

iConf .NET Video Conferencing SDK

Table of Contents

Overview	1
Installing the iConf .NET SDK	2
Getting Started	5
IP to IP Mode	5
Using the Central Server Service	8
Choosing a Database	8
Using the Embedded Database	9
Using a SQL Server Database	9
Database Scripts	10
MYSQL SCRIPTS	10
SQL SERVER SCRIPTS	11
Using a MySQL Database	12
Configure Listening Ports	13
Starting the Central Server	14
Registering a new user	15
Logging in	16
Logging out	17
Adding a new Contact	18
Accepting/Rejecting a contact request	19
Viewing Pending Contact Request	19
Starting a conference with 1 or more users	21
Accepting a conference request	21
Viewing a user (video stream)	22
Updating your Profile picture	23
Sharing Files	23
Symbol Reference	25
iConfServer.NET Namespace	25
Classes	25
iConfServerDotNet Class	25
iConfServerDotNet.iConfServerDotNet Constructor	45
iConfServerDotNet Structures	45
iConfServerDotNet Enumerations	46

iConfServerDotNet Fields	48
iConfServerDotNet Methods	48
iConfServerDotNet Properties	87
iConfServerDotNet Delegates	88
iConfServerDotNet Events	99
Interfaces	106
ICallback Interface	106
ICallback Methods	107
iConfClient.NET Namespace	111
Classes	111
iConfClientDotNet Class	112
iConfClientDotNet.iConfClientDotNet Constructor	118
iConfClientDotNet Enumerations	118
iConfClientDotNet Fields	119
iConfClientDotNet Methods	119
iConfClientDotNet Properties	126
iConfClientDotNet Delegates	127
iConfClientDotNet Events	129
Interfaces	131
ICallback Interface	132
ICallback Methods	132
IComEvents Interface	133
IComEvents Methods	133
IOBJECTSafety Interface	134
IOBJECTSafety Methods	134

Index

b

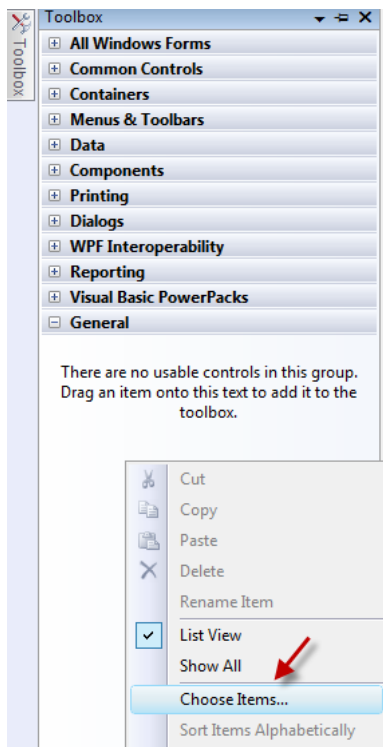
1 Overview

The iConf .NET Video Conferencing SDK is an easy to use Video Conferencing SDK that allows .NET developers to integrate Video Conferencing features into an existing application or create video conferencing applications from scratch without having to tangle with complex communication or video APIs.

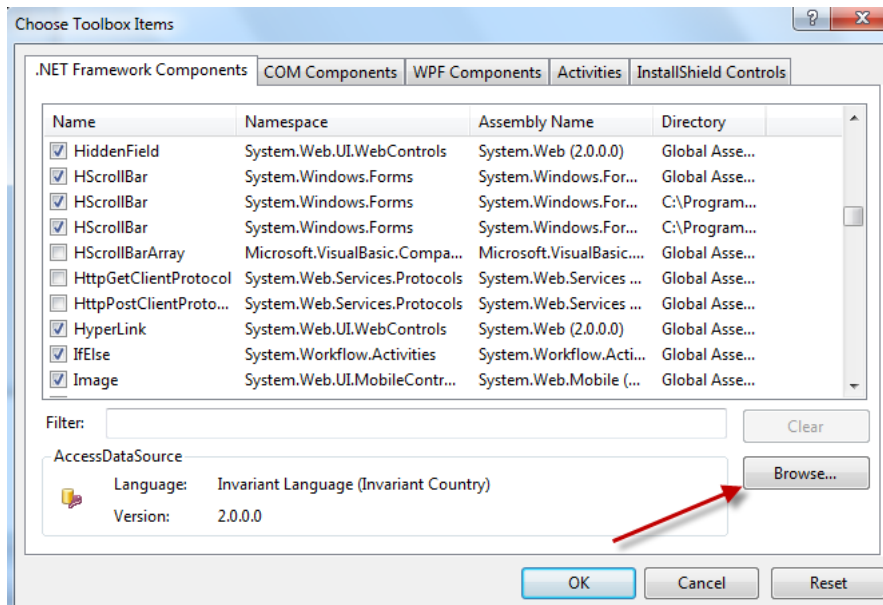
2 Installing the iConf .NET SDK

Installing the iConf.NET SDK

1. Download and run the Installer
2. Open Visual Studio .NET
3. Open the project where you would like to use the iConf.NET SDK components
4. Right click in a section of the Toolbox and Select Choose Items..



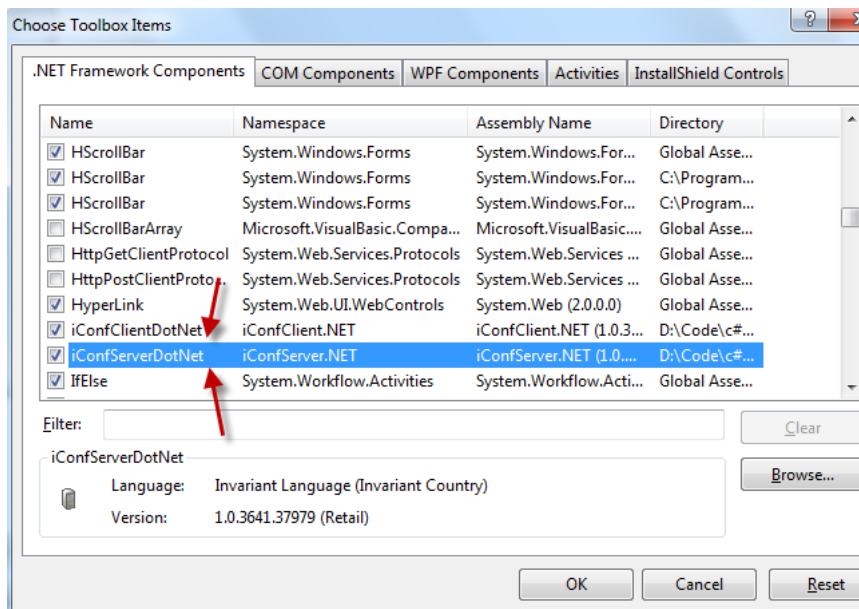
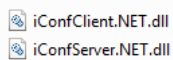
In the Choose Toolbox items dialog click on Browse



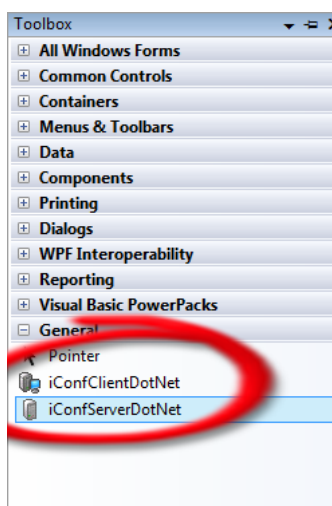
Go to the C:\PROGRAM FILES\AVSPEED Applications\iConf.NET SDK\Components

select the iConfClient.Net.dll

click on Open, then click again on Browse and this time, select the iConfServer.Net.dll file



Click on OK and both components will appear in your Toolbox, ready to be dropped on your forms and used in your application.



3 Getting Started

3.1 IP to IP Mode

Using the iConf SDK to create an IP to IP video conferencing application

This tutorial will show you how to create a simple IP to IP video conferencing application using the iConf Server .NET and iConf Client .NET components.

Example

Language : C#

Step 1 :

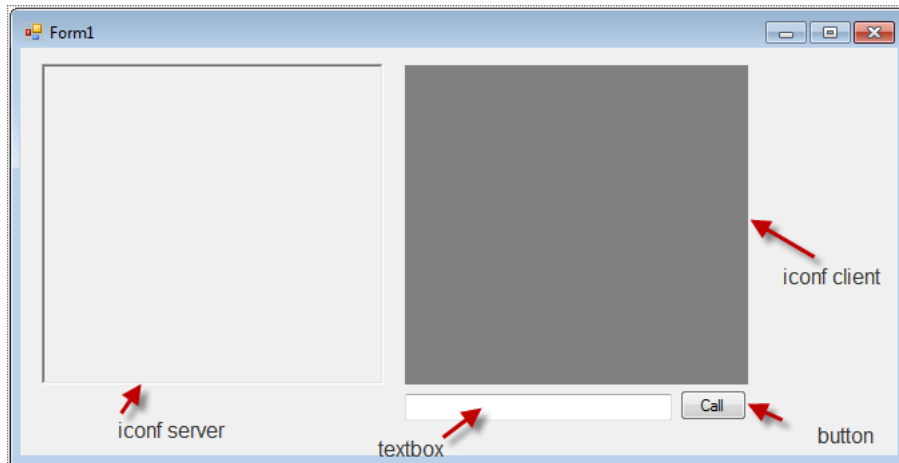
As shown below, drag

1 iconf server component (name it ics)

1 iconf client component (name it icc)

1 textbox (name it txtip)

1 button (name it btnCall)



Take a look at the code sample below and attach all events as described.

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Collections;
using iConfServer.NET.HelperClasses;
```

```
namespace IP2IPDemo
```

```

{
    public partial class Form1 : Form
    {
        /// <summary>
        /// THIS APPLICATION NEEDS RUN THIS APPLICATION NEEDS TO RUN ON 2 DIFFERENT PCS
        /// (OR Virtual Machines)
        /// OR CAN BE USED IN CALLBACK mode by using the local ip when placing a call
        /// This is a simple demo which uses basic settings
        /// For help email us at support@avspeed.com
        /// </summary>
        public Form1()
        {
            InitializeComponent();
            txtIP.Text = ics.GetLocalIp()[0].ToString();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                ics.InitializeAudioSystem(iConfServer.NET.iConfServerDotNet.audioType.Direct
Sound,
-1, -1, 16000, 10);

                //select the first available video device
                ics.SelectVideoDevice(0);

                ArrayList lst = ics.GetVideoSizes();

                //start previewing video using the default video size
                ics.StartPreview(0);
            }
            catch(Exception ex)
            {
                MessageBox.Show(string.Format("Error initializing {0}", ex.Message));
            }
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            //stop listening
            ics.Listen(false, ics.GetLocalIp()[0].ToString(), 9990, 17860, 17861);

            //stop the video preview
            ics.StopPreview();
        }

        private void btnCall_Click(object sender, EventArgs e)
        {
            string myIp = ics.GetLocalIp()[0].ToString();
            string ipToCall = txtIP.Text;

            const int videoPort = 9990;
            const int audioTcpPort = 17860;
            const int audioUdpPort = 17861;

            //place a call to an iConf Server
            //note when using the Call function we have the ability to supply callback
parameters
            //which will help the peer connection call us back
            icc.AutoReconnect = true;
            icc.Call(ipToCall, videoPort, 0, 0, "test", ics.CallBackId, myIp, videoPort,
audioTcpPort, audioUdpPort, "");
        }
    }
}

```

```

        private void ics_IncomingCall(object sender, string authenticationData, int
socketHandle, string callbackid, string callbackipaddress, int callbackvideoport, int
callbackaudiocport, int callbackaudiopport)
        {
            Invoke(new MethodInvoker(delegate
            {
                //accept the incoming call
                ics.AcceptCall("n/a", socketHandle);

                //call back to have a 1 on one video conference
                icc.Call(callbackipaddress, callbackvideoport, 0, 0, "n/a", callbackid,
                    ics.GetLocalIp()[0].ToString(), 0, 0, 0, "");
            }));
        }

        private void ics_ClientDisconnected(object sender, string ipAddress, int port)
        {
            Invoke(new MethodInvoker(delegate
            {
                //peer has disconnected
                //disconnect our client as well
                icc.Disconnect();
            }));
        }

        private void ics_CannotRunGraph()
        {
            MessageBox.Show("Video Capture device might be in use");
        }

        private void btnHangUp_Click(object sender, EventArgs e)
        {
            icc.Disconnect();
            icc.ClearImage();
        }

        private void ics_VideoPreviewStarted(int videoWidth, int videoHeight, string
deviceName)
        {
            // this is where we are going to initialize the video encoder -
            // note that this is a change from previous versions

            Invoke(new MethodInvoker(delegate
            {
                this.Text = string.Format("Simple IP 2 IP Demo - {0}x{1}", videoWidth,
videoHeight);

                ics.VideoRateControl =
iConfServer.NET.iConfServerDotNet.RateControl.ConstantQuality;
                //Initialize Codec
                ics.SetEncoderProperties(VideoCodecs.MPEG4, 20, 8000, 0, 0, 0);

                if (!ics.IsListening)
                //listen for incoming connections
                ics.Listen(true, ics.GetLocalIp()[0].ToString(), 9990, 17860, 17861);
            }));
        }
    }
}

```

You will need 2 computers to run this sample application

Run 1 instance of the application on computer A and the other instance on computer B

Type the IP address of computer B in the text box and click on the Call button.

Wait for computer B to receive and accept your call

3.2 Using the Central Server Service

The iConf Central Server Service (iCS) is a Windows Service Application that acts as a directory service for the iConf Client.NET/Server.NET components. The iCS is necessary when one iConf Client.NET/Server.NET is trying to contact a host who has a private IP address internal to a local area network that is inaccessible to the outside world, or when the host is blocked by a firewall. The iConf Server.NET component registers users with the iCS, logs into the iCS, can add, remove contacts and place calls to users in its contact list.

The iConf Central Server Service cannot be behind a router or firewall.

The iConf Central Server Service can maintain a list of registered users in an Embedded database, a MySQL Database or a Microsoft SQL Server Database.

3.2.1 Choosing a Database

The iConf Central Server Service maintains registered users in a database.

That database can either be :

- an Embedded single file Database pre-installed with the service)
- a MySQL Database
- a SQL Server Database

To configure the iConf Central Service Service to use a specific database, you have to modify its configuration file which can be found by default in

C:\Program Files\iConfSDK.NET\CentralServer

The name of the configuration file is **CentralServerNET.exe.config**

3.2.2 Using the Embedded Database

The iConf Central Server Service comes preinstalled with an embedded Database which makes it easy to get up and running without having to install a full blown database system.

To configure the iConf Central Server service to use the Embedded Database, open the Central Service Configuration File in a text editor.

The Central Service Configuration File can be found in C:\Program Files\iConf.NET SDK\Central Server

Next, locate the "connectionString" attribute of the "EmbeddedConnectionString" key and set the DataSource to the path where the Embedded Database file can be found.

By the default, the Embedded Database file is located in the Central Server folder in your target installation directory.

Set the DBType setting value to "EMBEDDED" as shown below

Save the configuration file and start (or restart) the Central Server service.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="SQLServerConnectionString" connectionString="Data
Source=AVSPEED\AVSPEED;Initial Catalog=usersDB;Integrated Security=True" />
    <add name="EmbeddedConnectionString" connectionString="Data
Source=|DataDirectory|\users.vdb3" providerName="VistaDB.NET20" />
    <add name="MySQLConnectionString" connectionString="server=localhost;user
id=test;Password=test;persist security info=True;database=testdb"
providerName="MySQL.Data.MySqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="DBType" value="EMBEDDED" />
    <add key="EnableLog" value="False" />
    <add key="TCPCentralListeningPort" value="8081" />
    <add key="TCPAudioListeningPort" value="17860" />
    <add key="UDPAudioDataPort" value="17861" />
    <add key="ReflectorTcpPortStart" value="20000" />
    <add key="TCPCentralListeningIP" value="0.0.0.0" />
    <add key="TCPAdministrationProfilesPort" value="8082" />
    <add key="LoginMessage" value="Welcome to the Central Server." />
    <add key="RegistrationMessage" value="Thank you for registering." />
  </appSettings>
</configuration>
```

3.2.3 Using a SQL Server Database

To configure the iConf Central Server service to use a SQL Server Database, open the Central Service Configuration File in a text editor.

The Central Service Configuration File can be found in C:\Program Files\iConfSDK.NET\Central Server

Next, locate the "connectionString" attribute of the "SQLServerConnectionString"

```
connectionString="Data Source=the database server;Initial Catalog=the database name;Integrated Security=True" />
```

Set the DBType setting value to "MSSQL" as shown below

Save the configuration file and start (or restart) the iConf Central Server Service.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="SQLServerConnectionString" connectionString="Data
Source=AVSPEED\AVSPEED;Initial Catalog=usersDB;Integrated Security=True" />
    <add name="EmbeddedConnectionString" connectionString="Data
Source=|DataDirectory|\users.vdb3" providerName="VistaDB.NET20" />
    <add name="MySQLConnectionString" connectionString="server=localhost;user
id=test;Password=test;persist security info=True;database=testdb"
providerName="MySQL.Data.MySqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="DBType" value="MSSQL" />
    <add key="EnableLog" value="False" />
    <add key="TCPCentralListeningPort" value="8081" />
    <add key="TCPAudioListeningPort" value="17860" />
    <add key="UDPAudioDataPort" value="17861" />
    <add key="ReflectorTcpPortStart" value="20000" />
    <add key="TCPCentralListeningIP" value="0.0.0.0" />
    <add key="TCPAdministrationProfilesPort" value="8082" />
    <add key="LoginMessage" value="Welcome to the Central Server." />
    <add key="RegistrationMessage" value="Thank you for registering." />
  </appSettings>
</configuration>
```

3.2.4 Database Scripts

Database scripts

Please run the following scripts depending on the Database that you choose to use with the iConf Central Server Service

SQL SERVER DATABASE (see page 11)

MYSQL DATABASE (see page 10)

3.2.4.1 MYSQL SCRIPTS

MySQL SCRIPTS

```
/*
MySQL - 5.1.41-community
*****
*/

/*!40101 SET NAMES utf8 */;
```

```

/*!40101 SET SQL_MODE='';

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
/*Table structure for table `iccontact` */

CREATE TABLE `iccontact` (
  `UserId` int(11) NOT NULL DEFAULT '0',
  `ContactId` int(11) NOT NULL DEFAULT '0',
  `Confirmed` tinyint(1) NOT NULL DEFAULT '0',
  `OriginatorId` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`UserId`,`ContactId`),
  KEY `fkUserId` (`UserId`,`ContactId`,`OriginatorId`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

/*Table structure for table `icuser` */

CREATE TABLE `icuser` (
  `UserId` int(11) NOT NULL AUTO_INCREMENT,
  `UserName` varchar(100) NOT NULL DEFAULT '',
  `Password` varchar(100) NOT NULL DEFAULT '',
  `LastLoginDate` datetime DEFAULT NULL,
  `UserExtraData` text NOT NULL,
  `UserStatus` char(3) DEFAULT NULL,
  PRIMARY KEY (`UserId`,`UserName`),
  KEY `UserId` (`UserId`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

3.2.4.2 SQL SERVER SCRIPTS

SQL SERVER SCRIPTS

```

GO
/***** Object: Table [dbo].[icuser]      Script Date: 01/24/2010 00:07:25 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[icuser](
    [UserId] [int] IDENTITY(1,1) NOT NULL,
    [UserName] [varchar](100) NOT NULL,
    [Password] [varchar](100) NOT NULL,
    [LastLoginDate] [datetime] NULL,
    [UserExtraData] [varchar](1000) NULL,
    [UserStatus] [varchar](3) NULL,
    CONSTRAINT [PK_icuser_1] PRIMARY KEY CLUSTERED
(
    [UserId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[iccontact]    Script Date: 01/24/2010 00:07:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE TABLE [dbo].[iccontact](
    [UserId] [int] NOT NULL,
    [ContactId] [int] NOT NULL,
    [Confirmed] [bit] NOT NULL CONSTRAINT [DF_iccontact_Confirmed] DEFAULT ((0)),
    [OriginatorId] [int] NOT NULL,
    CONSTRAINT [PK_iccontact] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [ContactId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: ForeignKey [FK_iccontact_icuser]    Script Date: 01/24/2010 00:07:22
*****/
ALTER TABLE [dbo].[iccontact] WITH CHECK ADD CONSTRAINT [FK_iccontact_icuser] FOREIGN
KEY([UserId])
REFERENCES [dbo].[icuser] ([UserId])
GO
ALTER TABLE [dbo].[iccontact] CHECK CONSTRAINT [FK_iccontact_icuser]
GO
/***** Object: ForeignKey [FK_iccontact_icuser1]    Script Date: 01/24/2010 00:07:22
*****/
ALTER TABLE [dbo].[iccontact] WITH CHECK ADD CONSTRAINT [FK_iccontact_icuser1] FOREIGN
KEY([ContactId])
REFERENCES [dbo].[icuser] ([UserId])
GO
ALTER TABLE [dbo].[iccontact] CHECK CONSTRAINT [FK_iccontact_icuser1]
GO
/***** Object: ForeignKey [FK_iccontact_icuser2]    Script Date: 01/24/2010 00:07:22
*****/
ALTER TABLE [dbo].[iccontact] WITH CHECK ADD CONSTRAINT [FK_iccontact_icuser2] FOREIGN
KEY([OriginatorId])
REFERENCES [dbo].[icuser] ([UserId])
GO
ALTER TABLE [dbo].[iccontact] CHECK CONSTRAINT [FK_iccontact_icuser2]
GO

```

3.2.5 Using a MySQL Database

To configure the iConf Central Server service to use a MySQL Database, open the Central Service Configuration File in a text editor.

The Central Service Configuration File can be found in C:\Program Files\iConf.NET SDK\Central Server

Next, locate the "connectionString" attribute of the "MySqlConnectionString"

connectionString="server=*the address of the server*;user id=*the user id*;Password=*the password*;persist security info=True;database=*the name of the database where the central server tables were created*"

Set the DBType setting value to "MYSQL" as shown below

Save the configuration file and start (or restart) the Central Server service.

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>

```



```
</configSections>
<connectionStrings>
  <add name="SQLServerConnectionString" connectionString="Data
Source=AVSPEED\AVSPEED;Initial Catalog=usersDB;Integrated Security=True" />
  <add name="EmbeddedConnectionString" connectionString="Data
Source=|DataDirectory|\users.vdb3" providerName="VistaDB.NET20" />
  <add name="MySQLConnectionString" connectionString="server=localhost;user
id=test;Password=test;persist security info=True;database=testdb"
providerName="MySQL.Data.MySqlClient" />
</connectionStrings>
<appSettings>
  <add key="DBType" value="MYSQL" />
  <add key="EnableLog" value="False" />
  <add key="TCPCentralListeningPort" value="8081" />
  <add key="TCPAudioListeningPort" value="17860" />
  <add key="UDPAudioDataPort" value="17861" />
  <add key="ReflectorTcpPortStart" value="20000" />
  <add key="TCPCentralListeningIP" value="0.0.0.0" />
  <add key="TCPAdministrationProfilesPort" value="8082" />
  <add key="LoginMessage" value="Welcome to the Central Server." />
  <add key="RegistrationMessage" value="Thank you for registering." />
</appSettings>
</configuration>
```

3.2.6 Configure Listening Ports

The iConf Central Server Service uses the following port configuration

1 configurable **TCP** port to listen to iConf Servers that need to REGISTER, LOGIN, LOGOUT, ADD/REMOVE contacts ...

TCPCentralListeningPort

1 configurable **TCP** port for audio communication

TCPAudioListeningPort

1 configurable **UDP** port for audio communication

UDPAudioDataPort

Starting port number for dynamic **TCP** ports for Video communication

ReflectorTcpPortStart

The port that should be used to Transfer files

FileTransferPort

The directory that should be used (on the central server side) to store the shared files

FileTransferDirectory

Starting port number for dynamic **TCP** ports for Screen Sharing

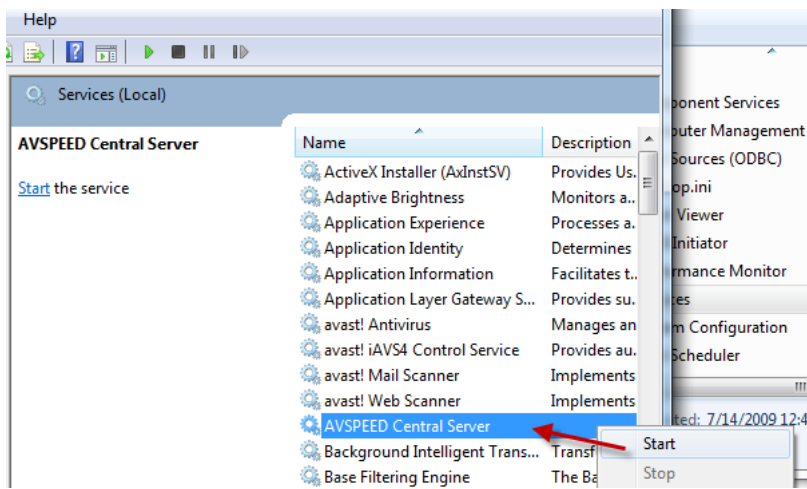
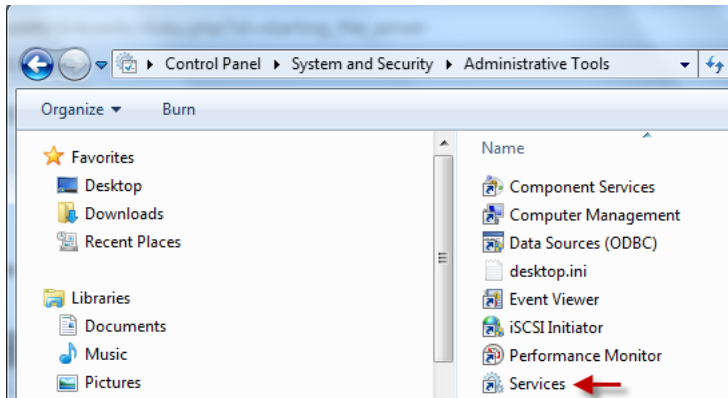
ScreenSharingTcpPortStart

The listening ports can be configured in the iConf Central Server Service configuration file

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
</configSections>
<connectionStrings>
<add      name="SQLServerConnectionString"      connectionString="Data      Source=AVSPEED\AVSPEED;Initial
Catalog=usersDB;Integrated Security=True" />
<add      name="EmbeddedConnectionString"      connectionString="Data      Source=|DataDirectory|\users.vdb3"
providerName="VistaDB.NET20" />
<add name="MySQLConnectionString" connectionString="server=localhost;user id=test;Password=test;persist security
info=True;database=testdb" providerName="MySql.Data.MySqlClient" />
</connectionStrings>
<appSettings>
<add key="DBType" value="EMBEDDED" />
<add key="EnableLog" value="False" />
<add key="TCPCentralListeningPort" value="8081" />
<add key="TCPAudioListeningPort" value="17860" />
<add key="UDPAudioDataPort" value="17861" />
<add key="ReflectorTcpPortStart" value="20000" />
<add key="ScreenSharingTcpPortStart" value="10000" />
<add key="TCPCentralListeningIP" value="0.0.0.0" />
<add key="TCPAdministrationProfilesPort" value="8082" />
<add key="LoginMessage" value="Welcome to the Central Server." />
<add key="RegistrationMessage" value="Thank you for registering." />
</appSettings>
</configuration>
```

3.2.7 Starting the Central Server

After configuring your Database Options and Listening Ports, you can start the iConf Central Server Service through the Windows Services Control Panel



3.2.8 Registering a new user

To Register a new user with the iConf Central Server Service, use the `iConfServerDotNet.RegisterUser` Method (see page 65)

Example

```
private void btnRegister_Click(object sender, EventArgs e)
{
    ics.RegisterUser("192.168.2.2", 8081, "user1", "password", "no extra data");
}

/// <summary>
/// delegate for registration response event handler
/// </summary>
/// <param name="registrationSuccess"></param>
/// <param name="messageFromServer"></param>
public delegate void ProcessRegistrationResponseDelegate( bool registrationSuccess,
string messageFromServer );

private void ProcessRegistrationResponse( bool registrationSuccess, string
messageFromServer )
{
    if (registrationSuccess)
    {

```

```

        txtUserName = regForm.txtUserName;
        txtPassword = regForm.txtPassword;

        MessageBox.Show("User successfully registered, please login", "Success",
            MessageBoxButtons.OK,
                MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Unable to register user : " + messageFromServer, "Error",
            MessageBoxButtons.OK,
                MessageBoxIcon.Error);
    }
}

/// <summary>
/// Registration response event handler
/// </summary>
/// <param name="sender"></param>
/// <param name="registrationSuccess"></param>
/// <param name="messageFromServer"></param>
private void ics_RegistrationResponse(object sender, bool registrationSuccess,
string messageFromServer)
{
    Invoke(new ProcessRegistrationResponseDelegate(ProcessRegistrationResponse),
new object[] { registrationSuccess, messageFromServer });
}

```

3.2.9 Logging in

To log into the Central Server, use the iConfServerDotNet.Login Method (see page 63)

Example

```

private void btnLogin_Click(object sender, System.EventArgs e)
{
    try
    {
        ics.Login(txtCentralIp.Text, Convert.ToInt32(txtCentralPort.Text),
txtUserName.Text, txtPassword.Text);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unable to login - make sure that the Central Server
Service is running and check the Port that it is listening on :\n" + ex.Message);
    }
}

/// <summary>
/// delegate for login reponse event handler
/// </summary>
public delegate void ProcessLoginResponseDelegate();

private void ProcessLoginResponse()
{
    this.lblLoginStatus.Text = "Logged in..";

    this.pbLoginStatus.Image =
CentralServerDemo.Properties.Resources.administrator_ok_16;

    this.ics.GetPendingContactRequests();
}

```

```

/// <summary>
/// Login response event handler
/// </summary>
/// <param name="sender"></param>
/// <param name="loginSuccess"></param>
/// <param name="messageFromServer"></param>
private void ics_LoginResponse(object sender, System.Boolean loginSuccess,
System.String messageFromServer)
{
    if (loginSuccess)
    {
        Invoke(new ProcessLoginResponseDelegate(ProcessLoginResponse), null);

        //get the list of contacts
        ics.GetContactList();

        ics.GetPendingContactRequests();
    }
    else
    {
        MessageBox.Show(messageFromServer, "Unable to login", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

```

3.2.10 Logging out

To Log out of the Central Server, use the iConfServerDotNet.Logout Method (see page 64)

Example

```

private void btnLogout_Click(object sender, System.EventArgs e)
{
    //leave any current conference
    ics.LeaveConference();

    //logout
    ics.Logout();
}

public delegate void ProcessLogoutResponseDelegate();

private void ProcessLogoutResponse()
{
    btnLogin.Enabled = true;

    btnLogout.Enabled = false;
}

private void ics_LogoutResponse(object sender, System.Boolean logoutSuccess,
System.String messageFromServer)
{
    if (logoutSuccess)
    {
        Invoke(new ProcessLogoutResponseDelegate(ProcessLogoutResponse), null);
    }
}

```

3.2.11 Adding a new Contact

To add a new contact, use the `iConfServerDotNet.AddUserAsContact` Method (see page 49)

New contact requests need to be accepted before being finalized.

Example

```
private void lblAdd_Click(object sender, EventArgs e)
{
    ics.AddUserAsContact(contactForm.txtContactToAdd.Text);
}

public delegate void ProcessAddContactResponseDelegate(bool addContactSuccess,
string messageFromServer);

private void ProcessAddContactResponse(bool addContactSuccess, string
messageFromServer)
{
    if (addContactSuccess)
    {
        //get contact list - note that contact added will show as offline until
        //contact request is accepted
        ics.GetContactList();
    }
    else
    {
        MessageBox.Show(messageFromServer, "Unable to add contact",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void ics_AddContactRequestResponse(object sender, bool addContactSuccess,
string messageFromServer)
{
    Invoke(new ProcessAddContactResponseDelegate(ProcessAddContactResponse), new
object[] { addContactSuccess, messageFromServer });
}

[.....THE USER ADDED AS A CONTACT ... GETS A NEW CONTACT REQUEST.... ]

public delegate void NewContactRequestDelegate( string fromUser );

private void NewContactRequest(string fromUser)
{
    if (MessageBox.Show("New contact request from " + fromUser, "Contact Request",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        ics.AcceptContactRequest(fromUser);
    }
}

private void ics_NewContactRequest(object sender, string fromuser)
{
    Invoke(new NewContactRequestDelegate(NewContactRequest), new object[] {
fromuser });
}
```

3.2.12 Accepting/Rejecting a contact request

When receiving a new contact request, the `iConfServerDotNet.NewContactRequest` Event (see page 103) is fired.

