerByIndex, [74](#)
- XSetBackgroundImage, [74](#)
- XSetCameraByIndex, [75](#)
- XStartPictureBroadcast, [75](#)
- XStopPictureBroadcast, [75](#)
- importlib
  - TrueConf\_CallXLib, [9](#)
- inviteToConference
  - ITrueConfCallX, [48](#)
- inviteToPodium
  - ITrueConfCallX, [49](#)
- lastSlide
  - ITrueConfCallX, [49](#)
- leavePodium
  - ITrueConfCallX, [49](#)
- login
  - ITrueConfCallX, [49](#)
- logout
  - ITrueConfCallX, [49](#)
- Microphone
  - ITrueConfCallX, [76](#)
- moveSlideDown
  - ITrueConfCallX, [50](#)
- moveSlideUp
  - ITrueConfCallX, [50](#)
- nextSlide
  - ITrueConfCallX, [50](#)
- OnAbookUpdate
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [13](#)
- OnAppUpdateAvailable
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [13](#)
- OnAudioCapterRmsLevelUpdated
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [13](#)
- OnBroadcastPictureStateChanged
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [13](#)
- OnCallHistoryUpdated
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [14](#)
- OnChangeVideoMatrixReport
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [14](#)
- OnChatMessageSent
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [14](#)
- OnCmdAddToAbook
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [14](#)
- OnCmdAddToGroup
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [15](#)
- OnCmdBlock
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [15](#)
- OnCmdChatClear
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [15](#)
- OnCmdCreateGroup
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [16](#)
- OnCmdRemoveFromAbook
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [16](#)
- OnCmdRemoveFromGroup
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [16](#)
- OnCmdRemoveGroup
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [17](#)
- OnCmdRenameGroup
  - TrueConf\_CallXLib::ITrueConfCallXEvents, [17](#)
- OnCmdRenameInAbook

- TrueConf\_CallXLib::\_ITrueConfCallXEvents, 17
- OnCmdUnblock
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 18
- OnCommandReceived
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 18
- OnCommandSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 18
- OnConferenceCreated
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnConferenceDeleted
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnContactBlocked
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnContactDeleted
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnContactUnblocked
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnDetailInfo
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 19
- OnDeviceModesDone
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 20
- OnFileAccepted
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 20
- OnFileConferenceSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 20
- OnFileRejected
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 20
- OnFileSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 21
- OnFileTransferAvailable
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 21
- OnGroupChatMessageSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 21
- OnGroupsUpdate
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 22
- OnHangUpPressed
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 22
- OnHardwareChanged
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 22
- OnIncomingChatMessage
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 22
- OnIncomingGroupChatMessage
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 22
- OnIncomingRequestToPodiumAnswered
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 23
- OnInviteReceived
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 23
- OnInviteRequestSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 23
- OnInviteSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 23
- OnJabraHangUpPressed
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 23
- OnJabraHookOffPressed
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnLogin
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnLogout
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnOffHookPressed
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnReceiversInfoUpdated
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnRecordRequest
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnRecordRequestReply
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 24
- OnRejectReceived
