

## OPERATION MANUAL FOR DECISION REMOTE VOICE

This manual will help you learn how to use the Remote Voice demo Application. In order to have positive results you must follow the instructions carefully so that you will not have any trouble in using this application.

Requirements:

- 1)Sound Card
- 2)Microphone
- 3)Internet Account
- 4)Two Computers, one for Local side and other for Remote Side.

This application software has two major parts

- a)Server mode
- b)Client mode

Here are the instructions on how to use this application software.

If you are Server:

- 1)Run Remote Voice Application Software.
- 2)Set Connect Port so that Client side can use for connection.
- 3)Click the "Wait for Connecting" Icon in the Window Applet.
- 4)In order to send voice just click "Let me say!"
- 5)In order to receive response of remote side just click "Let me hear!"
- 6)You can change the record buffer size for enhancement of sound that you receive.

If you are Client:

- 1)Run Remote Voice Application Software.
- 2)Set the Remote IP and Connect Port on the computer in which you want to connect to.
- 3)Afterwards click on "Connect to".
- 4)In order to receive incoming voice on remote side just click "Let me hear!"
- 5)In order to respond to received voice just click "Let me say!"
- 6)You can change the record buffer size for enhancement of sound that you receive.

Note:

You must not set the Wave property to mute in your volume control located in your Windows tray because you will not hear the voice that will be sent to you by the remote side.

Also, you can record file and send it to remote side by using the options property "Recording" in your [Volume Control] Properties.

//        Decision Computer Development Tool kits : Remote Voice Control

File Name :        AsyncVoice.ocx

Attribute :        Active X Component (by MFC library)

[Method]

////////////////////////////////////

BOOL Connect(LPCTSTR IP, long Port);

Parameter :        IP        - remote IP address.  
                 Port        - remote listening port.

Return : 1        - successful.  
         0        - can't start the connecting action.

Remark :        try to connect to remote side ! (as Client side)

////////////////////////////////////

BOOL Listen(long Port);

Parameter :        Port        - local listening port.

Return : 1        - successful.  
         0        - can't start the listening action.

Remark :        try to listen at 'Port' (as Server side)

////////////////////////////////////

BOOL CloseConnect();

Return : 1        - successful.  
         0        - can't stop the connection !

Remark :        try to stop the current connection.

////////////////////////////////////

BOOL EnableSpeaker(BOOL State);

Parameter :        State        - 1, start the speaker  
                                 - 0, stop the speaker

Return : 1        - successful.  
         0        - can't control the speaker

Remark :        try to open or close the speaker at local side.

//

BOOL EnableRecorder(BOOL State);

Parameter :      State      - 1, start the recorder  
                                 - 0, stop the recorder

Return : 1            - successful.  
          0            - can't control the recorder

Remark :            try to open or close the recorder at local side.

//

long ReceiveBufferAmount();

Return : return the data receive buffer size

Remark :            return the receive buffer size, the degree is 0.5 sec.

//

long SenderBufferAmount();

Return : return the data send buffer size

Remark :            return the send buffer size, the degree is 0.5 sec.

//

short SetPlayingBufferSize(short size);

Parameter :        set the playing buffer size. (0 - 16)

Return : return the buffer size you has set.

Remark :            to set the playing buffer size, the degree is  
                         0.5 sec. That means you can set the delay time  
                         to hear the voice from remote side. It's very  
                         useful for the low bandwidth network. To get  
                         enough voice data, then playing !

//  
//  
//

[Event]

//

void OnAccept();

Remark :            someone has connected to local side (when you listen as server side)

////////////////////////////////////

void OnConnect();

Remark :            we has connected to the remote side (when you connect as client side).

////////////////////////////////////

void OnClose();

Remark :            when the connection was closed by remote side.