To accept the contact request, use the `iConfServerDotNet.AcceptContactRequest` Method (see page 49)

To reject the contact request, use the `iConfServerDotNet.RemoveContact` Method (see page 67)

Example

Accepting a new contact request

```
public delegate void NewContactRequestDelegate( string fromUser );

private void NewContactRequest(string fromUser)
{
    if (MessageBox.Show("Accept new contact request from " + fromUser, "Contact
Request", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        //accept the contact request

        ics.AcceptContactRequest(fromUser);
    }
    else
    {
        //reject the contact request

        ics.RemoveContact(fromUser);
    }
}

private void ics_NewContactRequest(object sender, string fromuser)
{
    Invoke(new NewContactRequestDelegate(NewContactRequest), new object[] {
fromuser });
}
```

3.2.13 Viewing Pending Contact Request

When a user is offline and is sent a contact request, these requests get queued up and can be retrieved using the `iConfServerDotNet.GetPendingContactRequests` Method (see page 56)

Example

```
private void btnLogin_Click(object sender, System.EventArgs e)
{
    try
    {
        ics.Login(txtCentralIp.Text, Convert.ToInt32(txtCentralPort.Text),
txtUserName.Text, txtPassword.Text);
    }
    catch(Exception ex)
    {
        MessageBox.Show("Unable to login - make sure that the Central Server
Service is running and check the Port that it is listening on :\n" + ex.Message);
    }
}
```

```

    }
    /// <summary>
    /// delegate for login reponse event handler
    /// </summary>
    public delegate void ProcessLoginResponseDelegate();

    private void ProcessLoginResponse()
    {
        this.lblLoginStatus.Text = "Logged in..";

        this.pbLoginStatus.Image =
CentralServerDemo.Properties.Resources.administrator_ok_16;

        this.ics.GetPendingContactRequests();
    }

    /// <summary>
    /// Login response event handler
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="loginSuccess"></param>
    /// <param name="messageFromServer"></param>
    private void ics_LoginResponse(object sender, System.Boolean loginSuccess,
System.String messageFromServer)
    {
        if (loginSuccess)
        {
            Invoke(new ProcessLoginResponseDelegate(ProcessLoginResponse), null);

            //get the list of contacts
            ics.GetContactList();

            ics.GetPendingContactRequests();
        }
        else
        {
            MessageBox.Show(messageFromServer, "Unable to login", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    public delegate void ProcessPendingContactDelegate(Hashtable requests);

    private void ProcessPendingContact(Hashtable requests)
    {
        if (requests.Count > 0)
        {
            //add pending contact requests to a list

            this.pendingForm.lstPending.Items.Clear();

            foreach (DictionaryEntry de in requests)
            {
                pendingForm.lstPending.Items.Add(de.Value.ToString());
            }
            pendingForm.Show();
        }
    }

    private void ics_PendingContactRequest(object sender, Hashtable contactsHash,
string contacts)
    {
        Invoke(new ProcessPendingContactDelegate(ProcessPendingContact), new object[] {
contactsHash });
    }

```



```
}
```

3.2.14 Starting a conference with 1 or more users

To start a conference with 1 or more user, use the `iConfServerDotNet.StartConference` Method (see page 78)

Assuming that you have a listview (`lvContact`) that contains a list of contacts

```
private void btnStartConference_Click(object sender, System.EventArgs e)
{
    if (lvContact.SelectedItems.Count != 0)
    {
        ArrayList usersToInvite = new ArrayList();

        foreach (ListViewItem item in lvContact.SelectedItems)
        {
            usersToInvite.Add(item.Text);
        }

        ics.StartConference(usersToInvite);
    }
}
```

The users that you send the conference request to (using the `iConfServerDotNet.StartConference` Method (see page 78)) receive a conference request through the `iConfServerDotNet.ConferenceRequest` Event (see page 100)

```
public delegate void ProcessConferenceRequestDelegate(String leader, ArrayList
otherUsers, string otherUsersStr, string conferenceId);

private void ProcessConferenceRequest(String leader, ArrayList otherUsers, string
otherUsersStr, string conferenceId)
{
    if (MessageBox.Show("Would you like to accept a conference request from " +
leader + "?", "Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
    {
        ics.JoinConference(conferenceId);
    }
}

private void ics_ConferenceRequest(object sender, System.String leader,
System.Collections.ArrayList otherUsers, System.String otherUsersStr, System.String
conferenceId)
{
    Invoke(new ProcessConferenceRequestDelegate(ProcessConferenceRequest), new
object[] { leader, otherUsers, otherUsersStr, conferenceId });
}
```

3.2.15 Accepting a conference request

To accept a conference request, use the `iConfServerDotNet.JoinConference` Method (see page 61)

the `iConfServerDotNet.JoinConference` Method (see page 61) takes in the `conferenceId` parameter which represents the conference that the user was invited to join.

The users that you send the conference request to (using the `iConfServerDotNet.StartConference` Method (see page 78)) receive a conference request through the `iConfServerDotNet.ConferenceRequest` Event (see page 100)

```
public delegate void ProcessConferenceRequestDelegate(String leader, ArrayList
otherUsers, string otherUsersStr, string conferenceId);

private void ProcessConferenceRequest(String leader, ArrayList otherUsers, string
otherUsersStr, string conferenceId)
{
    if (MessageBox.Show("Would you like to accept a conference request from " +
leader + "?", "Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
    {
        ics.JoinConference(conferenceId);
    }
}

private void ics_ConferenceRequest(object sender, System.String leader,
System.Collections.ArrayList otherUsers, System.String otherUsersStr, System.String
conferenceId)
{
    Invoke(new ProcessConferenceRequestDelegate(ProcessConferenceRequest), new
object[] { leader, otherUsers, otherUsersStr, conferenceId });
}
```

3.2.16 Viewing a user (video stream)

After joining a conference, you can initiate a request to view a user's video stream by calling the `iConfServerDotNet.ViewUser` Method (see page 86)

```
private void lvInConferenceWith_DoubleClick(object sender, System.EventArgs e)
{
    //view selected user
    string selectedUser = null;

    if (lvInConferenceWith.SelectedItems.Count <= 0)
    {
        return;
    }

    selectedUser = lvInConferenceWith.SelectedItems[0].Text;

    ics.ViewUser(selectedUser);
}
```

When the data needed to view the user is received, the `iConfServerDotNet.ViewUserResponseReceived` Event (see page 106) is fired.

You can then use the `iConfClientDotNet.Call` Method (see page 119) to place a call to the user and receive its video stream.

```
public delegate void ViewUserResponseDelegate(System.String userName, System.Int32
port, System.Int32 videoWidth, System.Int32 videoHeight, System.String videoCodec);

private void ViewUserResponse(System.String userName, System.Int32 port,
System.Int32 videoWidth, System.Int32 videoHeight, System.String videoCodec)
{
    //check to see if we are already viewing that user
    //if so bring form in front
    for (int i = Application.OpenForms.Count - 1; i >= 0; i--)
```

```

    {
        if (Application.OpenForms[i].Tag != null)
        {
            if (Application.OpenForms[i].Tag.ToString() == userName)
            {
                Application.OpenForms[i].BringToFront();
                return;
            }
        }
    }

    ClientForm client = null;

    client = new ClientForm();

    client.Tag = userName;

    client.Text = "Viewing > " + userName;

    client.Show();

    client.icc.Call(txtCentralIp.Text, port, videoWidth, videoHeight, "n/a", "",
        "", 0, 0, 0, videoCodec);
}

```

3.2.17 Updating your Profile picture

Each user registered with the Central Server can take a picture from the webcam that will be associated with the user account.

To take the picture from the webcam and attach it to the logged in user account, call the `iConfServerDotNet.UpdateProfilePicture` Method ([see page 86](#))

Once the picture is successfully sent and stored on the server, the `iConfServerDotNet.ProfilePictureUpdated` Event ([see page 103](#)) will be fired.

To view the newly updated picture, use the `iConfServerDotNet.GetProfilePicture` Method ([see page 56](#))

3.2.18 Sharing Files

The Central Server Service allows users to share files.

Central Server Service configuration

To enable sharing files through the Central Server,
setup the `FileTransferPort` and `FileTransferDirectory` config keys in the Central Server Service configuration file.

The following configuration uses port 17865 to transfer files, and files will be stored in C:\TEMP where the Central Server Service is running from.

```
<add key="FileTransferPort" value="17865" />
<add key="FileTransferDirectory" value="C:\TEMP" />
```

Sharing a file with a user

To Share a file with a user, call the `ShareFile` (see page 78) function

```
string fileId = icserver.ShareFile(@"C:\filetosshare.txt");
icserver.SendShareFileRequest(fileId, "user2");
```

The user 'user2' will then receive a

```
public delegate void ProcessFileShareRequestDelegate(
    string fileId, string fromuser, string filename, long fileSize);

private void ProcessFileShareRequest(string fileId, string fromuser, string
filename, long fileSize)
{
    string message = string.Empty;

    message = fromuser + " is sharing a file with you\n\n";
    message += " Filename : " + filename + "\n";
    message += " Size : " + fileSize + "\n\n";
    message += "Would you like to accept ?\n";

    if (MessageBox.Show(message, @"File sharing request", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
    {
        ics.GetFile(fileId, filename, @"C:\receivedfile.txt");
    }
}

private void ics_NewFileShareRequest(string fileId, string fromuser, string
filename, long fileSize)
{
    Invoke(new ProcessFileShareRequestDelegate(ProcessFileShareRequest), new
object[] { fileId, fromuser, filename, fileSize });
}
```

When the file download completes, the `iConfServerDotNet.FileDownloadCompleted` Event (see page 101) is fired.


To monitor the progress of a file download, use the `iConfServerDotNet.FileDownloadProgress` Event (see page 101)

4 Symbol Reference


4.1 iConfServer.NET Namespace

iConfServer.NET Requirements virtualw.dll avbox4.dll

Classes

	Name	Description
	iConfServerDotNet (🔗 see page 25)	Implementing interface that will be visible from JS


Interfaces

	Name	Description
	ICallback (🔗 see page 106)	This is class iConfServer.NET.ICallback.

4.1.1 Classes

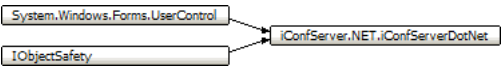
The following table lists classes in this documentation.

Classes

	Name	Description
	iConfServerDotNet (🔗 see page 25)	Implementing interface that will be visible from JS

4.1.1.1 iConfServerDotNet Class

Inheritance Hierarchy



C#

```
public class iConfServerDotNet : System.Windows.Forms.UserControl, IObjectSafety;
```

File


iConfServerDotNet.cs

Remarks


Implementing interface that will be visible from JS

Members







Methods

	Name	Description
	iConfServerDotNet (🔗 see page 45)	Initializes a new instance of the iConfServerDotNet class







iConfServerDotNet Structures

	Name	Description
	COPYDATASTRUCT (see page 46)	Used for WM_COPYDATA (see page 48) for string messages









iConfServerDotNet Enumerations

	Name	Description
	AudioPlayBackType (see page 46)	--deprecated 0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.
	audioType (see page 46)	enum representing the type of system that should be used for audio Windows Audio System or DirectSound
	DesktopSharingType (see page 46)	This is record iConfServer.NET.iConfServerDotNet.DesktopSharingType.
	H264Profile (see page 47)	This is record iConfServer.NET.iConfServerDotNet.H264Profile.
	h264SpeedLevel (see page 47)	The speed level of the H.264 encoder
	RateControl (see page 47)	This is record iConfServer.NET.iConfServerDotNet.RateControl.

iConfServerDotNet Fields

	Name	Description
	H263 (see page 48)	Constant representing the H263 codec selection. Can be used as a replacement for typing the string "H263"
	H264 (see page 48)	Constant representing the H264 codec selection. Can be used as a replacement for typing the string "H264"
	MPEG4 (see page 48)	Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"
	mycallbackid (see page 48)	This is mycallbackid, a member of class iConfServerDotNet.
	WM_COPYDATA (see page 48)	This is WM_COPYDATA, a member of class iConfServerDotNet.
	WM_USER (see page 48)	This is WM_USER, a member of class iConfServerDotNet.

iConfServerDotNet Methods













	Name	Description
	AcceptCall (see page 48)	Function used to accept a call from an iConf Client component
	AcceptContactRequest (see page 49)	Function used to accept a contact request with a specific user
	AddUserAsContact (see page 49)	Used to send a request to a user to be added as a contact
	AutoJoinRoomOnLogin (see page 49)	Used to automatically Join a room upon logging in to the Central Server
	CaptureFrame (see page 49)	Used to capture a video frame to a file or clipboard
	ChangePassword (see page 50)	Function used to change the password used to log into the central server
	ChangeStatus (see page 50)	Used to change the status of the user logged into the Central Server
	CRLFDelimitedStringToIList (see page 50)	This is CRLFDelimitedStringToIList, a member of class iConfServerDotNet.

✦	DeleteProfilePicture (see page 51)	Used to delete a the current logged in user's profile picture from the central server service.
✦	DeleteUser (see page 51)	Deletes a user from the central server
✦	DenyScreenControl (see page 51)	This is DenyScreenControl, a member of class iConfServerDotNet.
✦	DisconnectClient (see page 51)	Use this function to force a connected client to disconnect
✦	DisconnectFromCentralServer (see page 52)	This is DisconnectFromCentralServer, a member of class iConfServerDotNet.
✦	Dispose (see page 52)	Clean up any resources being used.
✦	EnableEchoCancellation (see page 52)	Enable/Disable echo cancellation
✦	EnableGrayScale (see page 52)	Enables/Disables grayscale video stream
✦	EnableVoiceActivation (see page 52)	Enables/disables voice activation
✦	GetAssemblyVersion (see page 53)	returns the version of the assembly
✦	GetAudioBytesReceived (see page 53)	Gets the number of audio bytes received
✦	GetAudioCompressorsForRecording (see page 53)	This function returns the list of installed Audio Compressors that can be used for recording
✦	GetAudioCompressorsForRecordingStr (see page 53)	Used to Get a character delimited string of installed Audio Compressors that can be used for recording. use only when using component as COM.
✦	GetAudioInputDevices (see page 53)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method
✦	GetAudioInputDevicesList (see page 54)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method
✦	GetAudioInputLevel (see page 54)	deprecated Gets the audio input (microphone) level range is 0 - 65535
✦	GetAudioInputLevel (see page 54)	Used to get the current audio input level - useful to display a vu meter in an application
✦	GetAudioOutputDevices (see page 54)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method
✦	GetAudioOutputDevicesList (see page 54)	Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method
✦	GetAudioOutputVolume (see page 54)	deprecated Gets the audio output volume level
✦	GetBoundIP (see page 55)	Use this function to get the IP address to which the server is bound
✦	GetBoundPort (see page 55)	Use this function to get the Port to which the iConf Server is bound
✦	GetConferenceRooms (see page 55)	Used to get list of existing conference rooms
✦	GetContactList (see page 55)	Function used to get the list of contacts for the current logged in user
✦	GetFile (see page 55)	This is GetFile, a member of class iConfServerDotNet.
✦	GetLocalIp (see page 55)	Helper function used to get the list of local IP Addresses
✦	GetLocalIpStr (see page 56)	Helper function used to get the list of local IP Addresses
✦	GetMyCallBackId (see page 56)	Used to get the CallBackId of the iConfServer when in IP to IP mode
✦	GetPendingContactRequests (see page 56)	Function used to gets the pending contact requests for the current logged in user












✦	GetProfilePicture (see page 56)	Used to get the profile picture of a iConf Central Server Service user Requires Central Server to have specified Admin profile port
✦	GetStatusMessages (see page 56)	Used to get a list of status messages for the current logged in user's contacts
✦	GetUserProfile (see page 56)	Function used to get the profile of a user
✦	GetUserScr (see page 57)	This is GetUserScr, a member of class iConfServerDotNet.
✦	GetVideoCompressorsForRecording (see page 57)	This function returns the list of installed Video Compressors that can be used for recording
✦	GetVideoCompressorsForRecordingStr (see page 57)	Used to Get a character delimited string of installed Video Compressors that can be used for recording. use only when using component as COM.
✦	GetVideoDevices (see page 57)	Use this method to build a list of Video Capture devices found on the system
✦	GetVideoDevicesStr (see page 58)	Used to Get a character delimited string of Video Devices. use only when using component as COM.
✦	GetVideoInputs (see page 58)	Gets a List of video inputs available for the selected video device
✦	GetVideoInputsStr (see page 58)	Gets a Comma Separated string of video inputs available for the selected video device (COM use only)
✦	GetVideoSizes (see page 58)	Gets the List of available video preview sizes This method should only be called after a call to SelectVideoDevice (see page 69) because each video device has different video available video sizes
✦	GetVideoSizesStr (see page 59)	Gets the list of available video preview sizes in the form of string separated by a specified delimiter
✦	GetVolume (see page 59)	Used to get the Volume of an Audio conference
✦	GrantScreenControl (see page 59)	This is GrantScreenControl, a member of class iConfServerDotNet.
✦	HoldRecording (see page 59)	hold the recording after starting the recording process
✦	InitializeAudioSystem (see page 60)	Use this function to initialize the audio system Initializing the audio system is required - When not called explicitly, the audio system will be initialized with some default parameters
✦	InviteClientToViewScreen (see page 60)	In IP to IP mode, this function sends an invitation to a selected or all connected clients using their socket handle
✦	InviteUsersToViewScreen (see page 60)	When logged into the Central Server, this function allows to send an invitation to attend a screen sharing session to a number of users
✦	IsConnected (see page 61)	checks connection to central server
✦	JoinConference (see page 61)	Function used to Join a conference using its 'conference id'
✦	JoinConferenceRoom (see page 62)	Used to join/create a conference room. If you try to join a conference room which does not exist it will be created as a new conference room. After the last user leaves the conference room, the room gets deleted from the server
✦	LeaveConference (see page 62)	Function used to leave the conference that the logged in user is currently participating in
✦	LeaveCurrentConferenceRoom (see page 63)	Leaves current conference room
✦	Listen (see page 63)	starts listening for incoming connections on the specified ip and port
✦	Logging (see page 63)	enables / disables logging log file is in temp path iconflog.txt
✦	Login (see page 63)	Function used to Log (see page 102) into the Central Server
✦	Logout (see page 64)	Function used to log out of the Central Server

◆	LoopBackAudioTest (see page 64)	This is LoopBackAudioTest, a member of class iConfServerDotNet.
◆	Mute (see page 64)	Mutes/Un-Mute incoming audio playback
◆	MuteContactInConference (see page 64)	used to /un-mute the audio of a contact while in a conference
◆	Pause (see page 64)	Pause/unpause video & audio streaming to connected clients
◆	ReconfigureAudioSystem (see page 65)	Use this function to reconfigure the audio system (choose different audio input/output device)
◆	RecordToNewFileNow (see page 65)	During recording, invoking RecordToNewFile closes the current recording file, and creates a new recording file.
◆	RegisterUser (see page 65)	Function used to Register a new user with a Central Server
◆	RejectCall (see page 66)	Function used to reject an incoming call from an iConf Client
◆	RejectCallEx (see page 66)	Function used to reject an incoming call from an iConf Client
◆	RejectConferenceRequest (see page 66)	use this function to reject a Conference Request when getting conference request
◆	RemoveContact (see page 67)	Used to remove a contact
◆	RemoveNoiseFromOutgoingAudio (see page 67)	Enables/Disables removing noise from outgoing audio
◆	RequestScreenControl (see page 67)	DEPRECATED Used to request control while in a screen sharing session (Central Server mode ONLY)
◆	ResumeRecording (see page 67)	Resumes the recording process previously heldby calling HoldRecording (see page 59)
◆	RevokeScreenControl (see page 67)	This is RevokeScreenControl, a member of class iConfServerDotNet.
◆	Search (see page 68)	This is Search, a member of class iConfServerDotNet.
◆	SelectIPCameraAsVideoSource (see page 68)	Used to select an IP Camera as video source instead of a video capture device
◆	SelectVideoCaptureDeviceAsVideoSource (see page 69)	Used to select a video file as video source instead of a video capture device
◆	SelectVideoDevice (see page 69)	Use this function to select a video capture device using its 0 based index
◆	SelectVideoFileAsVideoSource (see page 69)	USED to select a file as video source instead of a video capture device
◆	SelectVideoInput (see page 69)	Used to select a video Input
◆	SendCustomObject (see page 70)	Used to send a custom string to a list of users logged into the Central Server Can be used to send any type of object to a list of users logged into the Central Server. Example serialize object as string => SendCustomObject => CustomObjectReceived (see page 101) => Deserialize back to object
◆	SendKeepAlive (see page 70)	Sends a KEEP ALIVE message to the Central Server
◆ S	SendMessage (see page 70)	For use with WM_COPYDATA (see page 48) and COPYDATASTRUCT (see page 46)
◆	SendMessageToConference (see page 70)	Sends text message to Conference - Message 500 chars max!
◆	SendPrivatemessageToUser (see page 70)	When logged into the Central server, use this function to send a private message to a user
◆	SendShareFileRequest (see page 71)	Use this function to share a file with a user
◆	SendTextMessage (see page 71)	Use this function to send a Text message to all or a specific connected iConfClient.NET (see page 111)
◆	SetAudioInputlevel (see page 71)	deprecated Sets the audio input (microphone) level
◆	SetAudioOutputVolume (see page 71)	deprecated Sets the Audio Output Volume
◆	SetAudioPlaybackType (see page 72)	deprecated

◆	SetBrightness (see page 72)	Sets the brightness of the video stream 0 to reset
◆	SetContrast (see page 72)	Sets the contrast of the video stream. Use 0 to reset
◆	SetEncoderProperties (see page 73)	Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..
◆	SetEncoderPropertiesEx (see page 74)	Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..
◆	SetFrameCaptureInterval (see page 75)	Sets the number of frames to skip (no encoding and sending) between each captured frame.
◆	SetGainLevel (see page 75)	This is SetGainLevel, a member of class iConfServerDotNet.
◆	SetHue (see page 76)	Sets the hue of the video stream. Use 0 to reset
◆	SetIEMode (see page 76)	Must be set when using component in Internet Explorer
◆	SetOutputGain (see page 76)	This is SetOutputGain, a member of class iConfServerDotNet.
◆	SetPreviewFrameRate (see page 76)	Desired frame rate, expressed in frame per second
◆	SetSaturation (see page 76)	Sets the Saturation of the video stream 0 to reset
◆	SetStatusMessage (see page 77)	Sets the status message of the user currently logged into the Central Server
◆	SetUnlockCode (see page 77)	Used with the registered version of the iConf .NET SDK This function should be called before calling the StartPreview (see page 79) function
◆	SetVoiceActivationLevel (see page 77)	Sets the voice activation level
◆	SetVolume (see page 77)	Used to set the Volume of an Audio conference
◆	ShareFile (see page 78)	Use this function to send a file to the participants of a conference
◆	ShowVideoCaptureDevicePropertyDialog (see page 78)	Used to display the video capture device specific property dialog where sometimes settings like face tracking, brightness, backlight, hue, saturation and other device settings can be changed.
◆	StartConference (see page 78)	Used to start a conference with a selected number of users
◆	StartConferenceEx (see page 79)	Used to start a conference with a selected number of users
◆	StartMotionDetection (see page 79)	Enables motion detection in the video preview using teh specified sensitivity
◆	StartPreview (see page 79)	Use this function to start previewing video using the selected video capture device (see the M:iConfServer.NET.iConfServerDotNet.SelectVideoDevice(System.Int32)SelectVideoDevice (see page 69)function)
◆	StartRecording (see page 80)	Function used to record the activity of a window along with Audio. Can be used to record a conferencing session
◆	StartRecordingAudioConference (see page 80)	Records the audio from a conference to a wav file
◆	StartRecordingCurrentVideoPreview (see page 80)	Record the current video preview (with or without audio) to a wmv file
◆	StartRecordingCurrentVideoPreviewWmv9 (see page 82)	Record the current video preview (with or without audio) to a wmv 9 encoded file using a custom profile
◆	StartRecordingToWMV (see page 83)	Function used to record the activity of a window along with Audio. Can be used to record a conferencing session
◆	StartScreenSharing (see page 83)	This is StartScreenSharing, a member of class iConfServerDotNet.
◆	StartTransmittingAudio (see page 84)	Starts transmitting audio. Only call this function if you have explicitly called the StopTransmittingAudio (see page 85) function
◆	StopMotionDetection (see page 84)	Stops the motion detection

	StopPreview (see page 84)	Stops the video preview and ends the encoding of the video
	StopRecording (see page 84)	Used to stop a recording session started by calling the StartRecording (see page 80) function
	StopRecordingAudioConference (see page 85)	Stops the process of recording the audio conference (process started by calling StartRecordingAudioConference (see page 80))
	StopRecordingCurrentVideoPreview (see page 85)	Used to stop recording the current video preview function
	StopScreenSharing (see page 85)	Stops a screen sharing session
	StopTransmittingAudio (see page 85)	Stops transmitting audio
	TextOverLay (see page 85)	Enables/Disables Text Overlay
	UpdateProfilePicture (see page 86)	Asynchronous function used to Update the profile picture of the user currently logged into the Central Server.
	ViewUser (see page 86)	Send request to central server to view the video feed of a user
	waveOutGetVolume (see page 86)	This is waveOutGetVolume, a member of class iConfServerDotNet.
	waveOutSetVolume (see page 86)	This is waveOutSetVolume, a member of class iConfServerDotNet.
	XmitDesktop (see page 86)	This is XmitDesktop, a member of class iConfServerDotNet.

iConfServerDotNet Properties

	Name	Description
	CallbackId (see page 87)	the Callback Id of the iConf Server. What is a Callback Id ? A callback id is an identifier that gets used in the case of an incoming call, to determine if the incoming call can be automatically accepted.m
	CallbackObject (see page 87)	This is CallbackObject, a member of class iConfServerDotNet.
	ConnectedClientCount (see page 87)	returns the number of clients connected to the iConfServer.NET (see page 25) component in IP to IP mode
	CurrentVideoCodec (see page 87)	The selected video codec H263 (see page 48) / H264 (see page 48) /MPEG4 (see page 48)
	IsConnectedToCentralServer (see page 87)	Checks to see if there is an active connection to the Central Server.
	IsListening (see page 87)	This is IsListening, a member of class iConfServerDotNet.
	IsScreenSharingEnabled (see page 87)	Returns true if screen sharing is turned on, false otherwise
	PreviewHeight (see page 88)	returns teh height of the video preview
	PreviewWidth (see page 88)	returns the width of the video preview
	VideoDeviceCount (see page 88)	returns the number of video devices installed on the system
	VideoRateControl (see page 88)	This is VideoRateControl, a member of class iConfServerDotNet.












iConfServerDotNet Delegates


























Name	Description
AddContactRequestResponseDelegate (see page 88)	Delegate for AddContactRequestResponse (see page 100) event
CannotRunGraphDelegate (see page 88)	Delegate for the CannotRunGraph (see page 100) event
ChangePasswordResponseDelegate (see page 88)	Delegate for ChangePasswordResponse event
ClientConnectionLostDelegate (see page 89)	Delegate for ClieConnectionLost event

ClientDisconnectedDelegate (see page 89)	Delegate for ClientDisconnected (see page 100) event
ConferenceMessageReceivedDelegate (see page 89)	Delegate for ConferenceMessageReceived (see page 100) event
ConferencePrivateMessageReceivedDelegate (see page 89)	Delegate for ConferencePrivateMessageReceived (see page 100) event
ConferenceRequestDelegate (see page 90)	Delegate for ConferenceRequest (see page 100) event
ConferenceRequestRejectedDelegate (see page 90)	Delegate for ConferenceRequestRejected (see page 101) event
ContactListReceivedDelegate (see page 90)	Delegate for ContactListReceived (see page 101) Event
ContactStatusChangeDelegate (see page 91)	Delegate for ContactStatusChange event
ContactStatusMessageChangeDelegate (see page 91)	Delegate for ContactStatusMessageChanged (see page 101) event
CustomObjectReceivedDelegate (see page 91)	Delegate for CustomObjectReceived (see page 101) event
DisconnectedFromCentralServerDelegate (see page 91)	Delegate for DisconnectedFromCentralServer (see page 101) event
FileDownloadCompletedDelegate (see page 91)	Delegate for the FileDownloadCompleted (see page 101) event
FileDownloadProgressDelegate (see page 92)	Delegate for the FileDownloadProgress (see page 101) event
IncomingCallDelegate (see page 92)	Delegate for IncomingCall (see page 102) event
InvalidConferenceRoomPasswordDelegate (see page 93)	Delegate for InvalidConferenceRoomPassword (see page 102) event
JoinedConferenceRoomDelegate (see page 93)	Delegate for JoinedConferenceRoom (see page 102) event
LogDelegate (see page 93)	This is nested type iConfServer.NET.iConfServerDotNet.LogDelegate.
LoginResponseDelegate (see page 93)	Delegate for LoginResponse (see page 102) Event
LogoutResponseDelegate (see page 93)	Delegate for LogoutResponse (see page 102) event
MotionDetectedDelegate (see page 94)	Delegate for MotionDetected (see page 102) event
MotionNotDetectedDelegate (see page 94)	
NewContactRequestDelegate (see page 94)	Delegate for NewContactRequest (see page 103) event
NewDesktopAvailableDelegate (see page 94)	This is nested type iConfServer.NET.iConfServerDotNet.NewDesktopAvailableD elegate.
NewFileShareRequestDelegate (see page 94)	Delegate for the NewFileShareRequest (see page 103) event
NewVideoFrameAvailableDelegate (see page 94)	Delegate for the NewVideoFrameAvailable (see page 103) event
NewVideoFrameBytesAvailableDelegate (see page 95)	Delegate for the NewVideoFrameBytesAvailable (see page 103) event
PendingContactRequestResponseDelegate (see page 95)	Delegate for PendingContactRequestResponse event
ProfilePictureUpdatedDelegate (see page 95)	Delegate for ProfilePictureUpdated (see page 103) event
ReadyToShareScreenDelegate (see page 95)	Delegate for ReadyToShareScreen (see page 103) event
RecordingReadyToStartDelegate (see page 95)	This is nested type iConfServer.NET.iConfServerDotNet.RecordingReadyToStart Delegate.
RegistrationResponseDelegate (see page 96)	Delegate for RegistrationResponse (see page 104) Event
RemoveContactResponseDelegate (see page 96)	Delegate for RemoveContactResponse (see page 104) event
ScreenControlDeniedDelegate (see page 96)	Delegate for ScreenControlDenied (see page 104) event
ScreenControlGrantedDelegate (see page 96)	Delegate for ScreenControlGranted (see page 104) event
ScreenControlRequestDelegate (see page 96)	Delegate for ScreenControlRequest (see page 104) event

ScreenSharingSessionAcceptedDelegate (see page 97)	Delegate for ScreenSharingSessionAccepted (see page 104) event
ScreenSharingSessionAvailableDelegate (see page 97)	Delegate for the ScreenSharingSessionAvailable (see page 104) event
ScreenSharingSessionRejectedDelegate (see page 97)	Delegate for ScreenSharingSessionRejected (see page 105) event
SearchResponseDelegate (see page 97)	This is nested type iConfServer.NET.iConfServerDotNet.SearchResponseDelegate.
StatusMessagesReceivedDelegate (see page 98)	This is nested type iConfServer.NET.iConfServerDotNet.StatusMessagesReceivedDelegate.
TalkingDelegate (see page 98)	Delegate for Talking (see page 105) event
TextMessageReceivedDelegate (see page 98)	Delegate for the TextMessageReceived (see page 105) event
UserJoinedConferenceDelegate (see page 98)	Delegate for UserJoinedConference (see page 105) event
UserLeftConferenceDelegate (see page 98)	Delegate for UserLeftConference (see page 105) event
UserProfileReceivedDelegate (see page 99)	This is nested type iConfServer.NET.iConfServerDotNet.UserProfileReceivedDelegate.
VideoDeviceSelectedDelegate (see page 99)	Delegate for VideoDeviceSelected (see page 106) event
VideoPreviewStartedDelegate (see page 99)	Delegate for VideoPreviewStarted (see page 106) event
ViewUserResponseReceivedDelegate (see page 99)	Delegate for ViewUserResponse event


iConfServerDotNet Events

	Name	Description
	AddContactRequestResponse (see page 100)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.AddUserAsContact Method (see page 49)
	CannotRunGraph (see page 100)	Fired when the DirectX Video preview graph fails to run. Usually due to a failure to communicate with the video capture device (device might be in use)
	ChangePasswordResponse (see page 100)	fired when a response is received from the central server after a request to change password was made
	ClientConnectionLost (see page 100)	Fired when an iConf Client disconnects from the iConf Central Server Service use this event when you want to capture the handle of the client that has disconnected
	ClientDisconnected (see page 100)	Fired when an iConf Client disconnects from the iConf Central Server Service
	ConferenceMessageReceived (see page 100)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.SendMessageToConference Method (see page 70)
	ConferencePrivateMessageReceived (see page 100)	Fired when a private message is received from another user (see iConfServerDotNet.SendPrivatemessageToUser Method (see page 70))
	ConferenceRequest (see page 100)	Fired when a new conference request is received from a user
	ConferenceRequestRejected (see page 101)	Fired when a conference request has been rejected
	ContactListReceived (see page 101)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetContactList Method (see page 55)
	ContactStatusChanged (see page 101)	Fired when a user in the logged in user's contact list changes status

	ContactStatusMessageChanged (see page 101)	Fired when a user changes his/her status message
	CustomObjectReceived (see page 101)	Fired when a custom object is received from another user
	DisconnectedFromCentralServer (see page 101)	Fired when an iConf Server disconnects from the iConf Central Server Service
	FileDownloadCompleted (see page 101)	Fired when a file download completes successfully
	FileDownloadProgress (see page 101)	Use this event to report on the progress of a file download
	IncomingCall (see page 102)	Fired when a Call Request is received from an iConf Client
	InvalidConferenceRoomPassword (see page 102)	Fired when an attempt to join a password protected conference room fails due to an invalid password
	JoinedConferenceRoom (see page 102)	This is JoinedConferenceRoom, a member of class iConfServerDotNet.
	Log (see page 102)	This is Log, a member of class iConfServerDotNet.
	LoginResponse (see page 102)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.Login Method (see page 63)
	LogoutResponse (see page 102)	Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.LogoutResponse Method (see page 64)
	MotionDetected (see page 102)	Fired when motion is detected in the video preview
	MotionNotDetected (see page 102)	fired when motion is not detected in the video preview
	NewContactRequest (see page 103)	Fired when a new contact request is received from another user
	NewDesktopAvailable (see page 103)	This is NewDesktopAvailable, a member of class iConfServerDotNet.
	NewFileShareRequest (see page 103)	Fired when a file share request is received from a user (a user wants to share a file with you)
	NewVideoFrameAvailable (see page 103)	Fired when a new Video frame is available (Bitmap) for processing
	NewVideoFrameBytesAvailable (see page 103)	Fired when a new Video frame is available (byte[]) for processing
	PendingContactRequest (see page 103)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetPendingContactRequests Method (see page 56)
	ProfilePictureUpdated (see page 103)	Fired when the logged in user's profile picture gets successfully updated, following a call to the iConfServerDotNet.UpdateProfilePicture Method (see page 86)
	ReadyToShareScreen (see page 103)	Fired when a screen is ready to be shared following a call to StartScreenSharing (see page 83)
	RecordingReadyToStart (see page 104)	This is RecordingReadyToStart, a member of class iConfServerDotNet.
	RegistrationResponse (see page 104)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RegisterUser Method (see page 65)
	RemoveContactResponse (see page 104)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RemoveContact Method (see page 67)
	ScreenControlDenied (see page 104)	fired when a request to control the screen has been denied (central server mode)
	ScreenControlGranted (see page 104)	fired when a request to control the screen has been granted (central server mode)