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnRejectSent
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnRemarkCountDown
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnRequestInviteReceived
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnRestrictionsChanged
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnRoleChanged
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnSelfSSInfoUpdate
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 25
- OnServerConnected
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnServerDisconnected
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnSettingsChanged
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnSlideShowInfoUpdate
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnSlideShowStart
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnSlideShowStop
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 26
- OnStart
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 27
- OnStopCalling
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 27
- OnTestAudioCatcherStateUpdated
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 27
- OnToneDial
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 27
- OnUpdateAvatar
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 27
- OnUpdateCameraInfo
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 28
- OnUpdateParticipantList
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 28
- OnVideoMatrixChanged
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 28
- OnXAfterStart
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 28
- OnXChangeState
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 28
- OnXCommandExecution
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 29
- OnXError
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 29

- TrueConf\_CallXLib::\_ITrueConfCallXEvents, 29
- OnXFileReceive
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 29
- OnXFileReceiveProgress
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 30
- OnXFileRequestReceived
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 30
- OnXFileSend
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 30
- OnXFileSendError
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 30
- OnXFileStatusChange
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 30
- OnXLogin
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 31
- OnXLoginError
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 31
- OnXNotify
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 32
- OnXStartFail
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 32
- OnXTerminate
  - TrueConf\_CallXLib::\_ITrueConfCallXEvents, 32
- prevSlide
  - ITrueConfCallX, 50
- ptzDown
  - ITrueConfCallX, 50
- ptzLeft
  - ITrueConfCallX, 51
- ptzRight
  - ITrueConfCallX, 51
- ptzStop
  - ITrueConfCallX, 51
- ptzUp
  - ITrueConfCallX, 51
- ptzZoomIn
  - ITrueConfCallX, 51
- ptzZoomOut
  - ITrueConfCallX, 51
- reject
  - ITrueConfCallX, 51
- rejectInvitationToPodium
  - ITrueConfCallX, 52
- rejectPeer
  - ITrueConfCallX, 52
- rejectRequestToPodium
  - ITrueConfCallX, 52
- rejectRequestToPodium2
  - ITrueConfCallX, 52
- removeContactFromAbook
  - ITrueConfCallX, 53
- removeFromAbook
  - ITrueConfCallX, 53
- removeFromGroup
  - ITrueConfCallX, 53
- removeFromPodium
  - ITrueConfCallX, 54
- removeGroup
  - ITrueConfCallX, 54
- removePeer
  - ITrueConfCallX, 54
- renameGroup
  - ITrueConfCallX, 55
- renameInAbook
  - ITrueConfCallX, 55
- sendCommand
  - ITrueConfCallX, 55
- sendGroupMessage
  - ITrueConfCallX, 56
- sendMessage
  - ITrueConfCallX, 56
- setAudioMute
  - ITrueConfCallX, 57
- setHardware
  - ITrueConfCallX, 57
- setLastCallsViewed
  - ITrueConfCallX, 57
- setMicMute
  - ITrueConfCallX, 57
- setModes
  - ITrueConfCallX, 57
- setSelfieBroadcast
  - ITrueConfCallX, 57
- setSettings
  - ITrueConfCallX, 58
- setVideoMute
  - ITrueConfCallX, 58
- showSlide
  - ITrueConfCallX, 58
- shutdown
  - ITrueConfCallX, 58
- shutdown2
  - ITrueConfCallX, 58
- Speaker
  - ITrueConfCallX, 76
- startAudioRemark
  - ITrueConfCallX, 59
- startScreenCapture
  - ITrueConfCallX, 59
- startSlideShow
  - ITrueConfCallX, 59
- stopScreenCapture
  - ITrueConfCallX, 59
- stopSlideShow
  - ITrueConfCallX, 59
- switchVideoStream
  - ITrueConfCallX, 60
- takePicture
  - ITrueConfCallX, 60
- testAudioCapterStart
  - ITrueConfCallX, 60
- testAudioCapterStop
  - ITrueConfCallX, 60
- testAudioRenderer

- ITrueConfCallX, 61
- toneDial
  - ITrueConfCallX, 61
- TrueConf\_CallX.IDL, 77
- TrueConf\_CallXLib, 9
  - importlib, 9
- TrueConf\_CallXLib::ITrueConfCallXEvents, 11
  - OnAbookUpdate, 13
  - OnAppUpdateAvailable, 13
  - OnAudioCapturerRmsLevelUpdated, 13
  - OnBroadcastPictureStateChanged, 13
  - OnCallHistoryUpdated, 14
  - OnChangeVideoMatrixReport, 14
  - OnChatMessageSent, 14
  - OnCmdAddToAbook, 14
  - OnCmdAddToGroup, 15
  - OnCmdBlock, 15
  - OnCmdChatClear, 15
  - OnCmdCreateGroup, 16
  - OnCmdRemoveFromAbook, 16
  - OnCmdRemoveFromGroup, 16
  - OnCmdRemoveGroup, 17
  - OnCmdRenameGroup, 17
  - OnCmdRenameInAbook, 17
  - OnCmdUnblock, 18
  - OnCommandReceived, 18
  - OnCommandSent, 18
  - OnConferenceCreated, 19
  - OnConferenceDeleted, 19
  - OnContactBlocked, 19
  - OnContactDeleted, 19
  - OnContactUnblocked, 19
  - OnDetailInfo, 19
  - OnDeviceModesDone, 20
  - OnFileAccepted, 20
  - OnFileConferenceSent, 20
  - OnFileRejected, 20
  - OnFileSent, 21
  - OnFileTransferAvailable, 21
  - OnGroupChatMessageSent, 21
  - OnGroupsUpdate, 22
  - OnHangUpPressed, 22
  - OnHardwareChanged, 22
  - OnIncomingChatMessage, 22
  - OnIncomingGroupChatMessage, 22
  - OnIncomingRequestToPodiumAnswered, 23
  - OnInviteReceived, 23
  - OnInviteRequestSent, 23
  - OnInviteSent, 23
  - OnJabraHangUpPressed, 23
  - OnJabraHookOffPressed, 24
  - OnLogin, 24
  - OnLogout, 24
  - OnOffHookPressed, 24
  - OnReceiversInfoUpdated, 24
  - OnRecordRequest, 24
  - OnRecordRequestReply, 24
  - OnRejectReceived, 25
  - OnRejectSent, 25
  - OnRemarkCountDown, 25
  - OnRequestInviteReceived, 25
  - OnRestrictionsChanged, 25
  - OnRoleChanged, 25
  - OnSelfSSInfoUpdate, 25
  - OnServerConnected, 26
  - OnServerDisconnected, 26
  - OnSettingsChanged, 26
  - OnSlideShowInfoUpdate, 26
  - OnSlideShowStart, 26
  - OnSlideShowStop, 26
  - OnStart, 27
  - OnStopCalling, 27
  - OnTestAudioCapturerStateUpdated, 27
  - OnToneDial, 27
  - OnUpdateAvatar, 27
  - OnUpdateCameraInfo, 28
  - OnUpdateParticipantList, 28
  - OnVideoMatrixChanged, 28
  - OnXAfterStart, 28
  - OnXChangeState, 28
  - OnXCommandExecution, 29
  - OnXError, 29
  - OnXFileReceive, 29
  - OnXFileReceiveProgress, 30
  - OnXFileRequestReceived, 30
  - OnXFileSend, 30
  - OnXFileSendError, 30
  - OnXFileStatusChange, 30
  - OnXLogin, 31
  - OnXLoginError, 31
  - OnXNotify, 32
  - OnXStartFail, 32
  - OnXTerminate, 32
- TrueConf\_CallXLib::TrueConfCallX, 76
- turnRemoteCamera
  - ITrueConfCallX, 61
- turnRemoteMic
  - ITrueConfCallX, 62
- unblock
  - ITrueConfCallX, 62
- XAddSlide
  - ITrueConfCallX, 62
- XAddSlidesDialog
  - ITrueConfCallX, 63
- XCheckForEvent
  - ITrueConfCallX, 63
- XCommandLine
  - ITrueConfCallX, 76
- XDeselectCamera
  - ITrueConfCallX, 63
- XDeselectMic
  - ITrueConfCallX, 63
- XDeselectSpeaker
  - ITrueConfCallX, 64
- XEncryptedLogin



- ITrueConfCallX, 64
- XExecuteCommand
  - ITrueConfCallX, 64
- XFileAccept
  - ITrueConfCallX, 65
- XFileDelete
  - ITrueConfCallX, 65
- XFileReject
  - ITrueConfCallX, 65
- XFileSend
  - ITrueConfCallX, 66
- XFileSendAbort
  - ITrueConfCallX, 66
- XFileSendToConference
  - ITrueConfCallX, 66
- XFileStatsInfo
  - ITrueConfCallX, 67
- XFileTransferAvailable
  - ITrueConfCallX, 67
- XForceRestart
  - ITrueConfCallX, 67
- XGetCallXProcessId
  - ITrueConfCallX, 67
- XGetCameraByIndex
  - ITrueConfCallX, 67
- XGetCameraCount
  - ITrueConfCallX, 68
- XGetCameraList
  - ITrueConfCallX, 68
- XGetFileDownloadPath
  - ITrueConfCallX, 68
- XGetFileHashById
  - ITrueConfCallX, 68
- XGetFileIdByHash
  - ITrueConfCallX, 69
- XGetFileSendStatus
  - ITrueConfCallX, 69
- XGetFileTransferList
  - ITrueConfCallX, 69
- XGetHandle
  - ITrueConfCallX, 70
- XGetMicByIndex
  - ITrueConfCallX, 70
- XGetMicCount
  - ITrueConfCallX, 70
- XGetMicList
  - ITrueConfCallX, 71
- XGetSpeakerByIndex
  - ITrueConfCallX, 71
- XGetSpeakerCount
  - ITrueConfCallX, 71
- XGetSpeakerList
  - ITrueConfCallX, 72
- XHangUpForAll
  - ITrueConfCallX, 72
- XPrintPdf
  - ITrueConfCallX, 72
- XRemotePrintPdf
  - ITrueConfCallX, 72
- XRemoveSlide
  - ITrueConfCallX, 73
- XRestart
  - ITrueConfCallX, 73
- XSelectCamera
  - ITrueConfCallX, 73
- XSelectMicByIndex
  - ITrueConfCallX, 73
- XSelectSpeakerByIndex
  - ITrueConfCallX, 74
- XSetBackgroundImage
  - ITrueConfCallX, 74
- XSetCameraByIndex
  - ITrueConfCallX, 75
- XStartPictureBroadcast
  - ITrueConfCallX, 75
- XStopPictureBroadcast
  - ITrueConfCallX, 75