	ScreenControlRequest (see page 104)	fired when a request to control the screen is received while in a screen sharing session
	ScreenSharingSessionAccepted (see page 104)	Fired when an iconfclient.net has accepted a screen sharing session request
	ScreenSharingSessionAvailable (see page 104)	Fired when a screen sharing session becomes available
	ScreenSharingSessionRejected (see page 105)	Fired when an iconfclient.net has rejected a screen sharing session request
	SearchResponse (see page 105)	This is SearchResponse, a member of class iConfServerDotNet.
	StatusMessagesReceived (see page 105)	This is StatusMessagesReceived, a member of class iConfServerDotNet.
	Talking (see page 105)	Fired when a user starts talking - This event will only work properly when Voice Activation Detection is enabled (see EnableVoiceActivation (see page 52) function).
	TextMessageReceived (see page 105)	Fired when a text message is received by from an iConf Client
	UserJoinedConference (see page 105)	Fired when a user joins a conference that the current logged in user is participating in.
	UserLeftConference (see page 105)	Fired when a user leaves a conference that the current logged in user is participating in.
	UserProfileReceived (see page 105)	This is UserProfileReceived, a member of class iConfServerDotNet.
	VideoDeviceSelected (see page 106)	Fired when a new video device has been selected after calling SelectVideoDevice (see page 69) - After this event is fired, new video sizes might be available and if you are displaying a list video sizes, you should reload that list
	VideoPreviewStarted (see page 106)	This is VideoPreviewStarted, a member of class iConfServerDotNet.
	ViewUserResponseReceived (see page 106)	Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.ViewUserResponseReceived Method (see page 86)

Methods

	Name	Description
	iConfServerDotNet (see page 45)	Initializes a new instance of the iConfServerDotNet class







iConfServerDotNet Delegates

Name	Description
AddContactRequestResponseDelegate (see page 88)	Delegate for AddContactRequestResponse (see page 100) event
CannotRunGraphDelegate (see page 88)	Delegate for the CannotRunGraph (see page 100) event
ChangePasswordResponseDelegate (see page 88)	Delegate for ChangePasswordResponse event
ClientConnectionLostDelegate (see page 89)	Delegate for ClientConnectionLost event
ClientDisconnectedDelegate (see page 89)	Delegate for ClientDisconnected (see page 100) event
ConferenceMessageReceivedDelegate (see page 89)	Delegate for ConferenceMessageReceived (see page 100) event
ConferencePrivateMessageReceivedDelegate (see page 89)	Delegate for ConferencePrivateMessageReceived (see page 100) event
ConferenceRequestDelegate (see page 90)	Delegate for ConferenceRequest (see page 100) event
ConferenceRequestRejectedDelegate (see page 90)	Delegate for ConferenceRequestRejected (see page 101) event
ContactListReceivedDelegate (see page 90)	Delegate for ContactListReceived (see page 101) Event







ContactStatusChangeDelegate (see page 91)	Delegate for ContactStatusChange event
ContactStatusMessageChangeDelegate (see page 91)	Delegate for ContactStatusMessageChanged (see page 101) event
CustomObjectReceivedDelegate (see page 91)	Delegate for CustomObjectReceived (see page 101) event
DisconnectedFromCentralServerDelegate (see page 91)	Delegate for DisconnectedFromCentralServer (see page 101) event
FileDownloadCompletedDelegate (see page 91)	Delegate for the FileDownloadCompleted (see page 101) event
FileDownloadProgressDelegate (see page 92)	Delegate for the FileDownloadProgress (see page 101) event
IncomingCallDelegate (see page 92)	Delegate for IncomingCall (see page 102) event
InvalidConferenceRoomPasswordDelegate (see page 93)	Delegate for InvalidConferenceRoomPassword (see page 102) event
JoinedConferenceRoomDelegate (see page 93)	Delegate for JoinedConferenceRoom (see page 102) event
LogDelegate (see page 93)	This is nested type iConfServer.NET.iConfServerDotNet.LogDelegate.
LoginResponseDelegate (see page 93)	Delegate for LoginResponse (see page 102) Event
LogoutResponseDelegate (see page 93)	Delegate for LogoutResponse (see page 102) event
MotionDetectedDelegate (see page 94)	Delegate for MotionDetected (see page 102) event
MotionNotDetectedDelegate (see page 94)	
NewContactRequestDelegate (see page 94)	Delegate for NewContactRequest (see page 103) event
NewDesktopAvailableDelegate (see page 94)	This is nested type iConfServer.NET.iConfServerDotNet.NewDesktopAvailableDelegate.
NewFileShareRequestDelegate (see page 94)	Delegate for the NewFileShareRequest (see page 103) event
NewVideoFrameAvailableDelegate (see page 94)	Delegate for the NewVideoFrameAvailable (see page 103) event
NewVideoFrameBytesAvailableDelegate (see page 95)	Delegate for the NewVideoFrameBytesAvailable (see page 103) event
PendingContactRequestResponseDelegate (see page 95)	Delegate for PendingContactRequestResponse event
ProfilePictureUpdatedDelegate (see page 95)	Delegate for ProfilePictureUpdated (see page 103) event
ReadyToShareScreenDelegate (see page 95)	Delegate for ReadyToShareScreen (see page 103) event
RecordingReadyToStartDelegate (see page 95)	This is nested type iConfServer.NET.iConfServerDotNet.RecordingReadyToStartDelegate.
RegistrationResponseDelegate (see page 96)	Delegate for RegistrationResponse (see page 104) Event
RemoveContactResponseDelegate (see page 96)	Delegate for RemoveContactResponse (see page 104) event
ScreenControlDeniedDelegate (see page 96)	Delegate for ScreenControlDenied (see page 104) event
ScreenControlGrantedDelegate (see page 96)	Delegate for ScreenControlGranted (see page 104) event
ScreenControlRequestDelegate (see page 96)	Delegate for ScreenControlRequest (see page 104) event
ScreenSharingSessionAcceptedDelegate (see page 97)	Delegate for ScreenSharingSessionAccepted (see page 104) event
ScreenSharingSessionAvailableDelegate (see page 97)	Delegate for the ScreenSharingSessionAvailable (see page 104) event
ScreenSharingSessionRejectedDelegate (see page 97)	Delegate for ScreenSharingSessionRejected (see page 105) event
SearchResponseDelegate (see page 97)	This is nested type iConfServer.NET.iConfServerDotNet.SearchResponseDelegate.





StatusMessagesReceivedDelegate (see page 98)	This is nested type iConfServer.NET.iConfServerDotNet.StatusMessagesReceivedDelegate.
TalkingDelegate (see page 98)	Delegate for Talking (see page 105) event
TextMessageReceivedDelegate (see page 98)	Delegate for the TextMessageReceived (see page 105) event
UserJoinedConferenceDelegate (see page 98)	Delegate for UserJoinedConference (see page 105) event
UserLeftConferenceDelegate (see page 98)	Delegate for UserLeftConference (see page 105) event
UserProfileReceivedDelegate (see page 99)	This is nested type iConfServer.NET.iConfServerDotNet.UserProfileReceivedDelegate.
VideoDeviceSelectedDelegate (see page 99)	Delegate for VideoDeviceSelected (see page 106) event
VideoPreviewStartedDelegate (see page 99)	Delegate for VideoPreviewStarted (see page 106) event
ViewUserResponseReceivedDelegate (see page 99)	Delegate for ViewUserResponse event

iConfServerDotNet Enumerations

	Name	Description
	AudioPlayBackType (see page 46)	--deprecated 0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.
	audioType (see page 46)	enum representing the type of system that should be used for audio Windows Audio System or DirectSound
	DesktopSharingType (see page 46)	This is record iConfServer.NET.iConfServerDotNet.DesktopSharingType.
	H264Profile (see page 47)	This is record iConfServer.NET.iConfServerDotNet.H264Profile.
	h264SpeedLevel (see page 47)	The speed level of the H.264 encoder
	RateControl (see page 47)	This is record iConfServer.NET.iConfServerDotNet.RateControl.






iConfServerDotNet Events

	Name	Description
	AddContactRequestResponse (see page 100)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.AddUserAsContact Method (see page 49)
	CannotRunGraph (see page 100)	Fired when the DirectX Video preview graph fails to run. Usually due to a failure to communicate with the video capture device (device might be in use)
	ChangePasswordReponse (see page 100)	fired when a response is received from the central server after a request to change password was made
	ClientConnectionLost (see page 100)	Fired when an iConf Client disconnects from the iConf Central Server Service use this event when you want to capture the handle of the client that has disconnected
	ClientDisconnected (see page 100)	Fired when an iConf Client disconnects from the iConf Central Server Service
	ConferenceMessageReceived (see page 100)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.SendMessageToConference Method (see page 70)

	ConferencePrivateMessageReceived (see page 100)	Fired when a private message is received from another user (see iConfServerDotNet.SendPrivatemessageToUser Method (see page 70))
	ConferenceRequest (see page 100)	Fired when a new conference request is received from a user
	ConferenceRequestRejected (see page 101)	Fired when a conference request has been rejected
	ContactListReceived (see page 101)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetContactList Method (see page 55)
	ContactStatusChanged (see page 101)	Fired when a user in the logged in user's contact list changes status
	ContactStatusMessageChanged (see page 101)	Fired when a user changes his/her status message
	CustomObjectReceived (see page 101)	Fired when a custom object is received from another user
	DisconnectedFromCentralServer (see page 101)	Fired when an iConf Server disconnects from the iConf Central Server Service
	FileDownloadCompleted (see page 101)	Fired when a file download completes successfully
	FileDownloadProgress (see page 101)	Use thsi event to report on the progress of a file download
	IncomingCall (see page 102)	Fired when a Call Request is received from an iConf Client
	InvalidConferenceRoomPassword (see page 102)	Fired when an attempt to join a password protected conference room fails due to an invalid password
	JoinedConferenceRoom (see page 102)	This is JoinedConferenceRoom, a member of class iConfServerDotNet.
	Log (see page 102)	This is Log, a member of class iConfServerDotNet.
	LoginResponse (see page 102)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.Login Method (see page 63)
	LogoutResponse (see page 102)	Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.LogoutResponse Method (see page 64)
	MotionDetected (see page 102)	Fired when motion is detected in the video preview
	MotionNotDetected (see page 102)	fired when motion is not detected in the video preview
	NewContactRequest (see page 103)	Fired when a new contact request is received from another user
	NewDesktopAvailable (see page 103)	This is NewDesktopAvailable, a member of class iConfServerDotNet.
	NewFileShareRequest (see page 103)	Fired when a file share request is received from a user (a user wants to share a file with you)
	NewVideoFrameAvailable (see page 103)	Fired when a new Video frame is available (Bitmap) for processing
	NewVideoFrameBytesAvailable (see page 103)	Fired when a new Video frame is available (byte[]) for processing
	PendingContactRequest (see page 103)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetPendingContactRequests Method (see page 56)
	ProfilePictureUpdated (see page 103)	Fired when the logged in user's profile picture gets successfully updated, following a call to the iConfServerDotNet.UpdateProfilePicture Method (see page 86)
	ReadyToShareScreen (see page 103)	Fired when a screen is ready to be shared following a call to StartScreenSharing (see page 83)
	RecordingReadyToStart (see page 104)	This is RecordingReadyToStart, a member of class iConfServerDotNet.




	RegistrationResponse (see page 104)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RegisterUser Method (see page 65)
	RemoveContactResponse (see page 104)	Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RemoveContact Method (see page 67)
	ScreenControlDenied (see page 104)	fired when a request to control the screen has been denied (central server mode)
	ScreenControlGranted (see page 104)	fired when a request to control the screen has been granted (central server mode)
	ScreenControlRequest (see page 104)	fired when a request to control the screen is received while in a screen sharing session
	ScreenSharingSessionAccepted (see page 104)	Fired when an iconfclient.net has accepted a screen sharing session request
	ScreenSharingSessionAvailable (see page 104)	Fired when a screen sharing session becomes available
	ScreenSharingSessionRejected (see page 105)	Fired when an iconfclient.net has rejected a screen sharing session request
	SearchResponse (see page 105)	This is SearchResponse, a member of class iConfServerDotNet.
	StatusMessagesReceived (see page 105)	This is StatusMessagesReceived, a member of class iConfServerDotNet.
	Talking (see page 105)	Fired when a user starts talking - This event will only work properly when Voice Activation Detection is enabled (see EnableVoiceActivation (see page 52) function).
	TextMessageReceived (see page 105)	Fired when a text message is received by from an iConf Client
	UserJoinedConference (see page 105)	Fired when a user joins a conference that the current logged in user is participating in.
	UserLeftConference (see page 105)	Fired when a user leaves a conference that the current logged in user is participating in.
	UserProfileReceived (see page 105)	This is UserProfileReceived, a member of class iConfServerDotNet.
	VideoDeviceSelected (see page 106)	Fired when a new video device has been selected after calling SelectVideoDevice (see page 69) - After this event is fired, new video sizes might be available and if you are displaying a list video sizes, you should reload that list
	VideoPreviewStarted (see page 106)	This is VideoPreviewStarted, a member of class iConfServerDotNet.
	ViewUserResponseReceived (see page 106)	Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.ViewUserResponseReceived Method (see page 86)

iConfServerDotNet Fields
































	Name	Description
	H263 (see page 48)	Constant representing the H263 codec selection. Can be used as a replacement for typing the string "H263"
	H264 (see page 48)	Constant representing the H264 codec selection. Can be used as a replacement for typing the string "H264"
	MPEG4 (see page 48)	Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"
	mycallbackid (see page 48)	This is mycallbackid, a member of class iConfServerDotNet.
	WM_COPYDATA (see page 48)	This is WM_COPYDATA, a member of class iConfServerDotNet.

	WM_USER (see page 48)	This is WM_USER, a member of class iConfServerDotNet.
---	---	---






















iConfServerDotNet Methods

	Name	Description
	AcceptCall (see page 48)	Function used to accept a call from an iConf Client component
	AcceptContactRequest (see page 49)	Function used to accept a contact request with a specific user
	AddUserAsContact (see page 49)	Used to send a request to a user to be added as a contact
	AutoJoinRoomOnLogin (see page 49)	Used to automatically Join a room upon logging in to the Central Server
	CaptureFrame (see page 49)	Used to capture a video frame to a file or clipboard
	ChangePassword (see page 50)	Function used to change the password used to log into the central server
	ChangeStatus (see page 50)	Used to change the status of the user logged into the Central Server
	CRLFDelimitedStringToList (see page 50)	This is CRLFDelimitedStringToList, a member of class iConfServerDotNet.
	DeleteProfilePicture (see page 51)	Used to delete a the current logged in user's profile picture from the central server service.
	DeleteUser (see page 51)	Deletes a user from the central server
	DenyScreenControl (see page 51)	This is DenyScreenControl, a member of class iConfServerDotNet.
	DisconnectClient (see page 51)	Use this function to force a connected client to disconnect
	DisconnectFromCentralServer (see page 52)	This is DisconnectFromCentralServer, a member of class iConfServerDotNet.
	Dispose (see page 52)	Clean up any resources being used.
	EnableEchoCancellation (see page 52)	Enable/Disable echo cancellation
	EnableGrayScale (see page 52)	Enables/Disables grayscale video stream
	EnableVoiceActivation (see page 52)	Enables/disables voice activation
	GetAssemblyVersion (see page 53)	returns the version of the assembly
	GetAudioBytesReceived (see page 53)	Gets the number of audio bytes received
	GetAudioCompressorsForRecording (see page 53)	This function returns the list of installed Audio Compressors that can be used for recording
	GetAudioCompressorsForRecordingStr (see page 53)	Used to Get a character delimited string of installed Audio Compressors that can be used for recording. use only when using component as COM.
	GetAudioInputDevices (see page 53)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method
	GetAudioInputDevicesList (see page 54)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method
	GetAudioInputLevel (see page 54)	deprecated Gets the audio input (microphone) level range is 0 - 65535
	GetAudioInputLevel (see page 54)	Used to get the current audio input level - useful to display a vu meter in an application
	GetAudioOutputDevices (see page 54)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method




✦	GetAudioOutputDevicesList (see page 54)	Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method
✦	GetAudioOutputVolume (see page 54)	deprecated Gets the audio output volume level
✦	GetBoundIP (see page 55)	Use this function to get the IP address to which the server is bound
✦	GetBoundPort (see page 55)	Use this function to get the Port to which the iConf Server is bound
✦	GetConferenceRooms (see page 55)	Used to get list of existing conference rooms
✦	GetContactList (see page 55)	Function used to get the list of contacts for the current logged in user
✦	GetFile (see page 55)	This is GetFile, a member of class iConfServerDotNet.
✦	GetLocalIp (see page 55)	Helper function used to get the list of local IP Addresses
✦	GetLocalIpStr (see page 56)	Helper function used to get the list of local IP Addresses
✦	GetMyCallBackId (see page 56)	Used to get the CallBackId of the iConfServer when in IP to IP mode
✦	GetPendingContactRequests (see page 56)	Function used to gets the pending contact requests for the current logged in user
✦	GetProfilePicture (see page 56)	Used to get the profile picture of a iConf Central Server Service user Requires Central Server to have specified Admin profile port
✦	GetStatusMessages (see page 56)	Used to get a list of status messages for the current logged in user's contacts
✦	GetUserProfile (see page 56)	Function used to get the profile of a user
✦	GetUserScr (see page 57)	This is GetUserScr, a member of class iConfServerDotNet.
✦	GetVideoCompressorsForRecording (see page 57)	This function returns the list of installed Video Compressors that can be used for recording
✦	GetVideoCompressorsForRecordingStr (see page 57)	Used to Get a character delimited string of installed Video Compressors that can be used for recording. use only when using component as COM.
✦	GetVideoDevices (see page 57)	Use this method to build a list of Video Capture devices found on the system
✦	GetVideoDevicesStr (see page 58)	Used to Get a character delimited string of Video Devices. use only when using component as COM.
✦	GetVideoInputs (see page 58)	Gets a List of video inputs available for the selected video device
✦	GetVideoInputsStr (see page 58)	Gets a Comma Separated string of video inputs available for the selected video device (COM use only)
✦	GetVideoSizes (see page 58)	Gets the List of available video preview sizes This method should only be called after a call to SelectVideoDevice (see page 69) because each video device has different video available video sizes
✦	GetVideoSizesStr (see page 59)	Gets the list of available video preview sizes in the form of string separated by a specified delimiter
✦	GetVolume (see page 59)	Used to get the Volume of an Audio conference
✦	GrantScreenControl (see page 59)	This is GrantScreenControl, a member of class iConfServerDotNet.
✦	HoldRecording (see page 59)	hold the recording after starting the recording process
✦	InitializeAudioSystem (see page 60)	Use this function to initialize the audio system Initializing the audio system is required - When not called explicitly, the audio system will be initialized with some default parameters
✦	InviteClientToViewScreen (see page 60)	In IP to IP mode, this function sends an invitation to a selected or all connected clients using their socket handle









	InviteUsersToViewScreen (see page 60)	When logged into the Central Server, this function allows to send an invitation to attend a screen sharing session to a number of users
	IsConnected (see page 61)	checks connection to central server
	JoinConference (see page 61)	Function used to Join a conference using its 'conference id'
	JoinConferenceRoom (see page 62)	Used to join/create a conference room. If you try to join a conference room which does not exist it will be created as a new conference room. After the last user leaves the conference room, the room gets deleted from the server
	LeaveConference (see page 62)	Function used to leave the conference that the logged in user is currently participating in
	LeaveCurrentConferenceRoom (see page 63)	Leaves current conference room
	Listen (see page 63)	starts listening for incoming connections on the specified ip and port
	Logging (see page 63)	enables / disables logging log file is in temp path iconflog.txt
	Login (see page 63)	Function used to Log (see page 102) into the Central Server
	Logout (see page 64)	Function used to log out of the Central Server
	LoopBackAudioTest (see page 64)	This is LoopBackAudioTest, a member of class iConfServerDotNet.
	Mute (see page 64)	Mutes/Un-Mute incoming audio playback
	MuteContactInConference (see page 64)	used to /un-mute the audio of a contact while in a conference
	Pause (see page 64)	Pause/unpause video & audio streaming to connected clients
	ReconfigureAudioSystem (see page 65)	Use this function to reconfigure the audio system (choose different audio input/output device)
	RecordToNewFileNow (see page 65)	During recording, invoking RecordToNewFile closes the current recording file, and creates a new recording file.
	RegisterUser (see page 65)	Function used to Register a new user with a Central Server
	RejectCall (see page 66)	Function used to reject an incoming call from an iConf Client
	RejectCallEx (see page 66)	Function used to reject an incoming call from an iConf Client
	RejectConferenceRequest (see page 66)	use this function to reject a Conference Request when getting conference request
	RemoveContact (see page 67)	Used to remove a contact
	RemoveNoiseFromOutgoingAudio (see page 67)	Enables/Disables removing noise from outgoing audio
	RequestScreenControl (see page 67)	DEPRECATED Used to request control while in a screen sharing session (Central Server mode ONLY)
	ResumeRecording (see page 67)	Resumes the recording process previously heldby calling HoldRecording (see page 59)
	RevokeScreenControl (see page 67)	This is RevokeScreenControl, a member of class iConfServerDotNet.
	Search (see page 68)	This is Search, a member of class iConfServerDotNet.
	SelectIPCameraAsVideoSource (see page 68)	Used to select an IP Camera as video source instead of a video capture device
	SelectVideoCaptureDeviceAsVideoSource (see page 69)	Used to select a video file as video source instead of a video capture device
	SelectVideoDevice (see page 69)	Use this function to select a video capture device using its 0 based index
	SelectVideoFileAsVideoSource (see page 69)	Used to select a file as video source instead of a video capture device
	SelectVideoInput (see page 69)	Used to select a video Input

◆	SendCustomObject (see page 70)	Used to send a custom string to a list of users logged into the Central Server Can be used to send any type of object to a list of users logged into the Central Server. Example serialize object as string => SendCustomObject => CustomObjectReceived (see page 101) => Deserialize back to object
◆	SendKeepAlive (see page 70)	Sends a KEEP ALIVE message to the Central Server
◆ S	SendMessage (see page 70)	For use with WM_COPYDATA (see page 48) and COPYDATASTRUCT (see page 46)
◆	SendMessageToConference (see page 70)	Sends text message to Conference - Message 500 chars max!
◆	SendPrivatemessageToUser (see page 70)	When logged into the Central server, use this function to send a private message to a user
◆	SendShareFileRequest (see page 71)	Use this function to share a file with a user
◆	SendTextMessage (see page 71)	Use this function to send a Text message to all or a specific connected iConfClient.NET (see page 111)
◆	SetAudioInputLevel (see page 71)	deprecated Sets the audio input (microphone) level
◆	SetAudioOutputVolume (see page 71)	deprecated Sets the Audio Output Volume
◆	SetAudioPlaybackType (see page 72)	deprecated
◆	SetBrightness (see page 72)	Sets the brightness of the video stream 0 to reset
◆	SetContrast (see page 72)	Sets the contrast of the video stream. Use 0 to reset
◆	SetEncoderProperties (see page 73)	Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..
◆	SetEncoderPropertiesEx (see page 74)	Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..
◆	SetFrameCaptureInterval (see page 75)	Sets the number of frames to skip (no encoding and sending) between each captured frame.
◆	SetGainLevel (see page 75)	This is SetGainLevel, a member of class iConfServerDotNet.
◆	SetHue (see page 76)	Sets the hue of the video stream. Use 0 to reset
◆	SetIEMode (see page 76)	Must be set when using component in Internet Explorer
◆	SetOutputGain (see page 76)	This is SetOutputGain, a member of class iConfServerDotNet.
◆	SetPreviewFrameRate (see page 76)	Desired frame rate, expressed in frame per second
◆	SetSaturation (see page 76)	Sets the Saturation of the video stream 0 to reset
◆	SetStatusMessage (see page 77)	Sets the status message of the user currently logged into the Central Server
◆	SetUnlockCode (see page 77)	Used with the registered version of the iConf .NET SDK This function should be called before calling the StartPreview (see page 79) function
◆	SetVoiceActivationLevel (see page 77)	Sets the voice activation level
◆	SetVolume (see page 77)	Used to set the Volume of an Audio conference
◆	ShareFile (see page 78)	Use this function to send a file to the participants of a conference
◆	ShowVideoCaptureDevicePropertyDialog (see page 78)	Used to display the video capture device specific property dialog where sometimes settings like face tracking, brightness, backlight, hue, saturation and other device settings can be changed.
◆	StartConference (see page 78)	Used to start a conference with a selected number of users
◆	StartConferenceEx (see page 79)	Used to start a conference with a selected number of users
◆	StartMotionDetection (see page 79)	Enables motion detection in the video preview using teh specified sensitivity

	StartPreview (see page 79)	Use this function to start previewing video using the selected video capture device (see the <code>M:iConfServer.NET.iConfServerDotNet.SelectVideoDevice(System.Int32)SelectVideoDevice</code> (see page 69)function)
	StartRecording (see page 80)	Function used to record the activity of a window along with Audio. Can be used to record a conferencing session
	StartRecordingAudioConference (see page 80)	Records the audio from a conference to a wav file
	StartRecordingCurrentVideoPreview (see page 80)	Record the current video preview (with or without audio) to a wmv file
	StartRecordingCurrentVideoPreviewWmv9 (see page 82)	Record the current video preview (with or without audio) to a wmv 9 encoded file using a custom profile
	StartRecordingToWMV (see page 83)	Function used to record the activity of a window along with Audio. Can be used to record a conferencing session
	StartScreenSharing (see page 83)	This is StartScreenSharing, a member of class <code>iConfServerDotNet</code> .
	StartTransmittingAudio (see page 84)	Starts transmitting audio. Only call this function if you have explicitly called the <code>StopTransmittingAudio</code> (see page 85) function
	StopMotionDetection (see page 84)	Stops the motion detection
	StopPreview (see page 84)	Stops the video preview and ends the encoding of the video
	StopRecording (see page 84)	Used to stop a recording session started by calling the <code>StartRecording</code> (see page 80) function
	StopRecordingAudioConference (see page 85)	Stops the process of recording the audio conference (process started by calling <code>StartRecordingAudioConference</code> (see page 80))
	StopRecordingCurrentVideoPreview (see page 85)	Used to stop recording the current video preview function
	StopScreenSharing (see page 85)	Stops a screen sharing session
	StopTransmittingAudio (see page 85)	Stops transmitting audio
	TextOverLay (see page 85)	Enables/Disables Text Overlay
	UpdateProfilePicture (see page 86)	Asynchronous function used to Update the profile picture of the user currently logged into the Central Server.
	ViewUser (see page 86)	Send request to central server to view the video feed of a user
	waveOutGetVolume (see page 86)	This is waveOutGetVolume, a member of class <code>iConfServerDotNet</code> .
	waveOutSetVolume (see page 86)	This is waveOutSetVolume, a member of class <code>iConfServerDotNet</code> .
	XmitDesktop (see page 86)	This is XmitDesktop, a member of class <code>iConfServerDotNet</code> .

iConfServerDotNet Properties

	Name	Description
	CallbackId (see page 87)	the Callback Id of the iConf Server. What is a Callback Id ? A callback id is an identifier that gets used in the case of an incoming call, to determine if the incoming call can be automatically accepted.m
	CallbackObject (see page 87)	This is CallbackObject, a member of class <code>iConfServerDotNet</code> .
	ConnectedClientCount (see page 87)	returns the number of clients connected to the <code>iConfServer.NET</code> (see page 25) component in IP to IP mode

	CurrentVideoCodec (see page 87)	The selected video codec H263 (see page 48) / H264 (see page 48) /MPEG4 (see page 48)
	IsConnectedToCentralServer (see page 87)	Checks to see if there is an active connection to the Central Server.
	IsListening (see page 87)	This is IsListening, a member of class iConfServerDotNet.
	IsScreenSharingEnabled (see page 87)	Returns true if screen sharing is turned on, false otherwise
	PreviewHeight (see page 88)	returns teh height of the video preview
	PreviewWidth (see page 88)	returns the width of the video preview
	VideoDeviceCount (see page 88)	returns the number of video devices installed on the system
	VideoRateControl (see page 88)	This is VideoRateControl, a member of class iConfServerDotNet.

iConfServerDotNet Structures

	Name	Description
	COPYDATASTRUCT (see page 46)	Used for WM_COPYDATA (see page 48) for string messages

4.1.1.1.1 iConfServerDotNet.iConfServerDotNet Constructor

Initializes a new instance of the iConfServerDotNet class

C#

```
public iConfServerDotNet();
```

4.1.1.1.2 iConfServerDotNet Structures

4.1.1.1.2.1 iConfServer.NET.iConfServerDotNet.COPYDATASTRUCT Structure

C#

```
public struct COPYDATASTRUCT {
    public IntPtr dwData;
    public int cbData;
    public string lpData;
}
```

File

iConfServerDotNet.cs

Remarks

Used for WM_COPYDATA ([see page 48](#)) for string messages

4.1.1.1.3 iConfServerDotNet Enumerations

4.1.1.1.3.1 iConfServer.NET.iConfServerDotNet.AudioPlayBackType Enumeration

--deprecated 0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low lantency. 3 = JITTER uses a jitter buffer for keeping low latency.

C#

```
public enum AudioPlayBackType {
    STABLE = 0,
    LOWLATENCY = 1,
    VARIABLE = 2,
}
```

```
JITTER = 3  
}
```

File

iConfServerDotNet.cs

4.1.1.1.3.2 iConfServer.NET.iConfServerDotNet.audioType Enumeration

enum representing the type of system that should be used for audio Windows Audio System or DirectSound

C#

```
public enum audioType {  
    WindowsAudio = 1,  
    DirectSound = 2  
}
```

File

iConfServerDotNet.cs

4.1.1.1.3.3 iConfServer.NET.iConfServerDotNet.DesktopSharingType Enumeration

C#

```
public enum DesktopSharingType {  
    Full = 0  
}
```

File

iConfServerDotNet.cs

Remarks

This is record iConfServer.NET.iConfServerDotNet.DesktopSharingType.

4.1.1.1.3.4 iConfServer.NET.iConfServerDotNet.H264Profile Enumeration

C#

```
public enum H264Profile {  
    HIGH444 = 1,  
    HIGH422 = 2,  
    HIGH10 = 3,  
    HIGH = 4,  
    MAIN = 5,  
    BASELINE = 6  
}
```

File

iConfServerDotNet.cs

Remarks

This is record iConfServer.NET.iConfServerDotNet.H264Profile.

4.1.1.1.3.5 iConfServer.NET.iConfServerDotNet.h264SpeedLevel Enumeration

The speed level of the H.264 encoder

C#

```
public enum h264SpeedLevel {  
    LOW_SPEED = 1,  
    MEDIUM_LOW = 2,  
    MEDIUM_HIGH = 3,  
    HIGH_SPEED = 4  
}
```

```
}
```

File

iConfServerDotNet.cs

Description

```
LOW_SPEED = 1,  
MEDIUM_LOW = 2,  
MEDIUM_HIGH = 3,  
HIGH_SPEED = 4
```

4.1.1.1.3.6 iConfServer.NET.iConfServerDotNet.RateControl Enumeration

C#

```
public enum RateControl {  
    ConstantBitRate = 0,  
    ConstantQuality = 1  
}
```

File

iConfServerDotNet.cs

Remarks

This is record iConfServer.NET.iConfServerDotNet.RateControl.

4.1.1.1.4 iConfServerDotNet Fields

4.1.1.1.4.1 iConfServerDotNet.H263 Field

Constant representing the H263 codec selection. Can be used as a replacement for typing the string "H263"

C#

```
public const string H263 = "H263";
```

4.1.1.1.4.2 iConfServerDotNet.H264 Field

Constant representing the H264 codec selection. Can be used as a replacement for typing the string "H264"

C#

```
public const string H264 = "H264";
```

4.1.1.1.4.3 iConfServerDotNet.MPEG4 Field

Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"

C#

```
public const string MPEG4 = "MPEG4";
```

4.1.1.1.4.4 iConfServerDotNet.mycallbackid Field

C#

```
public string mycallbackid;
```

Remarks

This is mycallbackid, a member of class iConfServerDotNet.

4.1.1.1.4.5 iConfServerDotNet.WM_COPYDATA Field

C#

```
public const int WM_COPYDATA = 0x4A;
```

Remarks

This is WM_COPYDATA, a member of class iConfServerDotNet.

4.1.1.1.4.6 iConfServerDotNet.WM_USER Field

C#

```
public const int WM_USER = 0x400;
```

Remarks

This is WM_USER, a member of class iConfServerDotNet.

4.1.1.1.5 iConfServerDotNet Methods

4.1.1.1.5.1 iConfServerDotNet.AcceptCall Method

Function used to accept a call from an iConf Client component

C#

```
public void AcceptCall(  
    string acceptMessage,  
    int socketHandle  
);
```

Parameters

Parameters	Description
string acceptMessage	message to send to the client in response
int socketHandle	socket handle of the client that is calling

4.1.1.1.5.2 iConfServerDotNet.AcceptContactRequest Method

Function used to accept a contact request with a specific user

C#

```
public void AcceptContactRequest(  
    string withContact  
);
```

Parameters

Parameters	Description
string withContact	

4.1.1.1.5.3 iConfServerDotNet.AddUserAsContact Method

Used to send a request to a user to be added as a contact

C#

```
public void AddUserAsContact(  
    string userName  
);
```

Parameters

Parameters	Description
string userName	the user that needs to be added as a contact

4.1.1.1.5.4 iConfServerDotNet.AutoJoinRoomOnLogin Method

Used to automatically Join a room upon logging in to the Central Server

C#

```
public void AutoJoinRoomOnLogin(  
    string roomToJoin  
) ;
```

Parameters

Parameters	Description
string roomToJoin	

4.1.1.1.5.5 iConfServerDotNet.CaptureFrame Method

Used to capture a video frame to a file or clipboard

C#

```
public bool CaptureFrame(  
    string fileName  
) ;
```

Parameters

Parameters	Description
string fileName	The full path to the file where the frame should be stored. when not specified (Empty String), the frame will be captured to the Windows Clipboard. The extension of the file name specifies the image type that will be used to save the file.

Returns

true if successful, false otherwise.

Example

```
iConfServer.CaptureFrame( @"C:\TempTest.jpg");
```

4.1.1.1.5.6 iConfServerDotNet.ChangePassword Method

Function used to change the password used to log into the central server

C#

```
public void ChangePassword(  
    string oldpassword,  
    string newpassword  
) ;
```

Parameters

Parameters	Description
string oldpassword	The logged in user's old password
string newpassword	The new desired password for the logged in user

4.1.1.1.5.7 iConfServerDotNet.ChangeStatus Method

Used to change the status of the user logged into the Central Server

C#

```
public void ChangeStatus(  
    string newStatus  
) ;
```

Parameters

Parameters	Description
string newStatus	The new status

Notes

The Central Server service uses the code of **NLN** for the online status and **OFF** for the Offline status.

You can use your own 3 character codes to identify a custom status.

Example : BSY for busy, DND for Do not disturb, LUN for at lunch etc..

4.1.1.1.5.8 iConfServerDotNet.CRLFDelimitedStringToIList Method

C#

```
protected IList<string> CRLFDelimitedStringToIList(  
    string CRLFString  
) ;
```

Remarks

This is CRLFDelimitedStringToIList, a member of class iConfServerDotNet.

4.1.1.1.5.9 iConfServerDotNet.DeleteProfilePicture Method

Used to delete a the current logged in user's profile picture from the central server service.

C#

```
public void DeleteProfilePicture() ;
```

4.1.1.1.5.10 iConfServerDotNet.DeleteUser Method

Deletes a user from the central server

C#

```
public void DeleteUser(  
    string centralServerIP,  
    Int32 centralServerTCPListeningPort,  
    string userToDelete,  
    string adminPassword  
) ;
```

Parameters

Parameters	Description
string centralServerIP	
Int32 centralServerTCPListeningPort	
string userToDelete	
string adminPassword	

4.1.1.1.5.11 DenyScreenControl Method

4.1.1.1.5.11.1 iConfServerDotNet.DenyScreenControl Method (int)

C#

```
public void DenyScreenControl(  
    int socketHandle  
) ;
```

Remarks

This is DenyScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.11.2 iConfServerDotNet.DenyScreenControl Method (string)

C#

```
public void DenyScreenControl(  
    string toUser  
) ;
```

Remarks

This is DenyScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.12 iConfServerDotNet.DisconnectClient Method

Use this function to force a connected client to disconnect

C#

```
public void DisconnectClient(  
    int socketHandle  
) ;
```

Parameters

Parameters	Description
int socketHandle	-1 to disconnect all connected clients, or the socket handle of the iConfClient.NET (see page 111) to disconnect

4.1.1.1.5.13 iConfServerDotNet.DisconnectFromCentralServer Method

C#

```
public void DisconnectFromCentralServer() ;
```

Remarks

This is DisconnectFromCentralServer, a member of class iConfServerDotNet.

4.1.1.1.5.14 iConfServerDotNet.Dispose Method

Clean up any resources being used.

C#

```
protected override void Dispose(  
    bool disposing  
) ;
```

4.1.1.1.5.15 iConfServerDotNet.EnableEchoCancellation Method

Enable/Disable echo cancellation

C#

```
public bool EnableEchoCancellation(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	set to true to enable echo cancellation

4.1.1.1.5.16 iConfServerDotNet.EnableGrayScale Method

Enables/Disables grayscale video stream

C#

```
public void EnableGrayScale(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	True enables the grayscale preview feature - False, disables it

4.1.1.1.5.17 iConfServerDotNet.EnableVoiceActivation Method

Enables/disables voice activation

C#

```
public bool EnableVoiceActivation(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	true to enable voice activation, false to disable it

4.1.1.1.5.18 iConfServerDotNet.GetAssemblyVersion Method

returns the version of the assembly

C#

```
public string GetAssemblyVersion() ;
```

4.1.1.1.5.19 iConfServerDotNet.GetAudioBytesReceived Method

Gets the number of audio bytes received

C#

```
public long GetAudioBytesReceived() ;
```

4.1.1.1.5.20 iConfServerDotNet.GetAudioCompressorsForRecording Method

This function returns the list of installed Audio Compressors that can be used for recording

C#

```
public ArrayList GetAudioCompressorsForRecording() ;
```

Returns

ArrayList of Audio Compressors

4.1.1.1.5.21 iConfServerDotNet.GetAudioCompressorsForRecordingStr Method

Used to Get a character delimited string of installed Audio Compressors that can be used for recording. use only when using component as COM.

C#

```
public string GetAudioCompressorsForRecordingStr(  
    string delimiter  
) ;
```

Parameters

Parameters	Description
string delimiter	the delimiter that should be used in the returned string

Returns

character delimited string of installed Audio Compressors

4.1.1.1.5.22 iConfServerDotNet.GetAudioInputDevices Method

Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method

C#

```
public Hashtable GetAudioInputDevices() ;
```

Returns

Hashtable

4.1.1.1.5.23 iConfServerDotNet.GetAudioInputDevicesList Method

Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method

C#

```
public List<KeyValuePair<int, string>> GetAudioInputDevicesList() ;
```

Returns

Hashtable

4.1.1.1.5.24 iConfServerDotNet.GetAudioInputlevel Method

deprecated Gets the audio input (microphone) level range is 0 - 65535

C#

```
public Int32 GetAudioInputlevel() ;
```

Returns

Integer : range is 0 - 65535

4.1.1.1.5.25 iConfServerDotNet.GetAudioInputLevel Method

Used to get the current audio input level - useful to display a vu meter in an application

C#

```
public int GetAudioInputLevel() ;
```

Returns

int value between 0 and 20

4.1.1.1.5.26 iConfServerDotNet.GetAudioOutputDevices Method

Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices (see page 53) method

C#

```
public Hashtable GetAudioOutputDevices();
```

Returns

Hashtable

4.1.1.1.5.27 iConfServerDotNet.GetAudioOutputDevicesList Method

Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method

C#

```
public List<KeyValuePair<int, string>> GetAudioOutputDevicesList();
```

Returns

List

4.1.1.1.5.28 iConfServerDotNet.GetAudioOutputVolume Method

deprecated Gets the audio output volume level

C#

```
public Int32 GetAudioOutputVolume();
```

Returns

range 0 .. 65535

4.1.1.1.5.29 iConfServerDotNet.GetBoundIP Method

Use this function to get the IP address to which the server is bound

C#

```
public string GetBoundIP();
```

Returns

string

4.1.1.1.5.30 iConfServerDotNet.GetBoundPort Method

Use this function to get the Port to which the iConf Server is bound

C#

```
public int GetBoundPort();
```

Returns

int

4.1.1.1.5.31 iConfServerDotNet.GetConferenceRooms Method

Used to get list of existing conference rooms

C#

```
public List<string> GetConferenceRooms();
```

4.1.1.1.5.32 iConfServerDotNet.GetContactList Method

Function used to get the list of contacts for the current logged in user

C#

```
public void GetContactList();
```

4.1.1.1.5.33 iConfServerDotNet.GetFile Method

C#

```
public void GetFile(  
    string fileId,  
    string originalFileName,  
    string saveFileTo  
) ;
```

Remarks

This is GetFile, a member of class iConfServerDotNet.

4.1.1.1.5.34 iConfServerDotNet.GetLocalIp Method

Helper function used to get the list of local IP Addresses

C#

```
public ArrayList GetLocalIp();
```

Returns

ArrayList containing a list of Local Ip Addresses

4.1.1.1.5.35 iConfServerDotNet.GetLocalIpStr Method

Helper function used to get the list of local IP Addresses

C#

```
public string GetLocalIpStr();
```

Returns

ArrayList containing a list of Local Ip Addresses

4.1.1.1.5.36 iConfServerDotNet.GetMyCallbackId Method

Used to get the CallbackId of the iConfServer when in IP to IP mode

C#

```
public string GetMyCallbackId();
```

Returns

the CallbackId of the iConfServer when in IP to IP mode

4.1.1.1.5.37 iConfServerDotNet.GetPendingContactRequests Method

Function used to gets the pending contact requests for the current logged in user

C#

```
public void GetPendingContactRequests();
```

4.1.1.1.5.38 iConfServerDotNet.GetProfilePicture Method

Used to get the profile picture of a iConf Central Server Service user Requires Central Server to have specified Admin profile port

C#

```
public Image GetProfilePicture(  
    string userName  
);
```

Parameters

Parameters	Description
string userName	The user name whose profile picture should be retrieved

Returns

an Image (JPEG)

4.1.1.1.5.39 iConfServerDotNet.GetStatusMessages Method

Used to get a list of status messages for the current logged in user's contacts

C#

```
public void GetStatusMessages();
```

4.1.1.1.5.40 iConfServerDotNet.GetUserProfile Method

Function used to get the profile of a user

C#

```
public void GetUserProfile(  
    string userName  
);
```

Parameters

Parameters	Description
username	The user whose profile should be retrieved

4.1.1.1.5.41 iConfServerDotNet.GetUserScr Method

C#

```
public Bitmap GetUserScr(  
    int userid  
);
```

Remarks

This is GetUserScr, a member of class iConfServerDotNet.

4.1.1.1.5.42 iConfServerDotNet.GetVideoCompressorsForRecording Method

This function returns the list of installed Video Compressors that can be used for recording

C#

```
public ArrayList GetVideoCompressorsForRecording();
```

Returns

ArrayList of Video Compressors

4.1.1.1.5.43 iConfServerDotNet.GetVideoCompressorsForRecordingStr Method

Used to Get a character delimited string of installed Video Compressors that can be used for recording. use only when using component as COM.

C#

```
public string GetVideoCompressorsForRecordingStr(  
    string delimiter  
) ;
```

Parameters

Parameters	Description
string delimiter	the delimiter that should be used in the returned string

Returns

character delimited string of installed Video Compressors

4.1.1.1.5.44 iConfServerDotNet.GetVideoDevices Method

Use this method to build a list of Video Capture devices found on the system

C#

```
public ArrayList GetVideoDevices();
```

Returns

ArrayList containing the names of the video capture devices found on the system

Example

```
private int LoadDevices( ComboBox cbDevices ) {  
    ArrayList lstVideoDevices = new ArrayList();  
  
    lstVideoDevices = iConfServer.GetVideoDevices();  
  
    for ( int i = 0; i < lstVideoDevices.Count; i++ )  
    {  
        cbDevices.Items.Add(lstVideoDevices[i]);  
    }  
  
    return cbDevices.Items.Count;  
}
```

4.1.1.1.5.45 iConfServerDotNet.GetVideoDevicesStr Method

Used to Get a character delimited string of Video Devices. use only when using component as COM.

C#

```
public string GetVideoDevicesStr(  
    string delimiter  
) ;
```

Parameters

Parameters	Description
string delimiter	the delimiter that should be used in the returned string

4.1.1.1.5.46 iConfServerDotNet.GetVideoInputs Method

Gets a List of video inputs available for the selected video device

C#

```
public ArrayList GetVideoInputs();
```

Returns

ArrayList containing video inputs available for selected device

4.1.1.1.5.47 iConfServerDotNet.GetVideoInputsStr Method

Gets a Comma Separated string of video inputs available for the selected video device (COM use only)

C#

```
public String GetVideoInputsStr(  
    string delimiter  
);
```

4.1.1.1.5.48 iConfServerDotNet.GetVideoSizes Method

Gets the List of available video preview sizes This method should only be called after a call to SelectVideoDevice (see page 69) because each video device has different video available video sizes

C#

```
public ArrayList GetVideoSizes();
```

Returns

ArrayList of video sizes

4.1.1.1.5.49 iConfServerDotNet.GetVideoSizesStr Method

Gets the list of available video preview sizes in the form of string separated by a specified delimiter

C#

```
public String GetVideoSizesStr(  
    string delimiter  
);
```

Returns

ArrayList

4.1.1.1.5.50 iConfServerDotNet.GetVolume Method

Used to get the Volume of an Audio conference

C#

```
public int GetVolume();
```

Returns

current volume

4.1.1.1.5.51 GrantScreenControl Method

4.1.1.1.5.51.1 iConfServerDotNet.GrantScreenControl Method (int)

C#

```
public void GrantScreenControl(  
    int socketHandle  
) ;
```

Remarks

This is GrantScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.51.2 iConfServerDotNet.GrantScreenControl Method (string)

C#

```
public void GrantScreenControl(  
    string toUser  
) ;
```

Remarks

This is GrantScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.52 iConfServerDotNet.HoldRecording Method

hold the recording after starting the recording process

C#

```
public void HoldRecording(  
    bool holdrecording  
) ;
```

Parameters

Parameters	Description
bool holdrecording	true to hold the recording after starting the recording process

4.1.1.1.5.53 iConfServerDotNet.InitializeAudioSystem Method

Use this function to initialize the audio system Initializing the audio system is required - When not called explicitly, the audio system will be initialized with some default parameters

C#

```
public bool InitializeAudioSystem(  
    audioType audioType,  
    int inputDevice,  
    int outputDevice,  
    int frequency,  
    int audioquality  
) ;
```

Parameters

Parameters	Description
audioType audioType	1 for Windows Sound System or 2 for DirectSound
int inputDevice	the device id of the audio input device
int outputDevice	the device id index of the audio playback device
int frequency	The frequency the audio input device should use. It can either be 8000, 16000 or 32000 (Hz)
int audioquality	the quality that the audio encoder should use.A value between 1 and 10 is accepted. 4 is an acceptable voice quality for most use.

Remarks

Although DirectSound offers audio improvements, it is not always supported, so please refer to your system configuration before using it.

4.1.1.1.5.54 iConfServerDotNet.InviteClientToViewScreen Method

In IP to IP mode, this function sends an invitation to a selected or all connected clients using their socket handle

C#

```
public void InviteClientToViewScreen(  
    int socketHandle,  
    string invitationMessage  
) ;
```

Parameters

Parameters	Description
int socketHandle	The socket handle of the client the invitation should be sent to. Use -1 to send to all connected clients
string invitationMessage	An optional invitation message that should be sent along with the request

4.1.1.1.5.55 InviteUsersToViewScreen Method**4.1.1.1.5.55.1 iConfServerDotNet.InviteUsersToViewScreen Method (ArrayList)**

When logged into the Central Server, this function allows to send an invitation to attend a screen sharing session to a number of users

C#

```
public void InviteUsersToViewScreen(  
    ArrayList usersToInvite  
) ;
```

Parameters

Parameters	Description
ArrayList usersToInvite	An ArrayList containing the user names of the users that should be invited to view the screen

4.1.1.1.5.55.2 iConfServerDotNet.InviteUsersToViewScreen Method (List<string>)

When logged into the Central Server, this function allows to send an invitation to attend a screen sharing session to a number of users

C#

```
public void InviteUsersToViewScreen(  
    List<string> usersToInvite  
) ;
```

Parameters

Parameters	Description
List<string> usersToInvite	An Generic Listcontaining the user names of the users that should be invited to view the screen

4.1.1.1.5.56 iConfServerDotNet.IsConnected Method

checks connection to central server

C#

```
public bool IsConnected(  
    Socket socket  
) ;
```

Parameters

Parameters	Description
Socket socket	

4.1.1.1.5.57 JoinConference Method**4.1.1.1.5.57.1 iConfServerDotNet.JoinConference Method (string)**

Function used to Join a conference using its 'conference id'

C#

```
public bool JoinConference(  
    string conferenceId  
) ;
```

Parameters

Parameters	Description
string conferenceId	the id of the conference that the user wants to join

4.1.1.1.5.57.2 iConfServerDotNet.JoinConference Method (string, string)

Function used to Join a conference using its 'conference id' and password

C#

```
public bool JoinConference(  
    string conferenceId,  
    string password  
) ;
```

Parameters

Parameters	Description
string conferenceId	the id of the conference that the user wants to join
string password	the password to use to join the conference room

4.1.1.1.5.58 JoinConferenceRoom Method**4.1.1.1.5.58.1 iConfServerDotNet.JoinConferenceRoom Method (string)**

Used to join/create a conference room. If you try to join a conference room which does not exist it will be created as a new conference room. After the last user leaves the conference room, the room gets deleted from the server

C#

```
public bool JoinConferenceRoom(  
    string roomName  
) ;
```

Parameters

Parameters	Description
string roomName	the name of the room that should be created or that you would like to join

Example

```
ics.JoinConferenceRoom("Entertainment");
```

4.1.1.1.5.58.2 iConfServerDotNet.JoinConferenceRoom Method (string, string)

Used to join/create a password protected conference room. If you try to join a conference room which does not exist it will be created as a new conference room. After the last user leaves the conference room, the room gets deleted from the server

C#

```
public void JoinConferenceRoom(  
    string roomName,  
    string password  
);
```

Parameters

Parameters	Description
string roomName	the name of the room that should be created or that you would like to join
string password	The password that should be used to enter the room or the password that other users should enter to join the room

4.1.1.1.5.59 iConfServerDotNet.LeaveConference Method

Function used to leave the conference that the logged in user is currently participating in

C#

```
public bool LeaveConference();
```

Parameters

Parameters	Description
conferenceld	

4.1.1.1.5.60 iConfServerDotNet.LeaveCurrentConferenceRoom Method

Leaves current conference room

C#

```
public void LeaveCurrentConferenceRoom();
```

4.1.1.1.5.61 iConfServerDotNet.Listen Method

starts listening for incoming connections on the specified ip and port

C#

```
public bool Listen(  
    bool enabled,  
    string ip,  
    int port,  
    int tcpAudioListeningPort,  
    int udpAudioDataPort  
);
```

Parameters

Parameters	Description
bool enabled	set to true to listen for incoming connections, false to stop listening
string ip	the ip address our server is bound to

int port	the port our server is listening on
int tcpAudioListeningPort	the tcp port used to listen for incoming audio requests
int udpAudioDataPort	the udp port used to transfer the audio data

Returns

true if successful, false otherwise

Remarks

Only applies to Peer to Peer mode of operation

4.1.1.1.5.62 iConfServerDotNet.Logging Method

enables / disables logging log file is in temp path iconflog.txt

C#

```
public void Logging(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	true enabled / false disabled

4.1.1.1.5.63 iConfServerDotNet.Login Method

Function used to Log (see page 102) into the Central Server

C#

```
public void Login(  
    string centralServerIP,  
    Int32 centralServerTCPListeningPort,  
    string userName,  
    string password  
) ;
```

Parameters

Parameters	Description
string userName	The registered user name that should be used to log into the central server
string password	The password associated with the registered user name

4.1.1.1.5.64 iConfServerDotNet.Logout Method

Function used to log out of the Central Server

C#

```
public void Logout() ;
```

4.1.1.1.5.65 iConfServerDotNet.LoopBackAudioTest Method**C#**

```
public void LoopBackAudioTest() ;
```

Remarks

This is LoopBackAudioTest, a member of class iConfServerDotNet.

4.1.1.1.5.66 iConfServerDotNet.Mute Method

Mutes/Un-Mute incoming audio playback

C#

```
public bool Mute(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	true to mute, false to un mute

Returns

true if successful, false otherwise

4.1.1.1.5.67 iConfServerDotNet.MuteContactInConference Method

used to /un-mute the audio of a contact while in a conference

C#

```
public bool MuteContactInConference(  
    string contactname,  
    bool mute  
) ;
```

Parameters

Parameters	Description
string contactname	the contact to mute
bool mute	true to mute, false to un-mute the contact

4.1.1.1.5.68 iConfServerDotNet.Pause Method

Pause/unpause video & audio streaming to connected clients

C#

```
public void Pause(  
    bool pauseVideo,  
    bool pauseAudio  
) ;
```

4.1.1.1.5.69 iConfServerDotNet.ReconfigureAudioSystem Method

Use this function to reconfigure the audio system (choose different audio input/output device)

C#

```
public bool ReconfigureAudioSystem(  
    audioType audioType,  
    int inputDevice,  
    int outputDevice  
) ;
```

Parameters

Parameters	Description
audioType audioType	1 for Windows Sound System or 2 for DirectSound (Directsound recommended)
int inputDevice	the device id of the audio input device
int outputDevice	the device id of the audio playback device

Remarks

Although DirectSound offers audio improvements, it is not always supported, so please refer to your system configuration before using it.

4.1.1.1.5.70 iConfServerDotNet.RecordToNewFileNow Method

During recording, invoking RecordToNewFile closes the current recording file, and creates a new recording file.

C#

```
public void RecordToNewFileNow(  
    string newRecordingFileName,  
    bool resetStreamTime  
) ;
```

Parameters

Parameters	Description
string newRecordingFileName	
bool resetStreamTime	

4.1.1.1.5.71 iConfServerDotNet.RegisterUser Method

Function used to Register a new user with a Central Server

C#

```
public void RegisterUser(  
    string centralServerIP,  
    Int32 centralServerTCPListeningPort,  
    string userName,  
    string password,  
    string extraData  
) ;
```

Parameters

Parameters	Description
string centralServerIP	The Ip address of the Computer where the Central Server is running as a service
Int32 centralServerTCPListeningPort	The port on which the Central Server is listening for incoming connections
string userName	The username that should be registered
string password	The password chosen for the user that is being registered
string extraData	extra data which will be stored along with the registration details

4.1.1.1.5.72 iConfServerDotNet.RejectCall Method

Function used to reject an incoming call from an iConf Client

C#

```
public void RejectCall(  
    string rejectionMessage,  
    IntPtr socketHandle  
) ;
```

Parameters

Parameters	Description
string rejectionMessage	message to send to the client that requested the call
IntPtr socketHandle	socket handle of the client that is calling

4.1.1.1.5.73 iConfServerDotNet.RejectCallEx Method

Function used to reject an incoming call from an iConf Client

C#

```
public void RejectCallEx(  
    string rejectionMessage,  
    int socketHandle  
);
```

Parameters

Parameters	Description
string rejectionMessage	message to send the the client that requested the call
int socketHandle	socket handle of the client that is calling

4.1.1.1.5.74 iConfServerDotNet.RejectConferenceRequest Method

use this function to reject a Conference Request when getting conference request

C#

```
public void RejectConferenceRequest(  
    string conferenceLeader,  
    string rejectReason  
);
```

Parameters

Parameters	Description
string conferenceLeader	The name of the user (usually the conference leader) who should receive the Rejection notification
reasonsForRejection	The reason why the conference request was rejected

4.1.1.1.5.75 iConfServerDotNet.RemoveContact Method

Used to remove a contact

C#

```
public void RemoveContact(  
    string userName  
);
```

Parameters

Parameters	Description
string userName	the user that needs to be added as a contact

4.1.1.1.5.76 iConfServerDotNet.RemoveNoiseFromOutgoingAudio Method

Enables/Disables removing noise from outgoing audio

C#

```
public bool RemoveNoiseFromOutgoingAudio(  
    bool enabled  
);
```

Parameters

Parameters	Description
bool enabled	true to enable/ false to disable

Returns

true if successful, false otherwise

4.1.1.1.5.77 iConfServerDotNet.RequestScreenControl Method

DEPRECATED Used to request control while in a screen sharing session (Central Server mode ONLY)

C#

```
public void RequestScreenControl(  
    string toUser  
) ;
```

Parameters

Parameters	Description
string toUser	The user to whom the request should go to

4.1.1.1.5.78 iConfServerDotNet.ResumeRecording Method

Resumes the recording process previously heldby calling HoldRecording ([🔗](#) see page 59)

C#

```
public void ResumeRecording() ;
```

4.1.1.1.5.79 RevokeScreenControl Method**4.1.1.1.5.79.1 iConfServerDotNet.RevokeScreenControl Method (int)****C#**

```
public void RevokeScreenControl(  
    int socketHandle  
) ;
```

Remarks

This is RevokeScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.79.2 iConfServerDotNet.RevokeScreenControl Method (string)**C#**

```
public void RevokeScreenControl(  
    string toUser  
) ;
```

Remarks

This is RevokeScreenControl, a member of class iConfServerDotNet.

4.1.1.1.5.80 iConfServerDotNet.Search Method**C#**

```
public void Search(  
    string searchString  
) ;
```

Remarks

This is Search, a member of class iConfServerDotNet.

4.1.1.1.5.81 SelectIPCameraAsVideoSource Method

4.1.1.1.5.81.1 iConfServerDotNet.SelectIPCameraAsVideoSource Method (string)

Used to select an IP Camera as video source instead of a video capture device

C#

```
public bool SelectIPCameraAsVideoSource(  
    string IPEndPoint  
) ;
```

Parameters

Parameters	Description
string IPEndPoint	the URL/IP address of the IP camera

Returns

true if successful, false if not

4.1.1.1.5.81.2 iConfServerDotNet.SelectIPCameraAsVideoSource Method (string, string, string)

Used to select an IP Camera as video source instead of a video capture device

C#

```
public bool SelectIPCameraAsVideoSource(  
    string IPEndPoint,  
    string username,  
    string password  
) ;
```

Parameters

Parameters	Description
string IPEndPoint	the URL/IP address of the IP camera
string username	the username that should be used to authenticate
string password	the password that should be used to authenticate

4.1.1.1.5.82 iConfServerDotNet.SelectVideoCaptureDeviceAsVideoSource Method

Used to select a video file as video source instead of a video capture device

C#

```
public bool SelectVideoCaptureDeviceAsVideoSource() ;
```

Parameters

Parameters	Description
fileName	The video file that should be used as video source

4.1.1.1.5.83 iConfServerDotNet.SelectVideoDevice Method

Use this function to select a video capture device using its 0 based index

C#

```
public bool SelectVideoDevice(  
    int deviceindex  
) ;
```

Parameters

Parameters	Description
int deviceindex	0 based index of the video capture device

Returns

true if successful in selecting a device, false otherwise

4.1.1.1.5.84 iConfServerDotNet.SelectVideoFileAsVideoSource Method

Used to select a file as video source instead of a video capture device

C#

```
public bool SelectVideoFileAsVideoSource(  
    string fileName  
) ;
```

Parameters

Parameters	Description
string fileName	The video file that should be used as video source

Returns

true if successful, false if not

4.1.1.1.5.85 iConfServerDotNet.SelectVideoInput Method

Used to select a video Input

C#

```
public void SelectVideoInput(  
    int inputIndex  
) ;
```

Parameters

Parameters	Description
int inputIndex	

4.1.1.1.5.86 iConfServerDotNet.SendCustomObject Method

Used to send a custom string to a list of users logged into the Central Server Can be used to send any type of object to a list of users logged into the Central Server. Example serialize object as string => SendCustomObject => CustomObjectReceived (see page 101) => Deserialize back to object

C#

```
public void SendCustomObject(  
    string serializedObjectString,  
    ArrayList toUsers  
) ;
```

Parameters

Parameters	Description
string serializedObjectString	
ArrayList toUsers	

4.1.1.1.5.87 iConfServerDotNet.SendKeepAlive Method

Sends a KEEP ALIVE message to the Central Server

C#

```
public void SendKeepAlive() ;
```

4.1.1.1.5.88 iConfServerDotNet.SendMessage Method

C#

```
public static extern int SendMessage(  
    int hWnd,  
    int Msg,  
    int wParam,  
    ref COPYDATASTRUCT lParam  
);
```

Remarks

For use with WM_COPYDATA (see page 48) and COPYDATASTRUCT (see page 46)

4.1.1.1.5.89 iConfServerDotNet.SendMessageToConference Method

Sends text message to Conference - Message 500 chars max!

C#

```
public void SendMessageToConference(  
    string messageToSend  
);
```

Parameters

Parameters	Description
string messageToSend	

4.1.1.1.5.90 iConfServerDotNet.SendPrivatemessageToUser Method

When logged into the Central server, use this function to send a private message to a user

C#

```
public void SendPrivatemessageToUser(  
    string messageToSend,  
    string sendToUser  
);
```

Parameters

Parameters	Description
string messageToSend	The message that should be sent (max 500 characters)
string sendToUser	The user name of the user the message should be sent to

4.1.1.1.5.91 iConfServerDotNet.SendShareFileRequest Method

Use this function to share a file with a user

C#

```
public void SendShareFileRequest(  
    string fileId,  
    string toUser  
);
```

Parameters

Parameters	Description
string fileId	The id of the file that should be shared. To get the id of a file, capture the return value of the iConfServerDotNet.ShareFile Method (see page 78)
string toUser	The user that the file should be shared with

4.1.1.1.5.92 iConfServerDotNet.SendTextMessage Method

Use this function to send a Text message to all or a specific connected iConfClient.NET (see page 111)

C#

```
public void SendTextMessage(  
    int socketHandle,  
    string messageToSend  
) ;
```

Parameters

Parameters	Description
int socketHandle	-1 to send to all clients, or the socket handle of the iConfClient.NET (see page 111) to send the text message to
string messageToSend	a string representing the text message that should be sent

4.1.1.1.5.93 iConfServerDotNet.SetAudioInputlevel Method

deprecated Sets the audio input (microphone) level

C#

```
public void SetAudioInputlevel(  
    Int32 inputLevel  
) ;
```

Parameters

Parameters	Description
Int32 inputLevel	range is : 0 - 65535

4.1.1.1.5.94 iConfServerDotNet.SetAudioOutputVolume Method

deprecated Sets the Audio Output Volume

C#

```
public void SetAudioOutputVolume(  
    int volume  
) ;
```

Parameters

Parameters	Description
int volume	integer value from 0 (min volume) to 65535 (max volume)

4.1.1.1.5.95 iConfServerDotNet.SetAudioPlaybackType Method

deprecated

C#

```
public bool SetAudioPlaybackType(  
    int playbackType  
) ;
```

Parameters

Parameters	Description
int playbackType	0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.

Remarks

see AudioPlayBackType (see page 46) enum

4.1.1.1.5.96 iConfServerDotNet.SetBrightness Method

Sets the brightness of the video stream 0 to reset

C#

```
public void SetBrightness(  
    int value  
) ;
```

Parameters

Parameters	Description
int value	

4.1.1.1.5.97 iConfServerDotNet.SetContrast Method

Sets the contrast of the video stream. Use 0 to reset

C#

```
public void SetContrast(  
    int value  
) ;
```

Parameters

Parameters	Description
int value	

4.1.1.1.5.98 SetEncoderProperties Method**4.1.1.1.5.98.1 iConfServerDotNet.SetEncoderProperties Method (VideoCodecs, int, int, int, int, int)**

Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..

C#

```
public void SetEncoderProperties(  
    VideoCodecs videoCodec,  
    int IFrameFrequency,  
    int bitRate,  
    int numBframes,  
    int h264speedlevel,  
    int h264profile  
) ;
```

Parameters

Parameters	Description
VideoCodecs videoCodec	The video codec to use (H.263 or H.264)

int IFrameFrequency	Sets the amount of of P-Frames before another I-Frame will be generated.
int bitRate	Sets the bitrate at which the frames will be encoded
int numBFrames	amount of B-Frames generated between I/P-Frames. Set to 0 to disable B-Frames Note : only applies to h.264 encoder - for h.263 set to 0
int h264speedlevel	This property allows to fine-tune the required Quality to Speed tradeoff 1 - LOW_SPEED Low Speed, High Quality 2- MEDIUM_LOW Medium Low Speed, Medium High Quality 3- MEDIUM_HIGH Medium High Speed, Medium Low Quality 4- HIGH_SPEED High Speed, Low Quality Note : only applies to h.264 encoder - for h.263 set to 0
int h264profile	H.264 Profiles (http://en.wikipedia.org/wiki/H.264#Profiles) 1 - HIGH444 2 - HIGH422 3 - HIGH10 4 - HIGH 5 - MAIN 6 - BASELINE Note : only applies to h.264 encoder - for h.263 set to 0
inputResolutionHeight	input Width of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8
inputResolutionWidth	input Height of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8

4.1.1.1.5.98.2 iConfServerDotNet.SetEncoderProperties Method (string, int, int, int, int, int, int, int)

Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..

C#

```
public void SetEncoderProperties(
    string videoCodec,
    int IFrameFrequency,
    int bitRate,
    int numBFrames,
    int h264speedlevel,
    int h264profile,
    int inputResolutionWidth,
    int inputResolutionHeight
);
```

Parameters

Parameters	Description
string videoCodec	The video codec to use (H.263 or H.264)
int IFrameFrequency	Sets the amount of of P-Frames before another I-Frame will be generated.
int bitRate	Sets the bitrate at which the frames will be encoded
int numBFrames	amount of B-Frames generated between I/P-Frames. Set to 0 to disable B-Frames Note : only applies to h.264 encoder - for h.263 set to 0

int h264speedlevel	This property allows to fine-tune the required Quality to Speed tradeoff 1 - LOW_SPEED Low Speed, High Quality 2- MEDIUM_LOW Medium Low Speed, Medium High Quality 3- MEDIUM_HIGH Medium High Speed, Medium Low Quality 4- HIGH_SPEED High Speed, Low Quality Note : only applies to h.264 encoder - for h.263 set to 0
int h264profile	H.264 Profiles (http://en.wikipedia.org/wiki/H.264#Profiles) 1 - HIGH444 2 - HIGH422 3 - HIGH10 4 - HIGH 5 - MAIN 6 - BASELINE Note : only applies to h.264 encoder - for h.263 set to 0
int inputResolutionWidth	input Height of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8
int inputResolutionHeight	input Width of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8

4.1.1.1.5.98.3 iConfServerDotNet.SetEncoderProperties Method (string, int, int, int, int, int, int, int, int)

DEPRECATED

C#

```
public void SetEncoderProperties(
    string videoCodec,
    int IFrameFrequency,
    int bitRate,
    int numBFrames,
    int h264speedlevel,
    int h264profile,
    int inputResolutionWidth,
    int inputResolutionHeight,
    int frameRate
);
```

4.1.1.1.5.99 iConfServerDotNet.SetEncoderPropertiesEx Method

Sets the Video Encoder properties Allows to choose a Video Codec, frame rate etc..

C#

```
public void SetEncoderPropertiesEx(
    string videoCodec,
    int IFrameFrequency,
    int bitRate,
    int numBFrames,
    int h264speedlevel,
    int h264profile
);
```

Parameters

Parameters	Description
string videoCodec	The video codec to use (H.263 or H.264)
int IFrameFrequency	Sets the amount of of P-Frames before another I-Frame will be generated.
int bitRate	Sets the bitrate at which the frames will be encoded

int numBFrames	amount of B-Frames generated between I/P-Frames. Set to 0 to disable B-Frames Note : only applies to h.264 encoder - for h.263 set to 0
int h264speedlevel	This property allows to fine-tune the required Quality to Speed tradeoff 1 - LOW_SPEED Low Speed, High Quality 2- MEDIUM_LOW Medium Low Speed, Medium High Quality 3- MEDIUM_HIGH Medium High Speed, Medium Low Quality 4- HIGH_SPEED High Speed, Low Quality Note : only applies to h.264 encoder - for h.263 set to 0
int h264profile	H.264 Profiles (http://en.wikipedia.org/wiki/H.264#Profiles) 1 - HIGH444 2 - HIGH422 3 - HIGH10 4 - HIGH 5 - MAIN 6 - BASELINE Note : only applies to h.264 encoder - for h.263 set to 0
inputResolutionHeight	input Width of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8
inputResolutionWidth	input Height of the frames to be encoded. Width and Height must be divisible by 4 and greater or equal to 8

4.1.1.1.5.100 iConfServerDotNet.SetFrameCaptureInterval Method

Sets the number of frames to skip (no encoding and sending) between each captured frame.

C#

```
public void SetFrameCaptureInterval(
    int value
);
```

Parameters

Parameters	Description
int value	number of frames to skip between each captured frame. Use 0 to capture and encode all frames

4.1.1.1.5.101 iConfServerDotNet.SetGainLevel Method

C#

```
public void SetGainLevel(
    Int32 value
);
```

Remarks

This is SetGainLevel, a member of class iConfServerDotNet.

4.1.1.1.5.102 iConfServerDotNet.SetHue Method

Sets the hue of the video stream. Use 0 to reset

C#

```
public void SetHue(
    int value
);
```

Parameters

Parameters	Description
int value	_nt_

4.1.1.1.5.103 iConfServerDotNet.SetIEMode Method

Must be set when using component in Internet Explorer

C#

```
public void SetIEMode(  
    bool enabled  
) ;
```

Parameters

Parameters	Description
bool enabled	

4.1.1.1.5.104 iConfServerDotNet.SetOutputGain Method**C#**

```
public void SetOutputGain(  
    Int32 value  
) ;
```

Remarks

This is SetOutputGain, a member of class iConfServerDotNet.

4.1.1.1.5.105 iConfServerDotNet.SetPreviewFrameRate Method

Desired frame rate, expressed in frame per second

C#

```
public void SetPreviewFrameRate(  
    int value  
) ;
```

Parameters

Parameters	Description
int value	Used to set or retrieve the desired frame rate, expressed in frames per second. Set it to 0 to use the video capture device default frame rate

4.1.1.1.5.106 iConfServerDotNet.SetSaturation Method

Sets the Saturation of the video stream 0 to reset

C#

```
public void SetSaturation(  
    int value  
) ;
```

Parameters

Parameters	Description
int value	

4.1.1.1.5.107 iConfServerDotNet.SetStatusMessage Method

Sets the status message of the user currently logged into the Central Server

C#

```
public void SetStatusMessage(  
    string statusMessage  
) ;
```

Parameters

Parameters	Description
string statusMessage	

4.1.1.1.5.108 iConfServerDotNet.SetUnlockCode Method

Used with the registered version of the iConf .NET SDK This function should be called before calling the StartPreview (see page 79) function

C#

```
public void SetUnlockCode(  
    string contextString  
) ;
```

Parameters

Parameters	Description
string contextString	

4.1.1.1.5.109 iConfServerDotNet.SetVoiceActivationLevel Method

Sets the voice activation level

C#

```
public bool SetVoiceActivationLevel(  
    int level  
) ;
```

Parameters

Parameters	Description
int level	voice activation level between 0 min and 20 max

Returns

true if successful, false otherwise

4.1.1.1.5.110 iConfServerDotNet.SetVolume Method

Used to set the Volume of an Audio conference

C#

```
public bool SetVolume(  
    int volumeLevel  
) ;
```

Parameters

Parameters	Description
int volumeLevel	a value between 0 (min) and 255 (max)

Returns

true if successful, false otherwise

4.1.1.1.5.111 iConfServerDotNet.ShareFile Method

Use this function to send a file to the participants of a conference

C#

```
public string ShareFile(  
    string fileToShare  
) ;
```

Parameters

Parameters	Description
fileToSend	The full path, including file name of the file to send

4.1.1.1.5.112 iConfServerDotNet.ShowVideoCaptureDevicePropertyDialog Method

Used to display the video capture device specific property dialog where sometimes settings like face tracking, brightness, backlight, hue, saturation and other device settings can be changed.

C#

```
public bool ShowVideoCaptureDevicePropertyDialog() ;
```

Returns

true if successful false if not

4.1.1.1.5.113 StartConference Method

4.1.1.1.5.113.1 iConfServerDotNet.StartConference Method (ArrayList)

Used to start a conference with a selected number of users

C#

```
public void StartConference(  
    ArrayList users  
) ;
```

Parameters

Parameters	Description
ArrayList users	An ArrayList containing the user names of users that should be invited to the conference

4.1.1.1.5.113.2 iConfServerDotNet.StartConference Method (ArrayList, string)

C#

```
public void StartConference(  
    ArrayList users,  
    string conferencePassword  
) ;
```

Remarks

This is StartConference, a member of class iConfServerDotNet.

4.1.1.1.5.114 iConfServerDotNet.StartConferenceEx Method

Used to start a conference with a selected number of users

C#

```
public void StartConferenceEx(  
    string users
```

```
);
```

Parameters

Parameters	Description
string users	A comma delimited string containing the user names of users that should be invited to the conference

4.1.1.1.5.115 iConfServerDotNet.StartMotionDetection Method

Enables motion detection in the video preview using teh specified sensitivity

C#

```
public void StartMotionDetection(  
    int sensitivity  
) ;
```

Parameters

Parameters	Description
int sensitivity	1 (min sensitivity) to 9 (max sensitivity)

4.1.1.1.5.116 iConfServerDotNet.StartPreview Method

Use this function to start previewing video using the selected video capture device (see the M:iConfServer.NET.iConfServerDotNet.SelectVideoDevice(System.Int32)SelectVideoDevice (🔗 see page 69)function)

C#

```
public bool StartPreview(  
    int videoSizeIndex  
) ;
```

Parameters

Parameters	Description
int videoSizeIndex	The 0 based index of the video size that should be used to preview the video. To get the list of available Video sizes, refer to M:iConfServer.NET.iConfServerDotNet.GetVideoSizesGetVideoSizes (🔗 see page 58)

Returns

True if successful, false if not successful

Remarks

GetVideoSizes (🔗 see page 58)StopPreview (🔗 see page 84)

Related Topics

iConfServer.NET.iConfServerDotNet.GetVideoSizes (🔗 see page 58), M:iConfServer.NET.iConfServerDotNet.StopPreview

4.1.1.1.5.117 iConfServerDotNet.StartRecording Method

Function used to record the activity of a window along with Audio. Can be used to record a conferencing session

C#

```
public bool StartRecording(  
    int windowHandle,  
    int videoCompressorIndex,  
    int audioCompressorIndex,  
    bool realtimeCompression,  
    bool recordAudio,
```

```
        string fileName  
    );
```

Parameters

Parameters	Description
int windowHandle	The handle of the window that should be recorded
int videoCompressorIndex	The 0 based index of the video compressor to use
int audioCompressorIndex	The 0 based index of the Audio compressor to use
bool realtimeCompression	set true if the resulting video should be compressed in realtime or false to compress when the recording stops
bool recordAudio	set to true to record audio and video false to only record audio
string fileName	The file to which the video should be saved

4.1.1.1.5.118 iConfServerDotNet.StartRecordingAudioConference Method

Records the audio from a conference to a wav file

C#

```
public void StartRecordingAudioConference(  
    string filename  
);
```

Parameters

Parameters	Description
string filename	the full path of the file the audio conference should be recorded to example test.wav

4.1.1.1.5.119 StartRecordingCurrentVideoPreview Method**4.1.1.1.5.119.1 iConfServerDotNet.StartRecordingCurrentVideoPreview Method (bool, string)**

Record the current video preview (with or without audio) to a wmv file

C#

```
public bool StartRecordingCurrentVideoPreview(  
    bool recordAudio,  
    string fileName  
);
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved

4.1.1.1.5.119.2 iConfServerDotNet.StartRecordingCurrentVideoPreview Method (bool, string, bool)

Record the current video preview (with or without audio) to a wmv file

C#

```
public bool StartRecordingCurrentVideoPreview(  
    bool recordAudio,  
    string fileName,  
    bool recordingIsPriority  
);
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved
bool recordingsIsPriority	Set to true to gives priority to the recording activity over the preview

4.1.1.1.5.119.3 iConfServerDotNet.StartRecordingCurrentVideoPreview Method (bool, string, int, int, bool, int, int, int)

Record the current video preview (with or without audio) to a wmv file

C#

```
public bool StartRecordingCurrentVideoPreview(
    bool recordAudio,
    string fileName,
    int audioBitRate,
    int numberAudioChannels,
    bool useFixedFrameRate,
    int videoBitRate,
    int maxKeyFrameSpacing,
    int videoQuality
);
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved
int audioBitRate	the bit rate of the audio stream in bits per second. E.g. a value of 40000 specifies about 40 Kbits per second. Use -1 to use the default value
int numberAudioChannels	Specifies the number of channels of the audio stream. 1 for 1 channel (mono), 2 for 2 channels stereo. Use -1 to use the default value
bool useFixedFrameRate	Specifies if a fixed frame rate must be used to record the video stream
int videoBitRate	Specifies the bit rate of the video stream in bits per second. Use -1 to use the default value
int maxKeyFrameSpacing	Maximum interval between key frames. E.g. 20000 = 1 key frame every 2 seconds. Use -1 to use the default value
int videoQuality	Specifies the quality setting for recording the video stream, in the 0 ... 100 range (100 = maximum quality). Use -1 to use the default value

4.1.1.1.5.119.4 iConfServerDotNet.StartRecordingCurrentVideoPreview Method (bool, string, int, int, bool, int, int, int, bool)

Record the current video preview (with or without audio) to a wmv file

C#

```
public bool StartRecordingCurrentVideoPreview(
    bool recordAudio,
    string fileName,
    int audioBitRate,
    int numberAudioChannels,
    bool useFixedFrameRate,
    int videoBitRate,
    int maxKeyFrameSpacing,
    int videoQuality,
    bool recordingsIsPriority
);
```

```
        bool recordingIsPriority  
    );
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved
int audioBitRate	the bit rate of the audio stream in bits per second. E.g. a value of 40000 specifies about 40 Kbits per second. Use -1 to use the default value
int numberAudioChannels	Specifies the number of channels of the audio stream. 1 for 1 channel (mono), 2 for 2 channels stereo. Use -1 to use the default value
bool useFixedFrameRate	Specifies if a fixed frame rate must be used to record the video stream
int videoBitRate	Specifies the bit rate of the video stream in bits per second. Use -1 to use the default value
int maxKeyFrameSpacing	Maximum interval between key frames. E.g. 20000 = 1 key frame every 2 seconds. Use -1 to use the default value
int videoQuality	Specifies the quality setting for recording the video stream, in the 0 ... 100 range (100 = maximum quality). Use -1 to use the default value
bool recordingIsPriority	Set to true to gives priority to the recording activity over the preview

4.1.1.1.5.120 StartRecordingCurrentVideoPreviewWmv9 Method

4.1.1.1.5.120.1 iConfServerDotNet.StartRecordingCurrentVideoPreviewWmv9 Method (bool, string, string)

Record the current video preview (with or without audio) to a wmv 9 encoded file using a custom profile

C#

```
public bool StartRecordingCurrentVideoPreviewWmv9(  
    bool recordAudio,  
    string fileName,  
    string prxFilePath  
) ;
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved
string prxFilePath	The path to the .prx file that contains the profile settings to be used

Returns

true if recording successfully started

4.1.1.1.5.120.2 iConfServerDotNet.StartRecordingCurrentVideoPreviewWmv9 Method (bool, string, string, bool)

Record the current video preview (with or without audio) to a wmv 9 encoded file using a custom profile

C#

```
public bool StartRecordingCurrentVideoPreviewWmv9(  
    bool recordAudio,  
    string fileName,  
    string prxFilePath,  
    bool recordingIsPriority  
) ;
```

```
        string prxFilePath,  
        bool recordingIsPriority  
    );
```

Parameters

Parameters	Description
bool recordAudio	true to record audio as well, false otherwise
string fileName	the wmv file to which the recording should be saved
string prxFilePath	The path to the .prx file that contains the profile settings to be used
bool recordingIsPriority	Set to true to gives priority to the recording activity over the preview

Returns

true if recording successfully started

4.1.1.1.5.121 iConfServerDotNet.StartRecordingToWMV Method

Function used to record the activity of a window along with Audio. Can be used to record a conferencing session

C#

```
public bool StartRecordingToWMV(  
    int windowHandle,  
    bool recordAudio,  
    string fileName  
);
```

Parameters

Parameters	Description
int windowHandle	The handle of the window that should be recorded
bool recordAudio	set to true to record audio and video false to only record audio
string fileName	The file to which the video should be saved
videoCompressorIndex	The 0 based index of the video compressor to use
audioCompressorIndex	The 0 based index of the Audio compressor to use
realtimeCompression	set true if the resulting video should be compressed in realtime or false to compress when the recording stops

4.1.1.1.5.122 iConfServerDotNet.StartScreenSharing Method**C#**

```
public void StartScreenSharing(  
    DesktopSharingType sharingType  
);
```

Remarks

This is StartScreenSharing, a member of class iConfServerDotNet.

4.1.1.1.5.123 iConfServerDotNet.StartTransmittingAudio Method

Starts transmitting audio. Only call this function if you have explicitly called the StopTransmittingAudio (see page 85) function

C#

```
public bool StartTransmittingAudio();
```

Returns

true if successful, false otherwise

4.1.1.1.5.124 iConfServerDotNet.StopMotionDetection Method

Stops the motion detection

C#

```
public void StopMotionDetection();
```

4.1.1.1.5.125 iConfServerDotNet.StopPreview Method

Stops the video preview and ends the encoding of the video

C#

```
public bool StopPreview();
```

Returns

True if successful, false if unsuccessful

Remarks

StartPreview (see page 79)

Related Topics

M:iConfServer.NET.iConfServerDotNet.StartPreview(System.Int32)

4.1.1.1.5.126 StopRecording Method**4.1.1.1.5.126.1 iConfServerDotNet.StopRecording Method ()**

Used to stop a recording session started by calling the StartRecording (see page 80) function

C#

```
public void StopRecording();
```

4.1.1.1.5.126.2 iConfServerDotNet.StopRecording Method (int)

Used to stop a recording session started by calling the StartRecording (see page 80) function

C#

```
public void StopRecording(  
    int windowHandle  
);
```

4.1.1.1.5.127 iConfServerDotNet.StopRecordingAudioConference Method

Stops the process of recording the audio conference (process started by calling StartRecordingAudioConference (see page 80))

C#

```
public void StopRecordingAudioConference();
```

4.1.1.1.5.128 iConfServerDotNet.StopRecordingCurrentVideoPreview Method

Used to stop recording the current video preview function

C#

```
public void StopRecordingCurrentVideoPreview();
```


4.1.1.1.5.129 iConfServerDotNet.StopScreenSharing Method

Stops a screen sharing session

C#

```
public bool StopScreenSharing();
```

4.1.1.1.5.130 iConfServerDotNet.StopTransmittingAudio Method

Stops transmitting audio

C#

```
public bool StopTransmittingAudio();
```

Returns

true if successful, false otherwise

4.1.1.1.5.131 TextOverLay Method

4.1.1.1.5.131.1 iConfServerDotNet.TextOverLay Method (bool, string)

Enables/Disables Text Overlay

C#

```
public void TextOverLay(  
    bool enabled,  
    string overlayString  
);
```

Parameters

Parameters	Description
bool enabled	Set to true to enable the Text overlay
string overlayString	The text string that should be overlayed

4.1.1.1.5.131.2 iConfServerDotNet.TextOverLay Method (bool, string, Font, int, bool, int)

Enables/Disables Text Overlay

C#

```
public void TextOverLay(  
    bool enabled,  
    string overlayString,  
    Font font,  
    int fontColor,  
    bool transparent,  
    int backgroundColor  
);
```

Parameters

Parameters	Description
bool enabled	Set to true to enable the Text overlay
string overlayString	The text string that should be overlayed

4.1.1.1.5.132 iConfServerDotNet.UpdateProfilePicture Method

Asynchronous function used to Update the profile picture of the user currently logged into the Central Server.

C#

```
public void UpdateProfilePicture();
```

4.1.1.1.5.133 iConfServerDotNet.ViewUser Method

Send request to central server to view the video feed of a user

C#

```
public void ViewUser(  
    string userName  
) ;
```

Parameters

Parameters	Description
string userName	The user to view

4.1.1.1.5.134 iConfServerDotNet.waveOutGetVolume Method

C#

```
public static extern int waveOutGetVolume(  
    IntPtr hwo,  
    out uint dwVolume  
) ;
```

Remarks

This is waveOutGetVolume, a member of class iConfServerDotNet.

4.1.1.1.5.135 iConfServerDotNet.waveOutSetVolume Method

C#

```
public static extern int waveOutSetVolume(  
    IntPtr hwo,  
    uint dwVolume  
) ;
```

Remarks

This is waveOutSetVolume, a member of class iConfServerDotNet.

4.1.1.1.5.136 iConfServerDotNet.XmitDesktop Method

C#

```
protected bool XmitDesktop() ;
```

Remarks

This is XmitDesktop, a member of class iConfServerDotNet.

4.1.1.1.6 iConfServerDotNet Properties

4.1.1.1.6.1 iConfServerDotNet.CallbackId Property

the Callback Id of the iConf Server.

What is a Callback Id ? A callback id is an identifier that gets used in the case of an incoming call, to determine if the incoming call can be automatically accepted.

C#

```
public string CallbackId;
```

4.1.1.1.6.2 iConfServerDotNet.CallbackObject Property

C#

```
public ICallback CallbackObject;
```

Remarks

This is CallbackObject, a member of class iConfServerDotNet.

4.1.1.1.6.3 iConfServerDotNet.ConnectedClientCount Property

returns the number of clients connected to the iConfServer.NET (see page 25) component in IP to IP mode

C#

```
public Int32 ConnectedClientCount;
```

4.1.1.1.6.4 iConfServerDotNet.CurrentVideoCodec Property

The selected video codec H263 (see page 48) / H264 (see page 48) /MPEG4 (see page 48)

C#

```
public string CurrentVideoCodec;
```

4.1.1.1.6.5 iConfServerDotNet.IsConnectedToCentralServer Property

Checks to see if there is an active connection to the Central Server.

C#

```
public bool IsConnectedToCentralServer;
```

4.1.1.1.6.6 iConfServerDotNet.IsListening Property

C#

```
public bool IsListening;
```

Remarks

This is IsListening, a member of class iConfServerDotNet.

4.1.1.1.6.7 iConfServerDotNet.IsScreenSharingEnabled Property

Returns true if screen sharing is turned on, false otherwise

C#

```
public bool IsScreenSharingEnabled;
```

4.1.1.1.6.8 iConfServerDotNet.PreviewHeight Property

returns the height of the video preview

C#

```
public int PreviewHeight;
```

4.1.1.1.6.9 iConfServerDotNet.PreviewWidth Property

returns the width of the video preview

C#

```
public int PreviewWidth;
```

4.1.1.1.6.10 iConfServerDotNet.VideoDeviceCount Property

returns the number of video devices installed on the system

C#

```
public int VideoDeviceCount;
```

4.1.1.1.6.11 iConfServerDotNet.VideoRateControl Property

C#

```
public RateControl VideoRateControl;
```

Remarks

This is VideoRateControl, a member of class iConfServerDotNet.

4.1.1.1.7 iConfServerDotNet Delegates

4.1.1.1.7.1 iConfServerDotNet.AddContactRequestResponseDelegate Delegate

Delegate for AddContactRequestResponse (see page 100) event

C#

```
public delegate void AddContactRequestResponseDelegate(object sender, bool  
addContactSuccess, string messageFromServer);
```

Parameters

Parameters	Description
sender	
addContactSuccess	true if the contact request was sent successfully, false otherwise
messageFromServer	message returned by the server

4.1.1.1.7.2 iConfServerDotNet.CannotRunGraphDelegate Delegate

Delegate for the CannotRunGraph (see page 100) event

C#

```
public delegate void CannotRunGraphDelegate();
```

4.1.1.1.7.3 iConfServerDotNet.ChangePasswordResponseDelegate Delegate

Delegate for ChangePasswordResponse event

C#

```
public delegate void ChangePasswordResponseDelegate(string response, bool success);
```

Parameters

Parameters	Description
response	response received from central server
success	true if change password request was successful, false otherwise

4.1.1.1.7.4 iConfServerDotNet.ClientConnectionLostDelegate Delegate

Delegate for ClientConnectionLost event

C#

```
public delegate void ClientConnectionLostDelegate(int socketHandle);
```

Parameters

Parameters	Description
socketHandle	the socket handle of the client

4.1.1.1.7.5 iConfServerDotNet.ClientDisconnectedDelegate Delegate

Delegate for ClientDisconnected (see page 100) event

C#

```
public delegate void ClientDisconnectedDelegate(object sender, string ipAddress, int port);
```

Parameters

Parameters	Description
sender	
ipAddress	the IP address of the iConf Client that disconnected from the iConf Server
port	teh port that the iConf client was connected to

4.1.1.1.7.6 iConfServerDotNet.ConferenceMessageReceivedDelegate Delegate

Delegate for ConferenceMessageReceived (see page 100) event

C#

```
public delegate void ConferenceMessageReceivedDelegate(object sender, string messageReceived, string fromUser);
```

Parameters

Parameters	Description
sender	
messageReceived	teh message that was received
fromUser	the user that has sent the message

4.1.1.1.7.7 iConfServerDotNet.ConferencePrivateMessageReceivedDelegate Delegate

Delegate for ConferencePrivateMessageReceived (see page 100) event

C#

```
public delegate void ConferencePrivateMessageReceivedDelegate(string messageReceived, string fromUser);
```

Parameters

Parameters	Description
sender	
messageReceived	teh message that was received
fromUser	the user that has sent the message

4.1.1.1.7.8 iConfServerDotNet.ConferenceRequestDelegate Delegate

Delegate for ConferenceRequest (see page 100) event

C#

```
public delegate void ConferenceRequestDelegate(object sender, string leader, ArrayList otherUsers, string otherUsersStr, string conferenceId);
```

Parameters

Parameters	Description
sender	
leader	the contact who is the conference leader (the one who started/requested the conference)
otherUsers	the other users that were invited to the conference in the form of HashTable
otherUsersStr	the other users that were invited to the conference in the form of
conferenceld	the id of teh conference that needs to be joined (use with JoinConference (see page 61) function)

4.1.1.1.7.9 iConfServerDotNet.ConferenceRequestRejectedDelegate Delegate

Delegate for ConferenceRequestRejected (see page 101) event

C#

```
public delegate void ConferenceRequestRejectedDelegate(string rejectedByUser, string reasonsForRejection);
```

Parameters

Parameters	Description
rejectedByUser	the user that rejected the conference request
reasonsForRejection	reasons, as specified by the user, why the conference was rejected

4.1.1.1.7.10 iConfServerDotNet.ContactListReceivedDelegate Delegate

Delegate for ContactListReceived (see page 101) Event

C#

```
public delegate void ContactListReceivedDelegate(object sender, Hashtable contactListHash, string contactList);
```

Parameters

Parameters	Description
sender	
contactListHash	Hashtable containing contacts contact(key) and status (value)
contactList	comma separated string of contacts and statuses (last 3 characters of contact is the contact status)

4.1.1.1.7.11 iConfServerDotNet.ContactStatusChangeDelegate Delegate

Delegate for ContactStatusChange event

C#

```
public delegate void ContactStatusChangeDelegate(object sender, string contactUserName, string newStatusCode);
```

Parameters

Parameters	Description
sender	
contactUserName	the contact that has changed status
newStatusCode	the new status of the contact (3 characters code)

4.1.1.1.7.12 iConfServerDotNet.ContactStatusMessageChangeDelegate Delegate

Delegate for ContactStatusMessageChanged (see page 101) event

C#

```
public delegate void ContactStatusMessageChangeDelegate(string contactUserName, string newStatusMessage);
```

Parameters

Parameters	Description
contactUserName	
newStatusMessage	

4.1.1.1.7.13 iConfServerDotNet.CustomObjectReceivedDelegate Delegate

Delegate for CustomObjectReceived (see page 101) event

C#

```
public delegate void CustomObjectReceivedDelegate(string fromuser, string customobject);
```

Parameters

Parameters	Description
fromuser	
customobject	

4.1.1.1.7.14 iConfServerDotNet.DisconnectedFromCentralServerDelegate Delegate

Delegate for DisconnectedFromCentralServer (see page 101) event

C#

```
public delegate void DisconnectedFromCentralServerDelegate(object sender);
```

Parameters

Parameters	Description
sender	

4.1.1.1.7.15 iConfServerDotNet.FileDownloadCompletedDelegate Delegate

Delegate for the FileDownloadCompleted (see page 101) event

C#

```
public delegate void FileDownloadCompletedDelegate(string fileId, string originalFileName, string saveFileTo);
```

Parameters

Parameters	Description
fileId	The id of the file which has been downloaded
originalFileName	The name of the file that was shared
saveFileTo	The location and file name where the downloaded file will be saved

4.1.1.1.7.16 iConfServerDotNet.FileDownloadProgressDelegate Delegate

Delegate for the FileDownloadProgress (see page 101) event

C#

```
public delegate void FileDownloadProgressDelegate(
    string fileId, string originalFileName, string saveFileTo, int
    progressPercentage, long bytesReceived,
    long totalBytesToReceive);
```

Parameters

Parameters	Description
fileId	The id of the file that is being downloaded
originalFileName	The name of the file that was shared
saveFileTo	The location and file name where the downloaded file will be saved
progressPercentage	percentage of the download that has been completed (0 to 100)
bytesReceived	The number of bytes received
totalBytesToReceive	The total bytes remaining to be downloaded

4.1.1.1.7.17 iConfServerDotNet.IncomingCallDelegate Delegate

Delegate for IncomingCall (see page 102) event

C#

```
public delegate void IncomingCallDelegate(object sender, string authenticationData, Int32
    socketHandle, string callbackid, string callbackipaddress, int callbackvideoport, int
    callbackaudiotcpport, int callbackaudiudpport);
```

Parameters

Parameters	Description
sender	
authenticationData	the authentication data sent by the iConf Client
socketHandle	the handle of the socket connection which will be used to respond to the iConf Client
callbackid	call back id sent by the iConf Client. Will be used by the iConf Server to know whether the call should be automatically accepted (in the case of a callback)
callbackipaddress	the ip address sent by the peer connection which can be used to call back
callbackvideoport	the video port sent by the peer connection which can be used to call back
callbackaudiotcpport	the audio tcp port sent by the peer connection which can be used to call back
callbackaudiudpport	the audio udp port sent by the peer connection which can be used to call back

4.1.1.1.7.18 iConfServerDotNet.InvalidConferenceRoomPasswordDelegate Delegate

Delegate for InvalidConferenceRoomPassword (see page 102) event

C#

```
public delegate void InvalidConferenceRoomPasswordDelegate();
```

4.1.1.1.7.19 iConfServerDotNet.JoinedConferenceRoomDelegate Delegate

Delegate for JoinedConferenceRoom (see page 102) event

C#

```
public delegate void JoinedConferenceRoomDelegate(string roomName);
```

Parameters

Parameters	Description
roomName	

4.1.1.1.7.20 iConfServerDotNet.LogDelegate Delegate**C#**

```
public delegate void LogDelegate(string logMessage);
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.LogDelegate.

4.1.1.1.7.21 iConfServerDotNet.LoginResponseDelegate Delegate

Delegate for LoginResponse (see page 102) Event

C#

```
public delegate void LoginResponseDelegate(object sender, bool loginSuccess, string messageFromServer);
```

Parameters

Parameters	Description
sender	
loginSuccess	True if login was successful, false otherwise
messageFromServer	message received from server

4.1.1.1.7.22 iConfServerDotNet.LogoutResponseDelegate Delegate

Delegate for LogoutResponse (see page 102) event

C#

```
public delegate void LogoutResponseDelegate(object sender, bool logoutSuccess, string messageFromServer);
```

Parameters

Parameters	Description
sender	
logoutSuccess	true if logout was successful, false otherwise
messageFromServer	message from server

4.1.1.1.7.23 iConfServerDotNet.MotionDetectedDelegate Delegate

Delegate for MotionDetected (see page 102) event

C#

```
public delegate void MotionDetectedDelegate();
```

Parameters

Parameters	Description
BitmapFrame	

4.1.1.1.7.24 iConfServerDotNet.MotionNotDetectedDelegate Delegate

C#

```
public delegate void MotionNotDetectedDelegate();
```

4.1.1.1.7.25 iConfServerDotNet.NewContactRequestDelegate Delegate

Delegate for NewContactRequest (see page 103) event

C#

```
public delegate void NewContactRequestDelegate(object sender, string fromuser);
```

Parameters

Parameters	Description
sender	
fromuser	the user from which the contact request is coming from

4.1.1.1.7.26 iConfServerDotNet.NewDesktopAvailableDelegate Delegate

C#

```
public delegate void NewDesktopAvailableDelegate( int userid );
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.NewDesktopAvailableDelegate.

4.1.1.1.7.27 iConfServerDotNet.NewFileShareRequestDelegate Delegate

Delegate for the NewFileShareRequest (see page 103) event

C#

```
public delegate void NewFileShareRequestDelegate(string fileId, string fromuser, string filename, long fileSize);
```

Parameters

Parameters	Description
fileId	The id of the file that is being shared
fromuser	The user who has shared the file
filename	The name of the file that is being shared
fileSize	The size of the file that is being shared

4.1.1.1.7.28 iConfServerDotNet.NewVideoFrameAvailableDelegate Delegate

Delegate for the NewVideoFrameAvailable (see page 103) event

C#

```
public delegate void NewVideoFrameAvailableDelegate(Bitmap bitmap);
```

Parameters

Parameters	Description
bitmap	a Bitmap object

4.1.1.1.7.29 iConfServerDotNet.NewVideoFrameBytesAvailableDelegate Delegate

Delegate for the NewVideoFrameBytesAvailable (see page 103) event

C#

```
public delegate void NewVideoFrameBytesAvailableDelegate(byte[] bitmapbytes);
```

Parameters

Parameters	Description
bitmap	a Bitmap object

4.1.1.1.7.30 iConfServerDotNet.PendingContactRequestResponseDelegate Delegate

Delegate for PendingContactRequestResponse event

C#

```
public delegate void PendingContactRequestResponseDelegate(object sender, Hashtable contactsHash, string contacts);
```

Parameters

Parameters	Description
sender	
contactsHash	Hashtable containing contacts that have sent contact requests (that have not yet been accepted)
contacts	comma delimited string containing contacts that have sent contact requests (that have not yet been accepted)

4.1.1.1.7.31 iConfServerDotNet.ProfilePictureUpdatedDelegate Delegate

Delegate for ProfilePictureUpdated ([see page 103](#)) event

C#

```
public delegate void ProfilePictureUpdatedDelegate();
```

4.1.1.1.7.32 iConfServerDotNet.ReadyToShareScreenDelegate Delegate

Delegate for ReadyToShareScreen ([see page 103](#)) event

C#

```
public delegate void ReadyToShareScreenDelegate();
```

4.1.1.1.7.33 iConfServerDotNet.RecordingReadyToStartDelegate Delegate**C#**

```
public delegate void RecordingReadyToStartDelegate();
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.RecordingReadyToStartDelegate.

4.1.1.1.7.34 iConfServerDotNet.RegistrationResponseDelegate Delegate

Delegate for RegistrationResponse ([see page 104](#)) Event

C#

```
public delegate void RegistrationResponseDelegate(object sender, bool registrationSuccess, string messageFromServer);
```

Parameters

Parameters	Description
sender	

registrationSuccess	true if the registration of the new user succeeded, false otherwise
messageFromServer	message returned by the server

4.1.1.1.7.35 iConfServerDotNet.RemoveContactResponseDelegate Delegate

Delegate for RemoveContactResponse (see page 104) event

C#

```
public delegate void RemoveContactResponseDelegate(object sender, string messageFromServer, bool removalSuccess, string userToRemove);
```

Parameters

Parameters	Description
sender	
messageFromServer	message returned by the server
removalSuccess	true if the contact was successfully removed, false otherwise
userToRemove	contact that has been removed

4.1.1.1.7.36 iConfServerDotNet.ScreenControlDeniedDelegate Delegate

Delegate for ScreenControlDenied (see page 104) event

C#

```
public delegate void ScreenControlDeniedDelegate(string fromUser);
```

Parameters

Parameters	Description
fromUser	The name of the user that denied the request

4.1.1.1.7.37 iConfServerDotNet.ScreenControlGrantedDelegate Delegate

Delegate for ScreenControlGranted (see page 104) event

C#

```
public delegate void ScreenControlGrantedDelegate(string fromUser);
```

Parameters

Parameters	Description
fromUser	The name of the user that granted the request

4.1.1.1.7.38 iConfServerDotNet.ScreenControlRequestDelegate Delegate

Delegate for ScreenControlRequest (see page 104) event

C#

```
public delegate void ScreenControlRequestDelegate(string fromUser);
```

Parameters

Parameters	Description
fromUser	when in IP to IP mode fromUser will contain the socket handle of the client that made the request when in Central Server mode it will contain the name of the user that made the request

4.1.1.1.7.39 iConfServerDotNet.ScreenSharingSessionAcceptedDelegate Delegate

Delegate for ScreenSharingSessionAccepted (see page 104) event

C#

```
public delegate void ScreenSharingSessionAcceptedDelegate(Int32 socketHandle);
```

Parameters

Parameters	Description
socketHandle	the socket handle of the iconfclient.net that accepted the request

4.1.1.1.7.40 iConfServerDotNet.ScreenSharingSessionAvailableDelegate Delegate

Delegate for the ScreenSharingSessionAvailable (see page 104) event

C#

```
public delegate void ScreenSharingSessionAvailableDelegate(string fromUser, string keyToConnect);
```

Parameters

Parameters	Description
fromUser	The user requesting the sharing session
keyToConnect	The key that will be used to connect to the session

4.1.1.1.7.41 iConfServerDotNet.ScreenSharingSessionRejectedDelegate Delegate

Delegate for ScreenSharingSessionRejected (see page 105) event

C#

```
public delegate void ScreenSharingSessionRejectedDelegate(Int32 socketHandle, string rejectionMessage);
```

Parameters

Parameters	Description
socketHandle	the socket handle of the iconfclient.net that rejected the request
rejectionMessage	a message from the iconfclient.net

4.1.1.1.7.42 iConfServerDotNet.SearchResponseDelegate Delegate

C#

```
public delegate void SearchResponseDelegate(Hashtable results, string resultstr);
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.SearchResponseDelegate.

4.1.1.1.7.43 iConfServerDotNet.StatusMessagesReceivedDelegate Delegate

C#

```
public delegate void StatusMessagesReceivedDelegate(Hashtable statusMessageListHash, string customstring);
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.StatusMessagesReceivedDelegate.

4.1.1.1.7.44 iConfServerDotNet.TalkingDelegate Delegate

Delegate for Talking (see page 105) event

C#

```
public delegate void TalkingDelegate(string userName, bool talking);
```

Parameters

Parameters	Description
userName	userName of the user that is talking
talking	true when the user is talking, false when the user stops talking

4.1.1.1.7.45 iConfServerDotNet.TextMessageReceivedDelegate Delegate

Delegate for the TextMessageReceived (see page 105) event

C#

```
public delegate void TextMessageReceivedDelegate(string messageReceived);
```

Parameters

Parameters	Description
messageReceived	The message received from the iConf Client

4.1.1.1.7.46 iConfServerDotNet.UserJoinedConferenceDelegate Delegate

Delegate for UserJoinedConference (see page 105) event

C#

```
public delegate void UserJoinedConferenceDelegate(object sender, string userName);
```

Parameters

Parameters	Description
sender	
userName	the name of the user that is joining the conference that the logged in user is participating in

4.1.1.1.7.47 iConfServerDotNet.UserLeftConferenceDelegate Delegate

Delegate for UserLeftConference (see page 105) event

C#

```
public delegate void UserLeftConferenceDelegate(object sender, string userName);
```

Parameters

Parameters	Description
sender	
userName	teh name of the user that has left the conference that the logged in user is participating in

4.1.1.1.7.48 iConfServerDotNet.UserProfileReceivedDelegate Delegate

C#

```
public delegate void UserProfileReceivedDelegate(Hashtable results);
```

Remarks

This is nested type iConfServer.NET.iConfServerDotNet.UserProfileReceivedDelegate.

4.1.1.1.7.49 iConfServerDotNet.VideoDeviceSelectedDelegate Delegate

Delegate for VideoDeviceSelected (see page 106) event

C#

```
public delegate void VideoDeviceSelectedDelegate(string deviceName);
```

Parameters

Parameters	Description
deviceName	The name of the device that was selected

4.1.1.1.7.50 iConfServerDotNet.VideoPreviewStartedDelegate Delegate

Delegate for VideoPreviewStarted (see page 106) event

C#

```
public delegate void VideoPreviewStartedDelegate(int videoWidth, int videoHeight, string deviceName);
```

Parameters

Parameters	Description
videoWidth	The width of the video preview
videoHeight	The height of the video preview

4.1.1.1.7.51 iConfServerDotNet.ViewUserResponseReceivedDelegate Delegate

Delegate for ViewUserResponse event

C#

```
public delegate void ViewUserResponseReceivedDelegate(object sender, string userName, int port, int videoWidth, int videoHeight, string videoCodec);
```

Parameters

Parameters	Description
sender	
userName	the user that the request to view was for
port	the port to connect to to view the user (used in Call function)
videoWidth	the video width of the user's video stream
videoHeight	the video height of the user's video stream
videoCodec	the codec used by the user to encode the streaming video

4.1.1.1.8 iConfServerDotNet Events**4.1.1.1.8.1 iConfServerDotNet.AddContactRequestResponse Event**

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.AddUserAsContact Method (see page 49)

C#

```
public event AddContactRequestResponseDelegate AddContactRequestResponse;
```

Description

This event is fired in response to a call to the iConfServerDotNet.AddUserAsContact Method ([see page 49](#))

4.1.1.1.8.2 iConfServerDotNet.CannotRunGraph Event

Fired when the DirectX Video preview graph fails to run. Usually due to a failure to communicate with the video capture device (device might be in use)

C#

```
public event CannotRunGraphDelegate CannotRunGraph;
```

4.1.1.1.8.3 iConfServerDotNet.ChangePasswordResponse Event

fired when a response is received from the central server after a request to change password was made

C#

```
public event ChangePasswordResponseDelegate ChangePasswordResponse;
```

4.1.1.1.8.4 iConfServerDotNet.ClientConnectionLost Event

Fired when an iConf Client disconnects from the iConf Central Server Service use this event when you want to capture the handle of the client that has disconnected

C#

```
public event ClientConnectionLostDelegate ClientConnectionLost;
```

4.1.1.1.8.5 iConfServerDotNet.ClientDisconnected Event

Fired when an iConf Client disconnects from the iConf Central Server Service

C#

```
public event ClientDisconnectedDelegate ClientDisconnected;
```

4.1.1.1.8.6 iConfServerDotNet.ConferenceMessageReceived Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.SendMessageToConference Method ([see page 70](#))

C#

```
public event ConferenceMessageReceivedDelegate ConferenceMessageReceived;
```

4.1.1.1.8.7 iConfServerDotNet.ConferencePrivateMessageReceived Event

Fired when a private message is received from another user (see iConfServerDotNet.SendPrivatemessageToUser Method ([see page 70](#)))

C#

```
public event ConferencePrivateMessageReceivedDelegate ConferencePrivateMessageReceived;
```

4.1.1.1.8.8 iConfServerDotNet.ConferenceRequest Event

Fired when a new conference request is received from a user

C#

```
public event ConferenceRequestDelegate ConferenceRequest;
```

4.1.1.1.8.9 iConfServerDotNet.ConferenceRequestRejected Event

Fired when a conference request has been rejected

C#

```
public event ConferenceRequestRejectedDelegate ConferenceRequestRejected;
```

4.1.1.1.8.10 iConfServerDotNet.ContactListReceived Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetContactList Method (see page 55)

C#

```
public event ContactListReceivedDelegate ContactListReceived;
```

4.1.1.1.8.11 iConfServerDotNet.ContactStatusChanged Event

Fired when a user in the logged in user's contact list changes status

C#

```
public event ContactStatusChangeDelegate ContactStatusChanged;
```

4.1.1.1.8.12 iConfServerDotNet.ContactStatusMessageChanged Event

Fired when a user changes his/her status message

C#

```
public event ContactStatusMessageChangeDelegate ContactStatusMessageChanged;
```

4.1.1.1.8.13 iConfServerDotNet.CustomObjectReceived Event

Fired when a custom object is received from another user

C#

```
public event CustomObjectReceivedDelegate CustomObjectReceived;
```

4.1.1.1.8.14 iConfServerDotNet.DisconnectedFromCentralServer Event

Fired when an iConf Server disconnects from the iConf Central Server Service

C#

```
public event DisconnectedFromCentralServerDelegate DisconnectedFromCentralServer;
```

4.1.1.1.8.15 iConfServerDotNet.FileDownloadCompleted Event

Fired when a file download completes successfully

C#

```
public event FileDownloadCompletedDelegate FileDownloadCompleted;
```

4.1.1.1.8.16 iConfServerDotNet.FileDownloadProgress Event

Use this event to report on the progress of a file download

C#

```
public event FileDownloadProgressDelegate FileDownloadProgress;
```

4.1.1.1.8.17 iConfServerDotNet.IncomingCall Event

Fired when a Call Request is received from an iConf Client

C#

```
public event IncomingCallDelegate IncomingCall;
```

4.1.1.1.8.18 iConfServerDotNet.InvalidConferenceRoomPassword Event

Fired when an attempt to join a password protected conference room fails due to an invalid password

C#

```
public event InvalidConferenceRoomPasswordDelegate InvalidConferenceRoomPassword;
```

4.1.1.1.8.19 iConfServerDotNet.JoinedConferenceRoom Event

C#

```
public event JoinedConferenceRoomDelegate JoinedConferenceRoom;
```

Remarks

This is JoinedConferenceRoom, a member of class iConfServerDotNet.

4.1.1.1.8.20 iConfServerDotNet.Log Event

C#

```
public event LogDelegate Log;
```

Remarks

This is Log, a member of class iConfServerDotNet.

4.1.1.1.8.21 iConfServerDotNet.LoginResponse Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.Login Method (see page 63)

C#

```
public event LoginResponseDelegate LoginResponse;
```

4.1.1.1.8.22 iConfServerDotNet.LogoutResponse Event

Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.LogoutResponse Method (see page 64)

C#

```
public event LogoutResponseDelegate LogoutResponse;
```

4.1.1.1.8.23 iConfServerDotNet.MotionDetected Event

Fired when motion is detected in the video preview

C#

```
public event MotionDetectedDelegate MotionDetected;
```

4.1.1.1.8.24 iConfServerDotNet.MotionNotDetected Event

fired when motion is not detected in the video preview

C#

```
public event MotionNotDetectedDelegate MotionNotDetected;
```

4.1.1.1.8.25 iConfServerDotNet.NewContactRequest Event

Fired when a new contact request is received from another user

C#

```
public event NewContactRequestDelegate NewContactRequest;
```

4.1.1.1.8.26 iConfServerDotNet.NewDesktopAvailable Event

C#

```
public event NewDesktopAvailableDelegate NewDesktopAvailable;
```

Remarks

This is NewDesktopAvailable, a member of class iConfServerDotNet.

4.1.1.1.8.27 iConfServerDotNet.NewFileShareRequest Event

Fired when a file share request is received from a user (a user wants to share a file with you)

C#

```
public event NewFileShareRequestDelegate NewFileShareRequest;
```

4.1.1.1.8.28 iConfServerDotNet.NewVideoFrameAvailable Event

Fired when a new Video frame is available (Bitmap) for processing

C#

```
public event NewVideoFrameAvailableDelegate NewVideoFrameAvailable;
```

4.1.1.1.8.29 iConfServerDotNet.NewVideoFrameBytesAvailable Event

Fired when a new Video frame is available (byte[]) for processing

C#

```
public event NewVideoFrameBytesAvailableDelegate NewVideoFrameBytesAvailable;
```

4.1.1.1.8.30 iConfServerDotNet.PendingContactRequest Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.GetPendingContactRequests Method ([see page 56](#))

C#

```
public event PendingContactRequestResponseDelegate PendingContactRequest;
```

4.1.1.1.8.31 iConfServerDotNet.ProfilePictureUpdated Event

Fired when the logged in user's profile picture gets successfully updated, following a call to the iConfServerDotNet.UpdateProfilePicture Method ([see page 86](#))

C#

```
public event ProfilePictureUpdatedDelegate ProfilePictureUpdated;
```

4.1.1.1.8.32 iConfServerDotNet.ReadyToShareScreen Event

Fired when a screen is ready to be shared following a call to StartScreenSharing ([see page 83](#))

C#

```
public event ReadyToShareScreenDelegate ReadyToShareScreen;
```

4.1.1.1.8.33 iConfServerDotNet.RecordingReadyToStart Event

C#

```
public event RecordingReadyToStartDelegate RecordingReadyToStart;
```

Remarks

This is RecordingReadyToStart, a member of class iConfServerDotNet.

4.1.1.1.8.34 iConfServerDotNet.RegistrationResponse Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RegisterUser Method (see page 65)

C#

```
public event RegistrationResponseDelegate RegistrationResponse;
```

4.1.1.1.8.35 iConfServerDotNet.RemoveContactResponse Event

Fired when a response is received from the iConf Central Server Service in response to a call to the iConfServerDotNet.RemoveContact Method (see page 67)

C#

```
public event RemoveContactResponseDelegate RemoveContactResponse;
```

4.1.1.1.8.36 iConfServerDotNet.ScreenControlDenied Event

fired when a request to control the screen has been denied (central server mode)

C#

```
public event ScreenControlDeniedDelegate ScreenControlDenied;
```

4.1.1.1.8.37 iConfServerDotNet.ScreenControlGranted Event

fired when a request to control the screen has been granted (central server mode)

C#

```
public event ScreenControlGrantedDelegate ScreenControlGranted;
```

4.1.1.1.8.38 iConfServerDotNet.ScreenControlRequest Event

fired when a request to control the screen is received while in a screen sharing session

C#

```
public event ScreenControlRequestDelegate ScreenControlRequest;
```

4.1.1.1.8.39 iConfServerDotNet.ScreenSharingSessionAccepted Event

Fired when an iconfclient.net has accepted a screen sharing session request

C#

```
public event ScreenSharingSessionAcceptedDelegate ScreenSharingSessionAccepted;
```

4.1.1.1.8.40 iConfServerDotNet.ScreenSharingSessionAvailable Event

Fired when a screen sharing session becomes available

C#

```
public event ScreenSharingSessionAvailableDelegate ScreenSharingSessionAvailable;
```

4.1.1.1.8.41 iConfServerDotNet.ScreenSharingSessionRejected Event

Fired when an iconfclient.net has rejected a screen sharing session request

C#

```
public event ScreenSharingSessionRejectedDelegate ScreenSharingSessionRejected;
```

4.1.1.1.8.42 iConfServerDotNet.SearchResponse Event

C#

```
public event SearchResponseDelegate SearchResponse;
```

Remarks

This is SearchResponse, a member of class iConfServerDotNet.

4.1.1.1.8.43 iConfServerDotNet.StatusMessagesReceived Event

C#

```
public event StatusMessagesReceivedDelegate StatusMessagesReceived;
```

Remarks

This is StatusMessagesReceived, a member of class iConfServerDotNet.

4.1.1.1.8.44 iConfServerDotNet.Talking Event

Fired when a user starts talking - This event will only work properly when Voice Activation Detection is enabled (see EnableVoiceActivation (see page 52) function).

C#

```
public event TalkingDelegate Talking;
```

4.1.1.1.8.45 iConfServerDotNet.TextMessageReceived Event

Fired when a text message is received by from an iConf Client

C#

```
public event TextMessageReceivedDelegate TextMessageReceived;
```

4.1.1.1.8.46 iConfServerDotNet.UserJoinedConference Event

Fired when a user joins a conference that the current logged in user is participating in.

C#

```
public event UserJoinedConferenceDelegate UserJoinedConference;
```

4.1.1.1.8.47 iConfServerDotNet.UserLeftConference Event

Fired when a user leaves a conference that the current logged in user is participating in.

C#

```
public event UserLeftConferenceDelegate UserLeftConference;
```

4.1.1.1.8.48 iConfServerDotNet.UserProfileReceived Event

C#

```
public event UserProfileReceivedDelegate UserProfileReceived;
```

Remarks

This is UserProfileReceived, a member of class iConfServerDotNet.

4.1.1.1.8.49 iConfServerDotNet.VideoDeviceSelected Event

Fired when a new video device has been selected after calling SelectVideoDevice (see page 69) - After this event is fired, new video sizes might be available and if you are displaying a list video sizes, you should reload that list

C#

```
public event VideoDeviceSelectedDelegate VideoDeviceSelected;
```

4.1.1.1.8.50 iConfServerDotNet.VideoPreviewStarted Event

C#

```
public event VideoPreviewStartedDelegate VideoPreviewStarted;
```

Remarks

This is VideoPreviewStarted, a member of class iConfServerDotNet.

4.1.1.1.8.51 iConfServerDotNet.ViewUserResponseReceived Event

Fired when a response is received from the iConf Central Server Service in response to a call to the IComEvents.ViewUserResponseReceived Method ([see page 86](#))


C#

```
public event ViewUserResponseReceivedDelegate ViewUserResponseReceived;
```

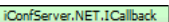
4.1.2 Interfaces

The following table lists interfaces in this documentation.

Interfaces

	Name	Description
	ICallback (see page 106)	This is class iConfServer.NET.ICallback.

4.1.2.1 ICallback Interface

Inheritance Hierarchy**C#**

```
public interface ICallback;
```







File

iConfServerDotNet.cs

Remarks

This is class iConfServer.NET.ICallback.

Members**ICallback Methods**

	Name	Description
	ClientConnectionLost (see page 108)	This is ClientConnectionLost, a member of class ICallback.
	ConferenceRequest (see page 108)	This is ConferenceRequest, a member of class ICallback.
	ContactListReceived (see page 108)	This is ContactListReceived, a member of class ICallback.
	ContactStatusChanged (see page 108)	This is ContactStatusChanged, a member of class ICallback.
	IncomingCall (see page 108)	This is IncomingCall, a member of class ICallback.
	JoinedConferenceRoom (see page 109)	This is JoinedConferenceRoom, a member of class ICallback.

◆	LoginResponse (see page 109)	This is LoginResponse, a member of class ICallback.
◆	LogoutResponse (see page 109)	This is LogoutResponse, a member of class ICallback.
◆	NewContactRequest (see page 109)	This is NewContactRequest, a member of class ICallback.
◆	PendingContactRequestResponse (see page 109)	This is PendingContactRequestResponse, a member of class ICallback.
◆	RegistrationResponse (see page 109)	This is RegistrationResponse, a member of class ICallback.
◆	SearchResponse (see page 110)	This is SearchResponse, a member of class ICallback.
◆	TextMessageReceived (see page 110)	This is TextMessageReceived, a member of class ICallback.
◆	UserJoinedConference (see page 110)	This is UserJoinedConference, a member of class ICallback.
◆	UserLeftConference (see page 110)	This is UserLeftConference, a member of class ICallback.
◆	VideoDeviceSelected (see page 110)	This is VideoDeviceSelected, a member of class ICallback.
◆	VideoPreviewStarted (see page 111)	This is VideoPreviewStarted, a member of class ICallback.
◆	ViewUserResponseReceived (see page 111)	This is ViewUserResponseReceived, a member of class ICallback.

ICallback Methods

	Name	Description
◆	ClientConnectionLost (see page 108)	This is ClientConnectionLost, a member of class ICallback.
◆	ConferenceRequest (see page 108)	This is ConferenceRequest, a member of class ICallback.
◆	ContactListReceived (see page 108)	This is ContactListReceived, a member of class ICallback.
◆	ContactStatusChanged (see page 108)	This is ContactStatusChanged, a member of class ICallback.
◆	IncomingCall (see page 108)	This is IncomingCall, a member of class ICallback.
◆	JoinedConferenceRoom (see page 109)	This is JoinedConferenceRoom, a member of class ICallback.
◆	LoginResponse (see page 109)	This is LoginResponse, a member of class ICallback.
◆	LogoutResponse (see page 109)	This is LogoutResponse, a member of class ICallback.
◆	NewContactRequest (see page 109)	This is NewContactRequest, a member of class ICallback.
◆	PendingContactRequestResponse (see page 109)	This is PendingContactRequestResponse, a member of class ICallback.
◆	RegistrationResponse (see page 109)	This is RegistrationResponse, a member of class ICallback.
◆	SearchResponse (see page 110)	This is SearchResponse, a member of class ICallback.
◆	TextMessageReceived (see page 110)	This is TextMessageReceived, a member of class ICallback.
◆	UserJoinedConference (see page 110)	This is UserJoinedConference, a member of class ICallback.
◆	UserLeftConference (see page 110)	This is UserLeftConference, a member of class ICallback.
◆	VideoDeviceSelected (see page 110)	This is VideoDeviceSelected, a member of class ICallback.
◆	VideoPreviewStarted (see page 111)	This is VideoPreviewStarted, a member of class ICallback.
◆	ViewUserResponseReceived (see page 111)	This is ViewUserResponseReceived, a member of class ICallback.

4.1.2.1.1 ICallback Methods

4.1.2.1.1.1 ICallback.ClientConnectionLost Method

C#

```
void ClientConnectionLost(
    Int32 socketHandle
);
```

Remarks

This is ClientConnectionLost, a member of class ICallback.

4.1.2.1.1.2 ICallback.ConferenceRequest Method

C#

```
void ConferenceRequest(  
    string leader,  
    string conferenceId,  
    string otherusers  
);
```

Remarks

This is ConferenceRequest, a member of class ICallback.

4.1.2.1.1.3 ICallback.ContactListReceived Method

C#

```
void ContactListReceived(  
    string contactList  
);
```

Remarks

This is ContactListReceived, a member of class ICallback.

4.1.2.1.1.4 ICallback.ContactStatusChanged Method

C#

```
void ContactStatusChanged(  
    string contactUserName,  
    string newStatusCode  
);
```

Remarks

This is ContactStatusChanged, a member of class ICallback.

4.1.2.1.1.5 ICallback.IncomingCall Method

C#

```
void IncomingCall(  
    string authenticationData,  
    Int32 socketHandle,  
    string callbackid,  
    string callbackipaddress,  
    int callbackvideoport,  
    int callbackaudiocpport,  
    int callbackaudiudpport  
);
```

Remarks

This is IncomingCall, a member of class ICallback.

4.1.2.1.1.6 ICallback.JoinedConferenceRoom Method

C#

```
void JoinedConferenceRoom(  
    string roomName  
);
```

Remarks

This is JoinedConferenceRoom, a member of class ICallback.

4.1.2.1.1.7 ICallback.LoginResponse Method

C#

```
void LoginResponse(  
    string loginSuccess,  
    string messageFromServer  
);
```

Remarks

This is LoginResponse, a member of class ICallback.

4.1.2.1.1.8 ICallback.LogoutResponse Method

C#

```
void LogoutResponse(  
    string logoutSuccess,  
    string messageFromServer  
);
```

Remarks

This is LogoutResponse, a member of class ICallback.

4.1.2.1.1.9 ICallback.NewContactRequest Method

C#

```
void NewContactRequest(  
    string fromuser  
);
```

Remarks

This is NewContactRequest, a member of class ICallback.

4.1.2.1.1.10 ICallback.PendingContactRequestResponse Method

C#

```
void PendingContactRequestResponse(  
    string contacts  
);
```

Remarks

This is PendingContactRequestResponse, a member of class ICallback.

4.1.2.1.1.11 ICallback.RegistrationResponse Method

C#

```
void RegistrationResponse(  
    string registrationSuccess,  
    string messageFromServer  
);
```

Remarks

This is RegistrationResponse, a member of class ICallback.

4.1.2.1.1.12 ICallback.SearchResponse Method

C#

```
void SearchResponse(  

```

```
        string res  
    );
```

Remarks

This is SearchResponse, a member of class ICallback.

4.1.2.1.1.13 ICallback.TextMessageReceived Method**C#**

```
void TextMessageReceived(  
    string messageReceived  
);
```

Remarks

This is TextMessageReceived, a member of class ICallback.

4.1.2.1.1.14 ICallback.UserJoinedConference Method**C#**

```
void UserJoinedConference(  
    string userName  
);
```

Remarks

This is UserJoinedConference, a member of class ICallback.

4.1.2.1.1.15 ICallback.UserLeftConference Method**C#**

```
void UserLeftConference(  
    string userName  
);
```

Remarks

This is UserLeftConference, a member of class ICallback.

4.1.2.1.1.16 ICallback.VideoDeviceSelected Method**C#**

```
void VideoDeviceSelected(  
    string deviceName  
);
```

Remarks

This is VideoDeviceSelected, a member of class ICallback.

4.1.2.1.1.17 ICallback.VideoPreviewStarted Method**C#**

```
void VideoPreviewStarted(  
    Int32 videoWidth,  
    Int32 videoHeight,  
    string deviceName  
);
```

Remarks

This is VideoPreviewStarted, a member of class ICallback.

4.1.2.1.1.18 ICallback.ViewUserResponseReceived Method

C#

```
void ViewUserResponseReceived(  
    string userName,  
    Int32 port,  
    Int32 videoWidth,  
    Int32 videoHeight,  
    string videoCodec  
) ;
```

Remarks

This is ViewUserResponseReceived, a member of class ICallback.




4.2 iConfClient.NET Namespace

This is namespace iConfClient.NET.

Classes

	Name	Description
	iConfClientDotNet (see page 112)	Implementing interface that will be visible from JS

Interfaces

	Name	Description
	ICallback (see page 132)	This is class iConfClient.NET.ICallback.
	IComEvents (see page 133)	Defines events that will be raised from the associated COM object. Don't derive from this interface. Instead, mark any class that uses this interface with the attribute [ComSourceInterfaces(typeof(IComEvents))]. Any class that uses this interface should implement a public event called MyFirstEvent using a delegate that returns void and accepts a single string parameter called args. e.g. public delegate void MyFirstEventHandler(string args);
	IObjectSafety (see page 134)	This is the only Guid that cannot be modified in this file

4.2.1 Classes

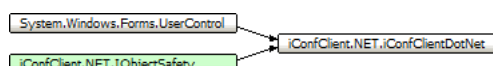
The following table lists classes in this documentation.

Classes

	Name	Description
	iConfClientDotNet (see page 112)	Implementing interface that will be visible from JS

4.2.1.1 iConfClientDotNet Class

Inheritance Hierarchy



C#

```
public class iConfClientDotNet : System.Windows.Forms.UserControl, IObjectSafety;
```

File

iConfClientDotNet.cs

Remarks







Implementing interface that will be visible from JS

Members**IObjectSafety Methods**

	Name	Description
◆	GetInterfaceSafetyOptions (see page 135)	This is GetInterfaceSafetyOptions, a member of class IObjectSafety.
◆	SetInterfaceSafetyOptions (see page 135)	This is SetInterfaceSafetyOptions, a member of class IObjectSafety.


iConfClientDotNet Class

	Name	Description
◆	AcceptScreenSharingSession (see page 119)	Used to Accept a screen sharing session deprecated
◆	Call (see page 119)	Function used to place a call to an iConf Server or get a video stream from a Central Server
◆	CaptureImage (see page 120)	Used to capture a still image from the incoming video stream to a file
◆	ClearImage (see page 121)	clears video feed receiving area
◆	Disconnect (see page 121)	Use this function to disconnects from an iConf Server
◆	Dispose (see page 121)	Clean up any resources being used.
◆	EnableEchoCancellation (see page 121)	Enable/Disable echo cancellation Note this function is currently experimental
◆	GetAssemblyVersion (see page 121)	returns the version of the assembly
◆	GetAudioOutputDevices (see page 121)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method
◆	GetAudioOutputDevicesList (see page 122)	Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method
◆	GetInterfaceSafetyOptions (see page 122)	COM support
◆	GetUserScr (see page 122)	This is GetUserScr, a member of class iConfClientDotNet.
◆	Logging (see page 122)	enables / disables logging log file is in temp path iconflog.txt
◆	Mute (see page 122)	Mutes/Un-Mute incoming audio playback
◆	ReconfigureAudioSystem (see page 123)	Use this function to reconfigure the audio system (choose different audio input/output device)
◆	RegisterClass (see page 123)	Register the class as a control and set it's CodeBase entry
◆	RejectScreenSharingSession (see page 123)	Used to reject a screen sharing session
◆	RequestScreenControl (see page 123)	Used to request control while in a screen sharing session
◆	SendTextMessage (see page 124)	Used to send a text message to the iConfServer.NET (see page 25) that the client is connected to.
◆	SetAudioPlaybackType (see page 124)	deprecated
◆	SetImage (see page 124)	Sets an image that should be displayed in the video feed receiving area

	SetInterfaceSafetyOptions (see page 125)	COM Support
	SetOutputGain (see page 125)	This is SetOutputGain, a member of class iConfClientDotNet.
	SetPlaybackDevice (see page 125)	Sets the device to be used when playing back incoming audio
	SetViewerSizeMode (see page 125)	Sets the Video Viewer size Mode The Default is StretchImage. refer to for the meaning of the possible values http://msdn.microsoft.com/en-us/library/system.windows.forms.pictureboxSizeMode.aspx Please note that when working at higher resolutions, it is better to pick AutoSize. The scaling provided by StretchImage is slow when working with higher resolutions.
	SetVolume (see page 126)	Used to set the Volume of an Audio conference
	UnregisterClass (see page 126)	Called to unregister the control



Methods

iConfClientDotNet Class

	Name	Description
	iConfClientDotNet (see page 118)	Initializes a new instance of the iConfClientDotNet class




iConfClientDotNet Enumerations

iConfClientDotNet Class

	Name	Description
	AudioPlayBackType (see page 118)	0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.
	audioType (see page 119)	enum representing the type of system that should be used for audio Windows Audio System or DirectSound





iConfClientDotNet Fields

iConfClientDotNet Class

	Name	Description
	H263 (see page 119)	Constant representing the H263 codec selection. Can be used as a replacement for typing the string "H263"
	H264 (see page 119)	Constant representing the H264 codec selection. Can be used as a replacement for typing the string "H24"
	MPEG4 (see page 119)	Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"

iConfClientDotNet Properties

iConfClientDotNet Class












	Name	Description
	AutoReconnect (see page 126)	Auto reconnect client upon disconnection.
	CallbackObject (see page 126)	This is CallbackObject, a member of class iConfClientDotNet.
	CurrentVideoHeight (see page 126)	Height of the video frames currently being displayed
	CurrentVideoWidth (see page 126)	Width of the video frames currently being displayed




iConfClientDotNet Delegates

iConfClientDotNet Class

Name	Description
CallAcceptedDelegate (see page 127)	Delegate for CallAccepted (see page 129) Event
CallRejectedDelegate (see page 127)	Delegate for CallRejected (see page 129) Event
ClientDisconnectedDelegate (see page 127)	Delegate for ClientDisconnected (see page 130) event.
IncomingVideoSizeChangedDelegate (see page 127)	Delegate for IncomingVideoSizeChanged (see page 130) Event
LogDelegate (see page 128)	This is nested type iConfClient.NET.iConfClientDotNet.LogDelegate.
NewBitmapFrameAvailableDelegate (see page 128)	Delegate for the NewBitmapFrameAvailable (see page 130) event
NewDesktopAvailableDelegate (see page 128)	This is nested type iConfClient.NET.iConfClientDotNet.NewDesktopAvailableDelegate.
NewVideoFrameBytesAvailableDelegate (see page 128)	Delegate for the NewVideoFrameBytesAvailable (see page 130) event
RequestToShareScreenDelegate (see page 128)	Delegate for RequestToShareScreen (see page 130) Event
ScreenControlDeniedDelegate (see page 128)	Delegate for the ScreenControlDenied (see page 130) event
ScreenControlGrantedDelegate (see page 129)	Delegate for the ScreenControlGranted (see page 131) event
ScreenSharingStoppedDelegate (see page 129)	This is nested type iConfClient.NET.iConfClientDotNet.ScreenSharingStoppedDelegate.
SocketErrorDelegate (see page 129)	Delegate for the SocketError (see page 131) event
TextMessageReceivedDelegate (see page 129)	Delegate for TextMessageReceived (see page 131) event


iConfClientDotNet Events**iConfClientDotNet Class**

	Name	Description
	CallAccepted (see page 129)	Fired when an outgoing call has been accepted by an iConfServer.NET (see page 25)
	CallRejected (see page 129)	Fired when an outgoing call has been rejected by an iConfServer.NET (see page 25)
	ClientDisconnected (see page 130)	This event is fired when an iConf Client disconnects from an iConf Server
	IncomingVideoSizeChanged (see page 130)	Fired when the Incoming Video Size changes
	Log (see page 130)	This is Log, a member of class iConfClientDotNet.
	NewBitmapFrameAvailable (see page 130)	Fired when a new Bitmap frame is available for viewing processing
	NewDesktopAvailable (see page 130)	This is NewDesktopAvailable, a member of class iConfClientDotNet.
	NewVideoFrameBytesAvailable (see page 130)	Fired when a new Video frame is available (byte[]) for processing
	RequestToShareScreen (see page 130)	Fired when a screen sharing request has been received (an iConfServer.NET (see page 25) is sharing its screen)
	ScreenControlDenied (see page 130)	Fired when screen control has been denied by an iConfServer.NET (see page 25)
	ScreenControlGranted (see page 131)	Fired when screen control has been granted by an iConfServer.NET (see page 25)

	ScreenSharingStopped (see page 131)	This is ScreenSharingStopped, a member of class iConfClientDotNet.
	SocketError (see page 131)	This event is fired when a socket error occurs
	TextMessageReceived (see page 131)	Fired when a Text Message is received from an iConfServer.NET (see page 25) Event fired when a Text message is received from an iConf Server

Methods

iConfClientDotNet Class

	Name	Description
	iConfClientDotNet (see page 118)	Initializes a new instance of the iConfClientDotNet class



iConfClientDotNet Delegates

iConfClientDotNet Class















Name	Description
CallAcceptedDelegate (see page 127)	Delegate for CallAccepted (see page 129) Event
CallRejectedDelegate (see page 127)	Delegate for CallRejected (see page 129) Event
ClientDisconnectedDelegate (see page 127)	Delegate for ClientDisconnected (see page 130) event.
IncomingVideoSizeChangedDelegate (see page 127)	Delegate for IncomingVideoSizeChanged (see page 130) Event
LogDelegate (see page 128)	This is nested type iConfClient.NET.iConfClientDotNet.LogDelegate.
NewBitmapFrameAvailableDelegate (see page 128)	Delegate for the NewBitmapFrameAvailable (see page 130) event
NewDesktopAvailableDelegate (see page 128)	This is nested type iConfClient.NET.iConfClientDotNet.NewDesktopAvailableDelegate.
NewVideoFrameBytesAvailableDelegate (see page 128)	Delegate for the NewVideoFrameBytesAvailable (see page 130) event
RequestToShareScreenDelegate (see page 128)	Delegate for RequestToShareScreen (see page 130) Event
ScreenControlDeniedDelegate (see page 128)	Delegate for the ScreenControlDenied (see page 130) event
ScreenControlGrantedDelegate (see page 129)	Delegate for the ScreenControlGranted (see page 131) event
ScreenSharingStoppedDelegate (see page 129)	This is nested type iConfClient.NET.iConfClientDotNet.ScreenSharingStoppedDelegate.
SocketErrorDelegate (see page 129)	Delegate for the SocketError (see page 131) event
TextMessageReceivedDelegate (see page 129)	Delegate for TextMessageReceived (see page 131) event

iConfClientDotNet Enumerations




iConfClientDotNet Class

	Name	Description
	AudioPlayBackType (see page 118)	0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.
	audioType (see page 119)	enum representing the type of system that should be used for audio Windows Audio System or DirectSound



iConfClientDotNet Events**iConfClientDotNet Class**

	Name	Description
	CallAccepted (see page 129)	Fired when an outgoing call has been accepted by an iConfServer.NET (see page 25)
	CallRejected (see page 129)	Fired when an outgoing call has been rejected by an iConfServer.NET (see page 25)
	ClientDisconnected (see page 130)	This event is fired when an iConf Client disconnects from an iConf Server
	IncomingVideoSizeChanged (see page 130)	Fired when the Incoming Video Size changes
	Log (see page 130)	This is Log, a member of class iConfClientDotNet.
	NewBitmapFrameAvailable (see page 130)	Fired when a new Bitmap frame is available for viewing processing
	NewDesktopAvailable (see page 130)	This is NewDesktopAvailable, a member of class iConfClientDotNet.
	NewVideoFrameBytesAvailable (see page 130)	Fired when a new Video frame is available (byte[]) for processing
	RequestToShareScreen (see page 130)	Fired when a screen sharing request has been received (an iConfServer.NET (see page 25) is sharing its screen)
	ScreenControlDenied (see page 130)	Fired when screen control has been denied by an iConfServer.NET (see page 25)
	ScreenControlGranted (see page 131)	Fired when screen control has been granted by an iConfServer.NET (see page 25)
	ScreenSharingStopped (see page 131)	This is ScreenSharingStopped, a member of class iConfClientDotNet.
	SocketError (see page 131)	This event is fired when a socket error occurs
	TextMessageReceived (see page 131)	Fired when a Text Message is received from an iConfServer.NET (see page 25) Event fired when a Text message is received from an iConf Server


iConfClientDotNet Fields**iConfClientDotNet Class**






























	Name	Description
	H263 (see page 119)	Constant representing the H263 codec selection. Can be used as a replacement for typing the string "H263"
	H264 (see page 119)	Constant representing the H264 codec selection. Can be used as a replacement for typing the string "H24"
	MPEG4 (see page 119)	Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"

IObjctSafety Methods




	Name	Description
	GetInterfaceSafetyOptions (see page 135)	This is GetInterfaceSafetyOptions, a member of class IObjctSafety.
	SetInterfaceSafetyOptions (see page 135)	This is SetInterfaceSafetyOptions, a member of class IObjctSafety.


iConfClientDotNet Class

	Name	Description
	AcceptScreenSharingSession (see page 119)	Used to Accept a screen sharing session deprecated

	Call (see page 119)	Function used to place a call to an iConf Server or get a video stream from a Central Server
	CaptureImage (see page 120)	Used to capture a still image from the incoming video stream to a file
	ClearImage (see page 121)	clears video feed receiving area
	Disconnect (see page 121)	Use this function to disconnects from an iConf Server
 	Dispose (see page 121)	Clean up any resources being used.
	EnableEchoCancellation (see page 121)	Enable/Disable echo cancellation Note this function is currently experimental
	GetAssemblyVersion (see page 121)	returns the version of the assembly
	GetAudioOutputDevices (see page 121)	Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method
	GetAudioOutputDevicesList (see page 122)	Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method
	GetInterfaceSafetyOptions (see page 122)	COM support
	GetUserScr (see page 122)	This is GetUserScr, a member of class iConfClientDotNet.
	Logging (see page 122)	enables / disables logging log file is in temp path iconflog.txt
	Mute (see page 122)	Mutes/Un-Mute incoming audio playback
	ReconfigureAudioSystem (see page 123)	Use this function to reconfigure the audio system (choose different audio input/output device)
 	RegisterClass (see page 123)	Register the class as a control and set it's CodeBase entry
	RejectScreenSharingSession (see page 123)	Used to reject a screen sharing session
	RequestScreenControl (see page 123)	Used to request control while in a screen sharing session
	SendTextMessage (see page 124)	Used to send a text message to the iConfServer.NET (see page 25) that the client is connected to.
	SetAudioPlaybackType (see page 124)	deprecated
	SetImage (see page 124)	Sets an image that should be displayed in the video feed receiving area
	SetInterfaceSafetyOptions (see page 125)	COM Support
	SetOutputGain (see page 125)	This is SetOutputGain, a member of class iConfClientDotNet.
	SetPlaybackDevice (see page 125)	Sets the device to be used when playing back incoming audio
	SetViewerSizeMode (see page 125)	Sets the Video Viewer size Mode The Default is StretchImage. refer to for the meaning of the possible values http://msdn.microsoft.com/en-us/library/system.windows.forms.pictureboxsizemode.aspx Please note that when working at higher resolutions, it is better to pick AutoSize. The scaling provided by StretchImage is slow when working with higher resolutions.
	SetVolume (see page 126)	Used to set the Volume of an Audio conference
 	UnregisterClass (see page 126)	Called to unregister the control

iConfClientDotNet Properties**iConfClientDotNet Class**

	Name	Description
	AutoReconnect (see page 126)	Auto reconnect client upon disconnection.
	CallbackObject (see page 126)	This is CallbackObject, a member of class iConfClientDotNet.
	CurrentVideoHeight (see page 126)	Height of the video frames currently being displayed

	CurrentVideoWidth (see page 126)	Width of the video frames currently being displayed
---	--	---

4.2.1.1.1 iConfClientDotNet.iConfClientDotNet Constructor

Initializes a new instance of the iConfClientDotNet class

C#

```
public iConfClientDotNet();
```

4.2.1.1.2 iConfClientDotNet Enumerations

4.2.1.1.2.1 iConfClient.NET.iConfClientDotNet.AudioPlayBackType Enumeration

0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.

C#

```
public enum AudioPlayBackType {
    STABLE = 0,
    LOWLATENCY = 1,
    VARIABLE = 2,
    JITTER = 3
}
```

File

iConfClientDotNet.cs

4.2.1.1.2.2 iConfClient.NET.iConfClientDotNet.audioType Enumeration

enum representing the type of system that should be used for audio Windows Audio System or DirectSound

C#

```
public enum audioType {
    WindowsAudio = 1,
    DirectSound = 2
}
```

File

iConfClientDotNet.cs

4.2.1.1.3 iConfClientDotNet Fields

4.2.1.1.3.1 iConfClientDotNet.H263 Field

C#

```
public const string H263 = "H263";
```

Remarks

Constant representing the H263 codec selection.

Can be used as a replacement for typing the string "H263"

4.2.1.1.3.2 iConfClientDotNet.H264 Field

C#

```
public const string H264 = "H264";
```

Remarks

Constant representing the H264 codec selection.

Can be used as a replacement for typing the string "H24"

4.2.1.1.3.3 iConfClientDotNet.MPEG4 Field

Constant representing the MPEG4 codec selection. Can be used as a replacement for typing the string "MPEG4"

C#

```
public const string MPEG4 = "MPEG4";
```

4.2.1.1.4 iConfClientDotNet Methods

4.2.1.1.4.1 iConfClientDotNet.AcceptScreenSharingSession Method

Used to Accept a screen sharing session deprecated

C#

```
public void AcceptScreenSharingSession(  
    string connectionString  
);
```

4.2.1.1.4.2 iConfClientDotNet.Call Method

Function used to place a call to an iConf Server or get a video stream from a Central Server

C#

```
public void Call(  
    string ipAddress,  
    int videoPort,  
    int videoWidth,  
    int videoHeight,  
    string authenticationData,  
    string callbackid,  
    string callbackip,  
    int callbackvideoport,  
    int callbackaudiotcpport,  
    int callbackaudioudpport,  
    string videoCodec  
);
```

Parameters

Parameters	Description
string ipAddress	The Ip address where the iConf Server is listening for incoming calls
int videoPort	The tcp port on which the iConf Server is listening for incoming calls
int videoWidth	the anticipated width of the incoming video stream - Note : should be set to 0 in Peer to Peer mode
int videoHeight	the anticipated height of the incoming video stream - Note : should be set to 0 in Peer to Peer mode
string authenticationData	The data that should be sent to the iConf Server for authentication

string callbackid	The callback id that should be provided to the iConf Server upon authentication. If the callbackid matches the iConf server's callbackid, the call will automatically be accepted by the iConf Server
string callbackip	The ip address that should be used by the peer to when making the callback call
int callbackvideoport	The video port that should be used by the peer to when making the callback call
int callbackaudiotcpport	The audio tcp port that should be used by the peer to when making the callback call
int callbackaudioudport	The audio udp port that should be used by the peer to when making the callback call
string videoCodec	The Video codec which was used to encode the incoming video stream - Note : should be set to empty string in Peer to Peer mode, because in Peer to Peer mode, the video codec is automatically detected on the receiving end.

4.2.1.1.4.3 iConfClientDotNet.CaptureImage Method

Used to capture a still image from the incoming video stream to a file

C#

```
public void CaptureImage(
    string fileName
);
```

Parameters

Parameters	Description
toFileName	The path to the file that the image should be saved to Accepted extensions are .bmp (bitmap) .jpg and .jpeg (JPEG)

4.2.1.1.4.4 iConfClientDotNet.ClearImage Method

clears video feed receiving area

C#

```
public void ClearImage();
```

Parameters

Parameters	Description
img	The Image to set

Example

Can be used upon disconnecting from an iConf Server (set a 'Not connected' image for example)

4.2.1.1.4.5 iConfClientDotNet.Disconnect Method

Use this function to disconnects from an iConf Server

C#

```
public void Disconnect();
```

4.2.1.1.4.6 iConfClientDotNet.Dispose Method

Clean up any resources being used.

C#

```
protected override void Dispose(
```

```
        bool disposing
    );
```

4.2.1.1.4.7 iConfClientDotNet.EnableEchoCancellation Method

Enable/Disable echo cancellation Note this function is currently experimental

C#

```
public bool EnableEchoCancellation(
    bool enabled
);
```

Parameters

Parameters	Description
bool enabled	set to true to enable echo cancellation

4.2.1.1.4.8 iConfClientDotNet.GetAssemblyVersion Method

returns the version of the assembly

C#

```
public string GetAssemblyVersion();
```

4.2.1.1.4.9 iConfClientDotNet.GetAudioOutputDevices Method

Used to retrieve a list of Audio Input devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioInputDevices method

C#

```
public Hashtable GetAudioOutputDevices();
```

Returns

Hashtable

4.2.1.1.4.10 iConfClientDotNet.GetAudioOutputDevicesList Method

Used to retrieve a list of Audio Output devices installed on the system Each device has a friendly name and a device Id both are stored in the return value of the GetAudioOutputDevicesList method

C#

```
public List<KeyValuePair<int, string>> GetAudioOutputDevicesList();
```

Returns

List

4.2.1.1.4.11 iConfClientDotNet.GetInterfaceSafetyOptions Method

COM support

C#

```
public int GetInterfaceSafetyOptions(
    ref Guid riid,
    out int pdwSupportedOptions,
    out int pdwEnabledOptions
);
```

Parameters

Parameters	Description
ref Guid riid	

out int pdwSupportedOptions	
out int pdwEnabledOptions	

4.2.1.1.4.12 iConfClientDotNet.GetUserScr Method

C#

```
public Bitmap GetUserScr(  
    string sessionId  
);
```

Remarks

This is GetUserScr, a member of class iConfClientDotNet.

4.2.1.1.4.13 iConfClientDotNet.Logging Method

enables / disables logging log file is in temp path iconflog.txt

C#

```
public void Logging(  
    bool enabled  
);
```

Parameters

Parameters	Description
bool enabled	true enabled / false disabled

4.2.1.1.4.14 iConfClientDotNet.Mute Method

Mutes/Un-Mute incoming audio playback

C#

```
public bool Mute(  
    bool enabled  
);
```

Parameters

Parameters	Description
bool enabled	true to mute, false to un mute

Returns

true if successful, false otherwise

4.2.1.1.4.15 iConfClientDotNet.ReconfigureAudioSystem Method

Use this function to reconfigure the audio system (choose different audio input/output device)

C#

```
public bool ReconfigureAudioSystem(  
    audioType audioType,  
    int outputDevice  
);
```

Parameters

Parameters	Description
audioType audioType	1 for Windows Sound System or 2 for DirectSound (Directsound recommended)
int outputDevice	the device id of the audio playback device

4.2.1.1.4.16 iConfClientDotNet.RegisterClass Method

Register the class as a control and set it's CodeBase entry

C#

```
public static void RegisterClass(  
    string key  
) ;
```

Parameters

Parameters	Description
string key	The registry key of the control

4.2.1.1.4.17 iConfClientDotNet.RejectScreenSharingSession Method

Used to reject a screen sharing session

C#

```
public void RejectScreenSharingSession(  
    string rejectionMessage  
) ;
```

Parameters

Parameters	Description
string rejectionMessage	the message that should be sent to the originator of the request

4.2.1.1.4.18 iConfClientDotNet.RequestScreenControl Method

Used to request control while in a screen sharing session

C#

```
public void RequestScreenControl ( ) ;
```

Parameters

Parameters	Description
toUser	The user to whom the request should go to

4.2.1.1.4.19 iConfClientDotNet.SendTextMessage Method

Used to send a text message to the iConfServer.NET ([see page 25](#)) that the client is connected to.

C#

```
public void SendTextMessage(  
    string messageToSend  
) ;
```

Parameters

Parameters	Description
string messageToSend	The text message that should be sent

4.2.1.1.4.20 iConfClientDotNet.SetAudioPlaybackType Method

deprecated

C#

```
public bool SetAudioPlaybackType(  
    int playbackType  
) ;
```

```
);
```

Parameters

Parameters	Description
int playbackType	0 = STABLE allows playback to buffer audio and only starts skipping packets if the buffer holds more than 10 packets. 1 = LOWLATENCY tries to keep the buffer below 2 packets after which it will start skipping packets. 2 = VARIABLE changes the playback sample rate to ensure low latency. 3 = JITTER uses a jitter buffer for keeping low latency.

Remarks

see AudioPlayBackType (see page 118) enum

4.2.1.1.4.21 iConfClientDotNet.SetImage Method

Sets an image that should be displayed in the video feed receiving area

C#

```
public void SetImage(  
    Image img  
);
```

Parameters

Parameters	Description
Image img	The Image to set

Example

Can be used upon disconnecting from an iConf Server (set a 'Not connected' image for example)

4.2.1.1.4.22 iConfClientDotNet.SetInterfaceSafetyOptions Method

COM Support

C#

```
public int SetInterfaceSafetyOptions(  
    ref Guid riid,  
    int dwOptionSetMask,  
    int dwEnabledOptions  
);
```

Parameters

Parameters	Description
ref Guid riid	
int dwOptionSetMask	
int dwEnabledOptions	

4.2.1.1.4.23 iConfClientDotNet.SetOutputGain Method

C#

```
public void SetOutputGain(  
    Int32 value  
);
```

Remarks

This is SetOutputGain, a member of class iConfClientDotNet.

4.2.1.1.4.24 iConfClientDotNet.SetPlaybackDevice Method

Sets the device to be used when playing back incoming audio

C#

```
public void SetPlaybackDevice(  
    int deviceId  
) ;
```

Parameters

Parameters	Description
int deviceId	the id of the audio output device

4.2.1.1.4.25 iConfClientDotNet.SetViewerSizeMode Method

Sets the Video Viewer size Mode The Default is StretchImage. refer to for the meaning of the possible values <http://msdn.microsoft.com/en-us/library/system.windows.forms.pictureboxsizemode.aspx> Please note that when working at higher resolutions, it is better to pick AutoSize. The scaling provided by StretchImage is slow when working with higher resolutions.

C#

```
public void SetViewerSizeMode(  
    PictureBoxSizeMode sizeMode  
) ;
```

Parameters

Parameters	Description
PictureBoxSizeMode sizeMode	

4.2.1.1.4.26 iConfClientDotNet.SetVolume Method

Used to set the Volume of an Audio conference

C#

```
public bool SetVolume(  
    int volumeLevel  
) ;
```

Parameters

Parameters	Description
int volumeLevel	a value between 0 (min) and 255 (max)

Returns

true if successful, false otherwise

4.2.1.1.4.27 iConfClientDotNet.UnregisterClass Method

Called to unregister the control

C#

```
public static void UnregisterClass(  
    string key  
) ;
```

Parameters

Parameters	Description
string key	The registry key

4.2.1.1.5 iConfClientDotNet Properties

4.2.1.1.5.1 iConfClientDotNet.AutoReconnect Property

Auto reconnect client upon disconnection.

C#

```
public bool AutoReconnect;
```

4.2.1.1.5.2 iConfClientDotNet.CallbackObject Property

C#

```
public ICallback CallbackObject;
```

Remarks

This is CallbackObject, a member of class iConfClientDotNet.

4.2.1.1.5.3 iConfClientDotNet.CurrentVideoHeight Property

Height of the video frames currently being displayed

C#

```
public int CurrentVideoHeight;
```

4.2.1.1.5.4 iConfClientDotNet.CurrentVideoWidth Property

Width of the video frames currently being displayed

C#

```
public int CurrentVideoWidth;
```

4.2.1.1.6 iConfClientDotNet Delegates

4.2.1.1.6.1 iConfClientDotNet.CallAcceptedDelegate Delegate

Delegate for CallAccepted ([see page 129](#)) Event

C#

```
public delegate void CallAcceptedDelegate( string acceptMessage );
```

Parameters

Parameters	Description
acceptMessage	contains the (optional) message sent by the iConfServer.NET (see page 25)

4.2.1.1.6.2 iConfClientDotNet.CallRejectedDelegate Delegate

Delegate for CallRejected ([see page 129](#)) Event

C#

```
public delegate void CallRejectedDelegate( string RejectionMessage );
```

Parameters

Parameters	Description
RejectionMessage	contains the (optional) message sent by the iConfServer.NET (see page 25)

4.2.1.1.6.3 iConfClientDotNet.ClientDisconnectedDelegate Delegate

Delegate for ClientDisconnected (see page 130) event.

C#

```
public delegate void ClientDisconnectedDelegate(object sender, string ipAddress, int port);
```

Parameters

Parameters	Description
sender	The object that triggered the event.
ipAddress	The ip address of the iConf Server that the iConf Client was connected to
port	The port that the iConf Client was connected to

4.2.1.1.6.4 iConfClientDotNet.IncomingVideoSizeChangedDelegate Delegate

Delegate for IncomingVideoSizeChanged (see page 130) Event

C#

```
public delegate void IncomingVideoSizeChangedDelegate(int newWidth, int newHeight);
```

Parameters

Parameters	Description
newWidth	The new incoming width of the encoded video frames
newHeight	The new incoming height of the encoded video frames

4.2.1.1.6.5 iConfClientDotNet.LogDelegate Delegate

C#

```
public delegate void LogDelegate(string logMessage);
```

Remarks

This is nested type iConfClient.NET.iConfClientDotNet.LogDelegate.

4.2.1.1.6.6 iConfClientDotNet.NewBitmapFrameAvailableDelegate Delegate

Delegate for the NewBitmapFrameAvailable (see page 130) event

C#

```
public delegate void NewBitmapFrameAvailableDelegate(Bitmap bmp);
```

Parameters

Parameters	Description
bitmapbytes	an array of byte containing a bitmap object

4.2.1.1.6.7 iConfClientDotNet.NewDesktopAvailableDelegate Delegate

C#

```
public delegate void NewDesktopAvailableDelegate(int sessionid);
```

Remarks

This is nested type iConfClient.NET.iConfClientDotNet.NewDesktopAvailableDelegate.

4.2.1.1.6.8 iConfClientDotNet.NewVideoFrameBytesAvailableDelegate Delegate

Delegate for the NewVideoFrameBytesAvailable (see page 130) event

C#

```
public delegate void NewVideoFrameBytesAvailableDelegate(byte[] bitmapbytes);
```

Parameters

Parameters	Description
bitmapbytes	an array of byte containing a bitmap object

4.2.1.1.6.9 iConfClientDotNet.RequestToShareScreenDelegate Delegate

Delegate for RequestToShareScreen (see page 130) Event

C#

```
public delegate void RequestToShareScreenDelegate(string fromIp, string keytoConnect, string invitationMessage);
```

Parameters

Parameters	Description
fromIp	The Ip Address that the request is coming from
keytoConnect	The key that should be used to connect and view the screen
invitationMessage	contains the (optional) message sent by the iConfServer.NET (see page 25)

4.2.1.1.6.10 iConfClientDotNet.ScreenControlDeniedDelegate Delegate

Delegate for the ScreenControlDenied (see page 130) event

C#

```
public delegate void ScreenControlDeniedDelegate();
```

4.2.1.1.6.11 iConfClientDotNet.ScreenControlGrantedDelegate Delegate

Delegate for the ScreenControlGranted (see page 131) event

C#

```
public delegate void ScreenControlGrantedDelegate();
```

4.2.1.1.6.12 iConfClientDotNet.ScreenSharingStoppedDelegate Delegate

C#

```
public delegate void ScreenSharingStoppedDelegate(int sessionId);
```

Remarks

This is nested type iConfClient.NET.iConfClientDotNet.ScreenSharingStoppedDelegate.

4.2.1.1.6.13 iConfClientDotNet.SocketErrorDelegate Delegate

Delegate for the SocketError (see page 131) event

C#

```
public delegate void SocketErrorDelegate(Exception ex);
```

Parameters

Parameters	Description
ex	The exception that was raised as part of the error

4.2.1.1.6.14 iConfClientDotNet.TextMessageReceivedDelegate Delegate

Delegate for TextMessageReceived (see page 131) event

C#

```
public delegate void TextMessageReceivedDelegate( string messageReceived );
```

Parameters

Parameters	Description
messageReceived	The message that has been received

4.2.1.1.7 iConfClientDotNet Events**4.2.1.1.7.1 iConfClientDotNet.CallAccepted Event**

Fired when an outgoing call has been accepted by an iConfServer.NET (see page 25)

C#

```
public event CallAcceptedDelegate CallAccepted;
```

4.2.1.1.7.2 iConfClientDotNet.CallRejected Event

Fired when an outgoing call has been rejected by an iConfServer.NET (see page 25)

C#

```
public event CallRejectedDelegate CallRejected;
```

4.2.1.1.7.3 iConfClientDotNet.ClientDisconnected Event

This event is fired when an iConf Client disconnects from an iConf Server

C#

```
public event ClientDisconnectedDelegate ClientDisconnected;
```

4.2.1.1.7.4 iConfClientDotNet.IncomingVideoSizeChanged Event

Fired when the Incoming Video Size changes

C#

```
public event IncomingVideoSizeChangedDelegate IncomingVideoSizeChanged;
```

4.2.1.1.7.5 iConfClientDotNet.Log Event**C#**

```
public event LogDelegate Log;
```

Remarks

This is Log, a member of class iConfClientDotNet.

4.2.1.1.7.6 iConfClientDotNet.NewBitmapFrameAvailable Event

Fired when a new Bitmap frame is available for viewing processing

C#

```
public event NewBitmapFrameAvailableDelegate NewBitmapFrameAvailable;
```

4.2.1.1.7.7 iConfClientDotNet.NewDesktopAvailable Event

C#

```
public event NewDesktopAvailableDelegate NewDesktopAvailable;
```

Remarks

This is NewDesktopAvailable, a member of class iConfClientDotNet.

4.2.1.1.7.8 iConfClientDotNet.NewVideoFrameBytesAvailable Event

Fired when a new Video frame is available (byte[]) for processing

C#

```
public event NewVideoFrameBytesAvailableDelegate NewVideoFrameBytesAvailable;
```

4.2.1.1.7.9 iConfClientDotNet.RequestToShareScreen Event

Fired when a screen sharing request has been received (an iConfServer.NET (see page 25) is sharing its screen)

C#

```
public event RequestToShareScreenDelegate RequestToShareScreen;
```

4.2.1.1.7.10 iConfClientDotNet.ScreenControlDenied Event

Fired when screen control has been denied by an iConfServer.NET (see page 25)

C#

```
public event ScreenControlDeniedDelegate ScreenControlDenied;
```

4.2.1.1.7.11 iConfClientDotNet.ScreenControlGranted Event

Fired when screen control has been granted by an iConfServer.NET (see page 25)

C#

```
public event ScreenControlGrantedDelegate ScreenControlGranted;
```

4.2.1.1.7.12 iConfClientDotNet.ScreenSharingStopped Event

C#

```
public event ScreenSharingStoppedDelegate ScreenSharingStopped;
```

Remarks

This is ScreenSharingStopped, a member of class iConfClientDotNet.

4.2.1.1.7.13 iConfClientDotNet.SocketError Event

This event is fired when a socket error occurs

C#

```
public event SocketErrorDelegate SocketError;
```

4.2.1.1.7.14 iConfClientDotNet.TextMessageReceived Event

Fired when a Text Message is received from an iConfServer.NET (see page 25)

Event fired when a Text message is received from an iConf Server

C#

```
public event TextMessageReceivedDelegate TextMessageReceived;
```

Returns

Add a return value description here...




Description

Event fired when a Text message is received from an iConf Server

4.2.2 Interfaces

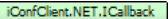
The following table lists interfaces in this documentation.

Interfaces

	Name	Description
	ICallback (see page 132)	This is class iConfClient.NET.ICallback.
	IComEvents (see page 133)	Defines events that will be raised from the associated COM object. Don't derive from this interface. Instead, mark any class that uses this interface with the attribute [ComSourceInterfaces(typeof(IComEvents))]. Any class that uses this interface should implement a public event called MyFirstEvent using a delegate that returns void and accepts a single string parameter called args. e.g. public delegate void MyFirstEventHandler(string args);
	IObjectSafety (see page 134)	This is the only Guid that cannot be modified in this file

4.2.2.1 ICallback Interface

Inheritance Hierarchy

 iConfClient.NET.ICallback

C#

```
public interface ICallback;
```





File

iConfClientDotNet.cs





Remarks

This is class iConfClient.NET.ICallback.

Members**ICallback Methods**

	Name	Description
	CallAccepted (see page 132)	This is CallAccepted, a member of class ICallback.
	CallRejected (see page 132)	This is CallRejected, a member of class ICallback.
	ClientDisconnected (see page 133)	This is ClientDisconnected, a member of class ICallback.
	TextMessageReceived (see page 133)	This is TextMessageReceived, a member of class ICallback.

ICallback Methods

	Name	Description
	CallAccepted (see page 132)	This is CallAccepted, a member of class ICallback.
	CallRejected (see page 132)	This is CallRejected, a member of class ICallback.
	ClientDisconnected (see page 133)	This is ClientDisconnected, a member of class ICallback.
	TextMessageReceived (see page 133)	This is TextMessageReceived, a member of class ICallback.

4.2.2.1.1 ICallback Methods

4.2.2.1.1.1 ICallback.CallAccepted Method

C#

```
void CallAccepted(  
    string acceptMessage  
) ;
```

Remarks

This is CallAccepted, a member of class ICallback.

4.2.2.1.1.2 ICallback.CallRejected Method

C#

```
void CallRejected(  
    string RejectionMessage  
) ;
```

Remarks

This is CallRejected, a member of class ICallback.

4.2.2.1.1.3 ICallback.ClientDisconnected Method

C#

```
void ClientDisconnected(  
    string ipAddress,  
    Int32 port  
) ;
```

Remarks

This is ClientDisconnected, a member of class ICallback.

4.2.2.1.1.4 ICallback.TextMessageReceived Method

C#

```
void TextMessageReceived(  
    string messageReceived  
) ;
```

Remarks

This is TextMessageReceived, a member of class ICallback.

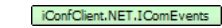
4.2.2.2 IComEvents Interface

Defines events that will be raised from the associated COM object. Don't derive from this interface. Instead, mark any class

that uses this interface with the attribute [ComSourceInterfaces(typeof(ComEvents))]. Any class that uses this interface should implement a public event called MyFirstEvent using a delegate that returns void and accepts a single string parameter called args. e.g.

```
public delegate void MyFirstEventHandler(string args);
```

Inheritance Hierarchy



C#

```
public interface IComEvents;
```

File

iConfClientDotNet.cs

Members

IComEvents Methods

	Name	Description
	CallAccepted (see page 134)	events here
	CallRejected (see page 134)	This is CallRejected, a member of class IComEvents.

IComEvents Methods

	Name	Description
	CallAccepted (see page 134)	events here
	CallRejected (see page 134)	This is CallRejected, a member of class IComEvents.

4.2.2.2.1 IComEvents Methods

4.2.2.2.1.1 IComEvents.CallAccepted Method

C#

```
void CallAccepted(  
    string acceptMessage  
);
```

Remarks

events here

4.2.2.2.1.2 IComEvents.CallRejected Method

C#

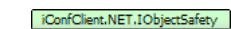
```
void CallRejected(  
    string RejectionMessage  
);
```

Remarks

This is CallRejected, a member of class IComEvents.

4.2.2.3 IObjectSafety Interface

Inheritance Hierarchy



C#

public interface IObjectSafety;

File



iConfClientDotNet.cs

Remarks



This is the only Guid that cannot be modified in this file

Members

IObjectSafety Methods

	Name	Description
	GetInterfaceSafetyOptions (see page 135)	This is GetInterfaceSafetyOptions, a member of class IObjectSafety.
	SetInterfaceSafetyOptions (see page 135)	This is SetInterfaceSafetyOptions, a member of class IObjectSafety.

IObjectSafety Methods

	Name	Description
	GetInterfaceSafetyOptions (see page 135)	This is GetInterfaceSafetyOptions, a member of class IObjectSafety.
	SetInterfaceSafetyOptions (see page 135)	This is SetInterfaceSafetyOptions, a member of class IObjectSafety.

4.2.2.3.1 IObjectSafety Methods

4.2.2.3.1.1 IObjectSafety.GetInterfaceSafetyOptions Method

C#

int GetInterfaceSafetyOptions(
 ref Guid riid,
 out int pdwSupportedOptions,
 out int pdwEnabledOptions
);

Remarks

This is GetInterfaceSafetyOptions, a member of class IObjectSafety.

4.2.2.3.1.2 IObjectSafety.SetInterfaceSafetyOptions Method

C#

int SetInterfaceSafetyOptions(
 ref Guid riid,
 int dwOptionSetMask,
 int dwEnabledOptions
);

Remarks

This is SetInterfaceSafetyOptions, a member of class IObjectSafety.

Index

A

Accepting a conference request 21
 Accepting/Rejecting a contact request 19
 Adding a new Contact 18

B

BASELINE enumeration member 47

C

Choosing a Database 8
 Configure Listening Ports 13
 ConstantBitRate enumeration member 47
 ConstantQuality enumeration member 47

D

Database Scripts 10
 DirectSound enumeration member 46, 119

F

Full enumeration member 46

G

Getting Started 5

H

HIGH enumeration member 47
 HIGH_SPEED enumeration member 47
 HIGH10 enumeration member 47
 HIGH422 enumeration member 47
 HIGH444 enumeration member 47

I

ICallback interface 106, 132
 about ICallback interface 106, 132
 CallAccepted 132
 CallRejected 132

ClientConnectionLost 108
 ClientDisconnected 133
 ConferenceRequest 108
 ContactListReceived 108
 ContactStatusChanged 108
 IncomingCall 108
 JoinedConferenceRoom 109
 LoginResponse 109
 LogoutResponse 109
 NewContactRequest 109
 PendingContactRequestResponse 109
 RegistrationResponse 109
 SearchResponse 110
 TextMessageReceived 110, 133
 UserJoinedConference 110
 UserLeftConference 110
 VideoDeviceSelected 110
 VideoPreviewStarted 111
 ViewUserResponseReceived 111
 IComEvents interface 133
 about IComEvents interface 133
 CallAccepted 134
 CallRejected 134
 iConfClient.NET 111
 iConfClient.NET namespace 111
 Classes 111
 Interfaces 131
 iConfClient.NET.ICallback 132
 iConfClient.NET.ICallback.CallAccepted 132
 iConfClient.NET.ICallback.CallRejected 132
 iConfClient.NET.ICallback.ClientDisconnected 133
 iConfClient.NET.ICallback.TextMessageReceived 133
 iConfClient.NET.IComEvents 133
 iConfClient.NET.IComEvents.CallAccepted 134
 iConfClient.NET.IComEvents.CallRejected 134
 iConfClient.NET.iConfClientDotNet 112
 iConfClient.NET.iConfClientDotNet.AcceptScreenSharingSession 119
 iConfClient.NET.iConfClientDotNet.AudioPlayBackType 118
 iConfClient.NET.iConfClientDotNet.AudioPlayBackType enumeration 118

iConfClient.NET.iConfClientDotNet.audioType 119	iConfClient.NET.iConfClientDotNet.NewBitmapFrameAvailable 130
iConfClient.NET.iConfClientDotNet.audioType enumeration 119	iConfClient.NET.iConfClientDotNet.NewBitmapFrameAvailableDelegate 128
iConfClient.NET.iConfClientDotNet.AutoReconnect 126	iConfClient.NET.iConfClientDotNet.NewDesktopAvailable 130
iConfClient.NET.iConfClientDotNet.Call 119	iConfClient.NET.iConfClientDotNet.NewDesktopAvailableDelegate 128
iConfClient.NET.iConfClientDotNet.CallAccepted 129	iConfClient.NET.iConfClientDotNet.NewVideoFrameBytesAvailable 130
iConfClient.NET.iConfClientDotNet.CallAcceptedDelegate 127	iConfClient.NET.iConfClientDotNet.NewVideoFrameBytesAvailableDelegate 128
iConfClient.NET.iConfClientDotNet.CallbackObject 126	iConfClient.NET.iConfClientDotNet.ReconfigureAudioSystem 123
iConfClient.NET.iConfClientDotNet.CallRejected 129	iConfClient.NET.iConfClientDotNet.RegisterClass 123
iConfClient.NET.iConfClientDotNet.CallRejectedDelegate 127	iConfClient.NET.iConfClientDotNet.RejectScreenSharingSession 123
iConfClient.NET.iConfClientDotNet.CaptureImage 120	iConfClient.NET.iConfClientDotNet.RequestScreenControl 123
iConfClient.NET.iConfClientDotNet.ClearImage 121	iConfClient.NET.iConfClientDotNet.RequestToShareScreen 130
iConfClient.NET.iConfClientDotNet.ClientDisconnected 130	iConfClient.NET.iConfClientDotNet.RequestToShareScreenDelegate 128
iConfClient.NET.iConfClientDotNet.ClientDisconnectedDelegate 127	iConfClient.NET.iConfClientDotNet.ScreenControlDenied 130
iConfClient.NET.iConfClientDotNet.CurrentVideoHeight 126	iConfClient.NET.iConfClientDotNet.ScreenControlDeniedDelegate 128
iConfClient.NET.iConfClientDotNet.CurrentVideoWidth 126	iConfClient.NET.iConfClientDotNet.ScreenControlGranted 131
iConfClient.NET.iConfClientDotNet.Disconnect 121	iConfClient.NET.iConfClientDotNet.ScreenControlGrantedDelegate 129
iConfClient.NET.iConfClientDotNet.Dispose 121	iConfClient.NET.iConfClientDotNet.ScreenSharingStopped 131
iConfClient.NET.iConfClientDotNet.EnableEchoCancellation 121	iConfClient.NET.iConfClientDotNet.ScreenSharingStoppedDelegate 129
iConfClient.NET.iConfClientDotNet.GetAssemblyVersion 121	iConfClient.NET.iConfClientDotNet.SendMessage 124
iConfClient.NET.iConfClientDotNet.GetAudioOutputDevices 121	iConfClient.NET.iConfClientDotNet.SetAudioPlaybackType 124
iConfClient.NET.iConfClientDotNet.GetAudioOutputDevicesList 122	iConfClient.NET.iConfClientDotNet.SetImage 124
iConfClient.NET.iConfClientDotNet.GetInterfaceSafetyOptions 122	iConfClient.NET.iConfClientDotNet.SetInterfaceSafetyOptions 125
iConfClient.NET.iConfClientDotNet.GetUserScr 122	iConfClient.NET.iConfClientDotNet.SetOutputGain 125
iConfClient.NET.iConfClientDotNet.H263 119	iConfClient.NET.iConfClientDotNet.SetPlaybackDevice 125
iConfClient.NET.iConfClientDotNet.H264 119	
iConfClient.NET.iConfClientDotNet.iConfClientDotNet 118	
iConfClient.NET.iConfClientDotNet.IncomingVideoSizeChanged 130	
iConfClient.NET.iConfClientDotNet.IncomingVideoSizeChangedDelegate 127	
iConfClient.NET.iConfClientDotNet.Log 130	
iConfClient.NET.iConfClientDotNet.LogDelegate 128	
iConfClient.NET.iConfClientDotNet.Logging 122	
iConfClient.NET.iConfClientDotNet.MPEG4 119	
iConfClient.NET.iConfClientDotNet.Mute 122	

iConfClient.NET.iConfClientDotNet.SetViewerSizeMode 125	Log 130
iConfClient.NET.iConfClientDotNet.SetVolume 126	LogDelegate 128
iConfClient.NET.iConfClientDotNet.SocketError 131	Logging 122
iConfClient.NET.iConfClientDotNet.SocketErrorDelegate 129	MPEG4 119
iConfClient.NET.iConfClientDotNet.TextMessageReceived 131	Mute 122
iConfClient.NET.iConfClientDotNet.TextMessageReceivedDelegate 129	NewBitmapFrameAvailable 130
iConfClient.NET.iConfClientDotNet.UnregisterClass 126	NewBitmapFrameAvailableDelegate 128
iConfClient.NET.IObjectSafety 134	NewDesktopAvailable 130
iConfClient.NET.IObjectSafety.GetInterfaceSafetyOptions 135	NewDesktopAvailableDelegate 128
iConfClient.NET.IObjectSafety.SetInterfaceSafetyOptions 135	NewVideoFrameBytesAvailable 130
iConfClientDotNet class 112	NewVideoFrameBytesAvailableDelegate 128
about iConfClientDotNet class 112	ReconfigureAudioSystem 123
AcceptScreenSharingSession 119	RegisterClass 123
AutoReconnect 126	RejectScreenSharingSession 123
Call 119	RequestScreenControl 123
CallAccepted 129	RequestToShareScreen 130
CallAcceptedDelegate 127	RequestToShareScreenDelegate 128
CallbackObject 126	ScreenControlDenied 130
CallRejected 129	ScreenControlDeniedDelegate 128
CallRejectedDelegate 127	ScreenControlGranted 131
CaptureImage 120	ScreenControlGrantedDelegate 129
ClearImage 121	ScreenSharingStopped 131
ClientDisconnected 130	ScreenSharingStoppedDelegate 129
ClientDisconnectedDelegate 127	SendTextMessage 124
CurrentVideoHeight 126	SetAudioPlaybackType 124
CurrentVideoWidth 126	SetImage 124
Disconnect 121	SetInterfaceSafetyOptions 125
Dispose 121	SetOutputGain 125
EnableEchoCancellation 121	SetPlaybackDevice 125
GetAssemblyVersion 121	SetViewerSizeMode 125
GetAudioOutputDevices 121	SetVolume 126
GetAudioOutputDevicesList 122	SocketError 131
GetInterfaceSafetyOptions 122	SocketErrorDelegate 129
GetUserScr 122	TextMessageReceived 131
H263 119	TextMessageReceivedDelegate 129
H264 119	UnregisterClass 126
iConfClientDotNet 118	iConfServer.NET 25
IncomingVideoSizeChanged 130	iConfServer.NET namespace 25
IncomingVideoSizeChangedDelegate 127	Classes 25
	Interfaces 106
	iConfServer.NET.ICallback 106

iConfServer.NET.ICallback.ClientConnectionLost	108	iConfServer.NET.iConfServerDotNet.CaptureFrame	49
iConfServer.NET.ICallback.ConferenceRequest	108	iConfServer.NET.iConfServerDotNet.ChangePassword	50
iConfServer.NET.ICallback.ContactListReceived	108	iConfServer.NET.iConfServerDotNet.ChangePasswordResponse	100
iConfServer.NET.ICallback.ContactStatusChanged	108	iConfServer.NET.iConfServerDotNet.ChangePasswordResponseDelegate	88
iConfServer.NET.ICallback.IncomingCall	108	iConfServer.NET.iConfServerDotNet.ChangeStatus	50
iConfServer.NET.ICallback.JoinedConferenceRoom	109	iConfServer.NET.iConfServerDotNet.ClientConnectionLost	100
iConfServer.NET.ICallback.LoginResponse	109	iConfServer.NET.iConfServerDotNet.ClientConnectionLostDelegate	89
iConfServer.NET.ICallback.LogoutResponse	109	iConfServer.NET.iConfServerDotNet.ClientDisconnected	100
iConfServer.NET.ICallback.NewContactRequest	109	iConfServer.NET.iConfServerDotNet.ClientDisconnectedDelegate	89
iConfServer.NET.ICallback.PendingContactRequestResponse	109	iConfServer.NET.iConfServerDotNet.ConferenceMessageReceived	100
iConfServer.NET.ICallback.RegistrationResponse	109	iConfServer.NET.iConfServerDotNet.ConferenceMessageReceivedDelegate	89
iConfServer.NET.ICallback.SearchResponse	110	iConfServer.NET.iConfServerDotNet.ConferencePrivateMessageReceived	100
iConfServer.NET.ICallback.TextMessageReceived	110	iConfServer.NET.iConfServerDotNet.ConferencePrivateMessageReceivedDelegate	89
iConfServer.NET.ICallback.UserJoinedConference	110	iConfServer.NET.iConfServerDotNet.ConferenceRequest	100
iConfServer.NET.ICallback.UserLeftConference	110	iConfServer.NET.iConfServerDotNet.ConferenceRequestDelegate	90
iConfServer.NET.ICallback.VideoDeviceSelected	110	iConfServer.NET.iConfServerDotNet.ConferenceRequestRejected	101
iConfServer.NET.ICallback.VideoPreviewStarted	111	iConfServer.NET.iConfServerDotNet.ConferenceRequestRejectedDelegate	90
iConfServer.NET.ICallback.ViewUserResponseReceived	111	iConfServer.NET.iConfServerDotNet.ConnectedClientCount	87
iConfServer.NET.iConfServerDotNet	25	iConfServer.NET.iConfServerDotNet.ContactListReceived	101
iConfServer.NET.iConfServerDotNet.AcceptCall	48	iConfServer.NET.iConfServerDotNet.ContactListReceivedDelegate	90
iConfServer.NET.iConfServerDotNet.AcceptContactRequest	49	iConfServer.NET.iConfServerDotNet.ContactStatusChanged	101
iConfServer.NET.iConfServerDotNet.AddContactRequestResponse	100	iConfServer.NET.iConfServerDotNet.ContactStatusChangeDelegate	91
iConfServer.NET.iConfServerDotNet.AddContactRequestResponseDelegate	88		
iConfServer.NET.iConfServerDotNet.AddUserAsContact	49		
iConfServer.NET.iConfServerDotNet.AudioPlayBackType	46		
iConfServer.NET.iConfServerDotNet.AudioPlayBackType enumeration	46		
iConfServer.NET.iConfServerDotNet.audioType	46		
iConfServer.NET.iConfServerDotNet.audioType enumeration	46		
iConfServer.NET.iConfServerDotNet.AutoJoinRoomOnLogin	49		
iConfServer.NET.iConfServerDotNet.CallBackId	87		
iConfServer.NET.iConfServerDotNet.CallbackObject	87		
iConfServer.NET.iConfServerDotNet.CannotRunGraph	100		
iConfServer.NET.iConfServerDotNet.CannotRunGraphDelegate	88		

iConfServer.NET.iConfServerDotNet.ContactStatusMessageC hanged 101	iConfServer.NET.iConfServerDotNet.GetAssemblyVersion 53
iConfServer.NET.iConfServerDotNet.ContactStatusMessageC hangeDelegate 91	iConfServer.NET.iConfServerDotNet.GetAudioBytesReceived 53
iConfServer.NET.iConfServerDotNet.COPYDATASTRUCT 46	iConfServer.NET.iConfServerDotNet.GetAudioCompressorsF orRecording 53
iConfServer.NET.iConfServerDotNet.COPYDATASTRUCT structure 46	iConfServer.NET.iConfServerDotNet.GetAudioCompressorsF orRecordingStr 53
iConfServer.NET.iConfServerDotNet.CRLFDelimitedStringToL ist 50	iConfServer.NET.iConfServerDotNet.GetAudioInputDevices 53
iConfServer.NET.iConfServerDotNet.CurrentVideoCodec 87	iConfServer.NET.iConfServerDotNet.GetAudioInputDevicesLi st 54
iConfServer.NET.iConfServerDotNet.CustomObjectReceived 101	iConfServer.NET.iConfServerDotNet.GetAudioInputLevel 54
iConfServer.NET.iConfServerDotNet.CustomObjectReceived Delegate 91	iConfServer.NET.iConfServerDotNet.GetAudioInputLevel 54
iConfServer.NET.iConfServerDotNet.DeleteProfilePicture 51	iConfServer.NET.iConfServerDotNet.GetAudioOutputDevices 54
iConfServer.NET.iConfServerDotNet.DeleteUser 51	iConfServer.NET.iConfServerDotNet.GetAudioOutputDevices List 54
iConfServer.NET.iConfServerDotNet.DenyScreenControl 51	iConfServer.NET.iConfServerDotNet.GetAudioOutputVolume 54
iConfServer.NET.iConfServerDotNet.DesktopSharingType 46	iConfServer.NET.iConfServerDotNet.GetBoundIP 55
iConfServer.NET.iConfServerDotNet.DesktopSharingType enumeration 46	iConfServer.NET.iConfServerDotNet.GetBoundPort 55
iConfServer.NET.iConfServerDotNet.DisconnectClient 51	iConfServer.NET.iConfServerDotNet.GetConferenceRooms 55
iConfServer.NET.iConfServerDotNet.DisconnectedFromCentr alServer 101	iConfServer.NET.iConfServerDotNet.GetContactList 55
iConfServer.NET.iConfServerDotNet.DisconnectedFromCentr alServerDelegate 91	iConfServer.NET.iConfServerDotNet.GetFile 55
iConfServer.NET.iConfServerDotNet.DisconnectFromCentraLS erver 52	iConfServer.NET.iConfServerDotNet.GetLocalIp 55
iConfServer.NET.iConfServerDotNet.Dispose 52	iConfServer.NET.iConfServerDotNet.GetLocalIpStr 56
iConfServer.NET.iConfServerDotNet.EnableEchoCancellation 52	iConfServer.NET.iConfServerDotNet.GetMyCallBackId 56
iConfServer.NET.iConfServerDotNet.EnableGrayScale 52	iConfServer.NET.iConfServerDotNet.GetPendingContactRequ ests 56
iConfServer.NET.iConfServerDotNet.EnableVoiceActivation 52	iConfServer.NET.iConfServerDotNet.GetProfilePicture 56
iConfServer.NET.iConfServerDotNet.FileDownloadCompleted 101	iConfServer.NET.iConfServerDotNet.GetStatusMessages 56
iConfServer.NET.iConfServerDotNet.FileDownloadCompleted Delegate 91	iConfServer.NET.iConfServerDotNet.GetUserProfile 56
iConfServer.NET.iConfServerDotNet.FileDownloadProgress 101	iConfServer.NET.iConfServerDotNet.GetUserScr 57
iConfServer.NET.iConfServerDotNet.FileDownloadProgressD elegate 92	iConfServer.NET.iConfServerDotNet.GetVideoCompressorsF orRecording 57
	iConfServer.NET.iConfServerDotNet.GetVideoCompressorsF orRecordingStr 57
	iConfServer.NET.iConfServerDotNet.GetVideoDevices 57
	iConfServer.NET.iConfServerDotNet.GetVideoDevicesStr 58
	iConfServer.NET.iConfServerDotNet.GetVideoInputs 58

iConfServer.NET.iConfServerDotNet.GetVideoInputsStr	58	63	
iConfServer.NET.iConfServerDotNet.GetVideoSizes	58	iConfServer.NET.iConfServerDotNet.Listen	63
iConfServer.NET.iConfServerDotNet.GetVideoSizesStr	59	iConfServer.NET.iConfServerDotNet.Log	102
iConfServer.NET.iConfServerDotNet.GetVolume	59	iConfServer.NET.iConfServerDotNet.LogDelegate	93
iConfServer.NET.iConfServerDotNet.GrantScreenControl	59	iConfServer.NET.iConfServerDotNet.Logging	63
iConfServer.NET.iConfServerDotNet.H263	48	iConfServer.NET.iConfServerDotNet.Login	63
iConfServer.NET.iConfServerDotNet.H264	48	iConfServer.NET.iConfServerDotNet.LoginResponse	102
iConfServer.NET.iConfServerDotNet.H264Profile	47	iConfServer.NET.iConfServerDotNet.LoginResponseDelegate	
iConfServer.NET.iConfServerDotNet.H264Profile enumeration	47		93
iConfServer.NET.iConfServerDotNet.h264SpeedLevel	47	iConfServer.NET.iConfServerDotNet.Logout	64
iConfServer.NET.iConfServerDotNet.h264SpeedLevel enumeration	47	iConfServer.NET.iConfServerDotNet.LogoutResponse	102
iConfServer.NET.iConfServerDotNet.HoldRecording	59	iConfServer.NET.iConfServerDotNet.LogoutResponseDelegate	93
iConfServer.NET.iConfServerDotNet.iConfServerDotNet	45	iConfServer.NET.iConfServerDotNet.LoopBackAudioTest	64
iConfServer.NET.iConfServerDotNet.IncomingCall	102	iConfServer.NET.iConfServerDotNet.MotionDetected	102
iConfServer.NET.iConfServerDotNet.IncomingCallDelegate	92	iConfServer.NET.iConfServerDotNet.MotionDetectedDelegate	
iConfServer.NET.iConfServerDotNet.InitializeAudioSystem	60		94
iConfServer.NET.iConfServerDotNet.InvalidConferenceRoom Password	102	iConfServer.NET.iConfServerDotNet.MotionNotDetected	102
iConfServer.NET.iConfServerDotNet.InvalidConferenceRoom PasswordDelegate	93	iConfServer.NET.iConfServerDotNet.MotionNotDetectedDelegate	94
iConfServer.NET.iConfServerDotNet.InviteClientToViewScreen	60	iConfServer.NET.iConfServerDotNet.MPEG4	48
iConfServer.NET.iConfServerDotNet.InviteUsersToViewScreen	60, 61	iConfServer.NET.iConfServerDotNet.Mute	64
iConfServer.NET.iConfServerDotNet.IsConnected	61	iConfServer.NET.iConfServerDotNet.MuteContactInConference	64
iConfServer.NET.iConfServerDotNet.IsConnectedToCentralServer	87	iConfServer.NET.iConfServerDotNet.mycallbackid	48
iConfServer.NET.iConfServerDotNet.IsListening	87	iConfServer.NET.iConfServerDotNet.NewContactRequest	103
iConfServer.NET.iConfServerDotNet.IsScreenSharingEnabled	87	iConfServer.NET.iConfServerDotNet.NewContactRequestDelegate	94
iConfServer.NET.iConfServerDotNet.JoinConference	61	iConfServer.NET.iConfServerDotNet.NewDesktopAvailable	103
iConfServer.NET.iConfServerDotNet.JoinConferenceRoom	62	iConfServer.NET.iConfServerDotNet.NewDesktopAvailableDelegate	94
iConfServer.NET.iConfServerDotNet.JoinedConferenceRoom	102	iConfServer.NET.iConfServerDotNet.NewFileShareRequest	103
iConfServer.NET.iConfServerDotNet.JoinedConferenceRoom Delegate	93	iConfServer.NET.iConfServerDotNet.NewFileShareRequestDelegate	94
iConfServer.NET.iConfServerDotNet.LeaveConference	62	iConfServer.NET.iConfServerDotNet.NewVideoFrameAvailable	103
iConfServer.NET.iConfServerDotNet.LeaveCurrentConferenceRoom		iConfServer.NET.iConfServerDotNet.NewVideoFrameAvailableDelegate	

94	iConfServer.NET.iConfServerDotNet.RemoveContactResponse
iConfServer.NET.iConfServerDotNet.NewVideoFrameBytesAvailable	104
103	iConfServer.NET.iConfServerDotNet.RemoveContactResponseDelegate
iConfServer.NET.iConfServerDotNet.NewVideoFrameBytesAvailableDelegate	96
95	iConfServer.NET.iConfServerDotNet.RemoveNoiseFromOutgoingAudio
iConfServer.NET.iConfServerDotNet.Pause	64
iConfServer.NET.iConfServerDotNet.PendingContactRequest	103
iConfServer.NET.iConfServerDotNet.PendingContactRequestResponseDelegate	95
iConfServer.NET.iConfServerDotNet.PreviewHeight	88
iConfServer.NET.iConfServerDotNet.PreviewWidth	88
iConfServer.NET.iConfServerDotNet.ProfilePictureUpdated	103
iConfServer.NET.iConfServerDotNet.ProfilePictureUpdatedDelegate	95
iConfServer.NET.iConfServerDotNet.RateControl	47
iConfServer.NET.iConfServerDotNet.RateControl enumeration	47
iConfServer.NET.iConfServerDotNet.ReadyToShareScreen	103
iConfServer.NET.iConfServerDotNet.ReadyToShareScreenDelegate	95
iConfServer.NET.iConfServerDotNet.ReconfigureAudioSystem	65
iConfServer.NET.iConfServerDotNet.RecordingReadyToStart	104
iConfServer.NET.iConfServerDotNet.RecordingReadyToStartDelegate	95
iConfServer.NET.iConfServerDotNet.RecordToNewFileNow	65
iConfServer.NET.iConfServerDotNet.RegisterUser	65
iConfServer.NET.iConfServerDotNet.RegistrationResponse	104
iConfServer.NET.iConfServerDotNet.RegistrationResponseDelegate	96
iConfServer.NET.iConfServerDotNet.RejectCall	66
iConfServer.NET.iConfServerDotNet.RejectCallEx	66
iConfServer.NET.iConfServerDotNet.RejectConferenceRequest	66
iConfServer.NET.iConfServerDotNet.RemoveContact	67
	iConfServer.NET.iConfServerDotNet.RemoveContactResponse
	104
	iConfServer.NET.iConfServerDotNet.RemoveContactResponseDelegate
	96
	iConfServer.NET.iConfServerDotNet.RemoveNoiseFromOutgoingAudio
	67
	iConfServer.NET.iConfServerDotNet.RequestScreenControl
	67
	iConfServer.NET.iConfServerDotNet.ResumeRecording
	67
	iConfServer.NET.iConfServerDotNet.RevokeScreenControl
	67, 68
	iConfServer.NET.iConfServerDotNet.ScreenControlDenied
	104
	iConfServer.NET.iConfServerDotNet.ScreenControlDeniedDelegate
	96
	iConfServer.NET.iConfServerDotNet.ScreenControlGranted
	104
	iConfServer.NET.iConfServerDotNet.ScreenControlGrantedDelegate
	96
	iConfServer.NET.iConfServerDotNet.ScreenControlRequest
	104
	iConfServer.NET.iConfServerDotNet.ScreenControlRequestDelegate
	96
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionAccepted
	104
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionAcceptedDelegate
	97
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionAvailable
	104
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionAvailableDelegate
	97
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionRejected
	105
	iConfServer.NET.iConfServerDotNet.ScreenSharingSessionRejectedDelegate
	97
	iConfServer.NET.iConfServerDotNet.Search
	68
	iConfServer.NET.iConfServerDotNet.SearchResponse
	105
	iConfServer.NET.iConfServerDotNet.SearchResponseDelegate
	97

iConfServer.NET.iConfServerDotNet.SelectIPCameraAsVideo Source 68	77	iConfServer.NET.iConfServerDotNet.SetVolume 77
iConfServer.NET.iConfServerDotNet.SelectVideoCaptureDeviceAsVideoSource 69		iConfServer.NET.iConfServerDotNet.ShareFile 78
iConfServer.NET.iConfServerDotNet.SelectVideoDevice 69		iConfServer.NET.iConfServerDotNet.ShowVideoCaptureDevicePropertyDialog 78
iConfServer.NET.iConfServerDotNet.SelectVideoFileAsVideo Source 69		iConfServer.NET.iConfServerDotNet.StartConference 78
iConfServer.NET.iConfServerDotNet.SelectVideoInput 69		iConfServer.NET.iConfServerDotNet.StartConferenceEx 79
iConfServer.NET.iConfServerDotNet.SendCustomObject 70		iConfServer.NET.iConfServerDotNet.StartMotionDetection 79
iConfServer.NET.iConfServerDotNet.SendKeepAlive 70		iConfServer.NET.iConfServerDotNet.StartPreview 79
iConfServer.NET.iConfServerDotNet.SendMessage 70		iConfServer.NET.iConfServerDotNet.StartRecording 80
iConfServer.NET.iConfServerDotNet.SendMessageToConference 70		iConfServer.NET.iConfServerDotNet.StartRecordingAudioConference 80
iConfServer.NET.iConfServerDotNet.SendPrivatemessageTo User 70		iConfServer.NET.iConfServerDotNet.StartRecordingCurrentVideoPreview 80, 81, 82
iConfServer.NET.iConfServerDotNet.SendShareFileRequest 71		iConfServer.NET.iConfServerDotNet.StartRecordingCurrentVideoPreviewWmv9 82, 83
iConfServer.NET.iConfServerDotNet.SendTextMessage 71		iConfServer.NET.iConfServerDotNet.StartRecordingToWMV 83
iConfServer.NET.iConfServerDotNet.SetAudioInputlevel 71		iConfServer.NET.iConfServerDotNet.StartScreenSharing 83
iConfServer.NET.iConfServerDotNet.SetAudioOutputVolume 71		iConfServer.NET.iConfServerDotNet.StartTransmittingAudio 84
iConfServer.NET.iConfServerDotNet.SetAudioPlaybackType 72		iConfServer.NET.iConfServerDotNet.StatusMessagesReceived 105
iConfServer.NET.iConfServerDotNet.SetBrightness 72		iConfServer.NET.iConfServerDotNet.StatusMessagesReceivedDelegate 98
iConfServer.NET.iConfServerDotNet.SetContrast 72		iConfServer.NET.iConfServerDotNet.StopMotionDetection 84
iConfServer.NET.iConfServerDotNet.SetEncoderProperties 73, 74		iConfServer.NET.iConfServerDotNet.StopPreview 84
iConfServer.NET.iConfServerDotNet.SetEncoderPropertiesEx 74		iConfServer.NET.iConfServerDotNet.StopRecording 84
iConfServer.NET.iConfServerDotNet.SetFrameCaptureInterval 75		iConfServer.NET.iConfServerDotNet.StopRecordingAudioConference 85
iConfServer.NET.iConfServerDotNet.SetGainLevel 75		iConfServer.NET.iConfServerDotNet.StopRecordingCurrentVideoPreview 85
iConfServer.NET.iConfServerDotNet.SetHue 76		iConfServer.NET.iConfServerDotNet.StopScreenSharing 85
iConfServer.NET.iConfServerDotNet.SetIEMode 76		iConfServer.NET.iConfServerDotNet.StopTransmittingAudio 85
iConfServer.NET.iConfServerDotNet.SetOutputGain 76		iConfServer.NET.iConfServerDotNet.Talking 105
iConfServer.NET.iConfServerDotNet.SetPreviewFrameRate 76		iConfServer.NET.iConfServerDotNet.TalkingDelegate 98
iConfServer.NET.iConfServerDotNet.SetSaturation 76		iConfServer.NET.iConfServerDotNet.TextMessageReceived 105
iConfServer.NET.iConfServerDotNet.SetStatusMessage 77		iConfServer.NET.iConfServerDotNet.TextMessageReceivedDelegate
iConfServer.NET.iConfServerDotNet.SetUnlockCode 77		
iConfServer.NET.iConfServerDotNet.SetVoiceActivationLevel		

98	AutoJoinRoomOnLogin 49
iConfServer.NET.iConfServerDotNet.TextOverLay 85	CallBackId 87
iConfServer.NET.iConfServerDotNet.UpdateProfilePicture 86	CallbackObject 87
iConfServer.NET.iConfServerDotNet.UserJoinedConference 105	CannotRunGraph 100
iConfServer.NET.iConfServerDotNet.UserJoinedConferenceDelegate 98	CannotRunGraphDelegate 88
iConfServer.NET.iConfServerDotNet.UserLeftConference 105	CaptureFrame 49
iConfServer.NET.iConfServerDotNet.UserLeftConferenceDelegate 98	ChangePassword 50
iConfServer.NET.iConfServerDotNet.UserProfileReceived 105	ChangePasswordReponse 100
iConfServer.NET.iConfServerDotNet.UserProfileReceivedDelegate 99	ChangePasswordResponseDelegate 88
iConfServer.NET.iConfServerDotNet.VideoDeviceCount 88	ChangeStatus 50
iConfServer.NET.iConfServerDotNet.VideoDeviceSelected 106	ClientConnectionLost 100
iConfServer.NET.iConfServerDotNet.VideoDeviceSelectedDelegate 99	ClientConnectionLostDelegate 89
iConfServer.NET.iConfServerDotNet.VideoPreviewStarted 106	ClientDisconnected 100
iConfServer.NET.iConfServerDotNet.VideoPreviewStartedDelegate 99	ClientDisconnectedDelegate 89
iConfServer.NET.iConfServerDotNet.VideoRateControl 88	ConferenceMessageReceived 100
iConfServer.NET.iConfServerDotNet.ViewUser 86	ConferenceMessageReceivedDelegate 89
iConfServer.NET.iConfServerDotNet.ViewUserResponseReceived 106	ConferencePrivateMessageReceived 100
iConfServer.NET.iConfServerDotNet.ViewUserResponseReceivedDelegate 99	ConferencePrivateMessageReceivedDelegate 89
iConfServer.NET.iConfServerDotNet.waveOutGetVolume 86	ConferenceRequest 100
iConfServer.NET.iConfServerDotNet.waveOutSetVolume 86	ConferenceRequestDelegate 90
iConfServer.NET.iConfServerDotNet.WM_COPYDATA 48	ConferenceRequestRejected 101
iConfServer.NET.iConfServerDotNet.WM_USER 48	ConferenceRequestRejectedDelegate 90
iConfServer.NET.iConfServerDotNet.XmitDesktop 86	ConnectedClientCount 87
iConfServerDotNet class 25	ContactListReceived 101
about iConfServerDotNet class 25	ContactListReceivedDelegate 90
AcceptCall 48	ContactStatusChanged 101
AcceptContactRequest 49	ContactStatusChangeDelegate 91
AddContactRequestResponse 100	ContactStatusMessageChanged 101
AddContactRequestResponseDelegate 88	ContactStatusMessageChangeDelegate 91
AddUserAsContact 49	CRLFDelimitedStringToList 50
	CurrentVideoCodec 87
	CustomObjectReceived 101
	CustomObjectReceivedDelegate 91
	DeleteProfilePicture 51
	DeleteUser 51
	DenyScreenControl 51
	DisconnectClient 51
	DisconnectedFromCentralServer 101
	DisconnectedFromCentralServerDelegate 91
	DisconnectFromCentralServer 52
	Dispose 52

EnableEchoCancellation 52	H263 48
EnableGrayScale 52	H264 48
EnableVoiceActivation 52	HoldRecording 59
FileDownloadCompleted 101	iConfServerDotNet 45
FileDownloadCompletedDelegate 91	IncomingCall 102
FileDownloadProgress 101	IncomingCallDelegate 92
FileDownloadProgressDelegate 92	InitializeAudioSystem 60
GetAssemblyVersion 53	InvalidConferenceRoomPassword 102
GetAudioBytesReceived 53	InvalidConferenceRoomPasswordDelegate 93
GetAudioCompressorsForRecording 53	InviteClientToViewScreen 60
GetAudioCompressorsForRecordingStr 53	InviteUsersToViewScreen 60, 61
GetAudioInputDevices 53	IsConnected 61
GetAudioInputDevicesList 54	IsConnectedToCentralServer 87
GetAudioInputLevel 54	IsListening 87
GetAudioInputLevel 54	IsScreenSharingEnabled 87
GetAudioOutputDevices 54	JoinConference 61
GetAudioOutputDevicesList 54	JoinConferenceRoom 62
GetAudioOutputVolume 54	JoinedConferenceRoom 102
GetBoundIP 55	JoinedConferenceRoomDelegate 93
GetBoundPort 55	LeaveConference 62
GetConferenceRooms 55	LeaveCurrentConferenceRoom 63
GetContactList 55	Listen 63
GetFile 55	Log 102
GetLocalIp 55	LogDelegate 93
GetLocalIpStr 56	Logging 63
GetMyCallbackId 56	Login 63
GetPendingContactRequests 56	LoginResponse 102
GetProfilePicture 56	LoginResponseDelegate 93
GetStatusMessages 56	Logout 64
GetUserProfile 56	LogoutResponse 102
GetUserScr 57	LogoutResponseDelegate 93
GetVideoCompressorsForRecording 57	LoopBackAudioTest 64
GetVideoCompressorsForRecordingStr 57	MotionDetected 102
GetVideoDevices 57	MotionDetectedDelegate 94
GetVideoDevicesStr 58	MotionNotDetected 102
GetVideoInputs 58	MotionNotDetectedDelegate 94
GetVideoInputsStr 58	MPEG4 48
GetVideoSizes 58	Mute 64
GetVideoSizesStr 59	MuteContactInConference 64
GetVolume 59	mycallbackid 48
GrantScreenControl 59	NewContactRequest 103

NewContactRequestDelegate 94	ScreenSharingSessionAccepted 104
NewDesktopAvailable 103	ScreenSharingSessionAcceptedDelegate 97
NewDesktopAvailableDelegate 94	ScreenSharingSessionAvailable 104
NewFileShareRequest 103	ScreenSharingSessionAvailableDelegate 97
NewFileShareRequestDelegate 94	ScreenSharingSessionRejected 105
NewVideoFrameAvailable 103	ScreenSharingSessionRejectedDelegate 97
NewVideoFrameAvailableDelegate 94	Search 68
NewVideoFrameBytesAvailable 103	SearchResponse 105
NewVideoFrameBytesAvailableDelegate 95	SearchResponseDelegate 97
Pause 64	SelectIPCameraAsVideoSource 68
PendingContactRequest 103	SelectVideoCaptureDeviceAsVideoSource 69
PendingContactRequestResponseDelegate 95	SelectVideoDevice 69
PreviewHeight 88	SelectVideoFileAsVideoSource 69
PreviewWidth 88	SelectVideoInput 69
ProfilePictureUpdated 103	SendCustomObject 70
ProfilePictureUpdatedDelegate 95	SendKeepAlive 70
ReadyToShareScreen 103	SendMessage 70
ReadyToShareScreenDelegate 95	SendMessageToConference 70
ReconfigureAudioSystem 65	SendPrivatemessageToUser 70
RecordingReadyToStart 104	SendShareFileRequest 71
RecordingReadyToStartDelegate 95	SendTextMessage 71
RecordToNewFileNow 65	SetAudioInputlevel 71
RegisterUser 65	SetAudioOutputVolume 71
RegistrationResponse 104	SetAudioPlaybackType 72
RegistrationResponseDelegate 96	SetBrightness 72
RejectCall 66	SetContrast 72
RejectCallEx 66	SetEncoderProperties 73, 74
RejectConferenceRequest 66	SetEncoderPropertiesEx 74
RemoveContact 67	SetFrameCaptureInterval 75
RemoveContactResponse 104	SetGainLevel 75
RemoveContactResponseDelegate 96	SetHue 76
RemoveNoiseFromOutgoingAudio 67	SetIEMode 76
RequestScreenControl 67	SetOutputGain 76
ResumeRecording 67	SetPreviewFrameRate 76
RevokeScreenControl 67, 68	SetSaturation 76
ScreenControlDenied 104	SetStatusMessage 77
ScreenControlDeniedDelegate 96	SetUnlockCode 77
ScreenControlGranted 104	SetVoiceActivationLevel 77
ScreenControlGrantedDelegate 96	SetVolume 77
ScreenControlRequest 104	ShareFile 78
ScreenControlRequestDelegate 96	ShowVideoCaptureDevicePropertyDialog 78

StartConference 78
 StartConferenceEx 79
 StartMotionDetection 79
 StartPreview 79
 StartRecording 80
 StartRecordingAudioConference 80
 StartRecordingCurrentVideoPreview 80, 81, 82
 StartRecordingCurrentVideoPreviewWmv9 82, 83
 StartRecordingToWMV 83
 StartScreenSharing 83
 StartTransmittingAudio 84
 StatusMessagesReceived 105
 StatusMessagesReceivedDelegate 98
 StopMotionDetection 84
 StopPreview 84
 StopRecording 84
 StopRecordingAudioConference 85
 StopRecordingCurrentVideoPreview 85
 StopScreenSharing 85
 StopTransmittingAudio 85
 Talking 105
 TalkingDelegate 98
 TextMessageReceived 105
 TextMessageReceivedDelegate 98
 TextOverLay 85
 UpdateProfilePicture 86
 UserJoinedConference 105
 UserJoinedConferenceDelegate 98
 UserLeftConference 105
 UserLeftConferenceDelegate 98
 UserProfileReceived 105
 UserProfileReceivedDelegate 99
 VideoDeviceCount 88
 VideoDeviceSelected 106
 VideoDeviceSelectedDelegate 99
 VideoPreviewStarted 106
 VideoPreviewStartedDelegate 99
 VideoRateControl 88
 ViewUser 86
 ViewUserResponseReceived 106
 ViewUserResponseReceivedDelegate 99

waveOutGetVolume 86
 waveOutSetVolume 86
 WM_COPYDATA 48
 WM_USER 48
 XmitDesktop 86
 Installing the iConf .NET SDK 2
 IObjectSafety interface 134
 about IObjectSafety interface 134
 GetInterfaceSafetyOptions 135
 SetInterfaceSafetyOptions 135
 IP to IP Mode 5

J

JITTER enumeration member 46, 118

L

Logging in 16
 Logging out 17
 LOW_SPEED enumeration member 47
 LOWLATENCY enumeration member 46, 118

M

MAIN enumeration member 47
 MEDIUM_HIGH enumeration member 47
 MEDIUM_LOW enumeration member 47
 MYSQL SCRIPTS 10

O

Overview 1

R

Registering a new user 15

S

Sharing Files 23
 SQL SERVER SCRIPTS 11
 STABLE enumeration member 46, 118
 Starting a conference with 1 or more users 21
 Starting the Central Server 14

U

- Updating your Profile picture 23
- Using a MySQL Database 12
- Using a SQL Server Database 9
- Using the Central Server Service 8
- Using the Embedded Database 9

V

- VARIABLE enumeration member 46, 118
- Viewing a user (video stream) 22
- Viewing Pending Contact Request 19

W

- WindowsAudio enumeration member 46, 119