

Revision No: 1.0
Date: 9/22/99

DECISION CHAT

-The Decision Chat OCX is used to create Chat-Like Application. It allows you to send data from one computer to another through Internet / Intranet connection. The OCX supports multiple Server and Client connections. Thus a single computer can act as both Server and Client to other computers. Index no. and Names are used to distinguish one connection to another.

-The Decision Chat OCX is divided to 2 main components:

I. Methods / Functions

II. Events

I. Methods / Functions:

1. long DecisionPassword (string Password)

-this is the first and foremost method to be used. This will enable the OCX to work. Without this, the OCX will not function properly. The "Password" is DECISION COMPUTER.

e.g.

```
ret=ocx1.DecisionPassword("DECISION COMPUTER"); //specify the password to activate the OCX
```

2. long SetNoOfConnections (long NoOfConnections)

-this is used to specify the no. of connections that would be entertained by the computer, regardless the computer acts as a server or client.

e.g.

```
ret=ocx1.SetNoOfConnections(10); //i will only entertain up to 10 connections
```

3. long StartServer (long PortNo, String PassCode, String Name)

-the method signifies that this computer will act as a Server. The "PortNo" signifies the Port no. that will be used by the client to connect to server. The "PassCode" signifies the passcode needed by the client to connect to the server. And the "Name" signifies the name associated with the server connection in the client side. An event "ClientRequest" is triggered whenever a client has successfully connected with the server. And an Index no is given as one of its parameter, signifying the connection made. For the return value, please refer to the Error Codes section.

e.g.

```
ret=StartServer(4000,"AstroBoy","Server1"); //set this computer as a server named Server1
```

4. long ConnectToServer (string ServerIP, long ServerPort, string ServerPassCode, string Name)

-this method is used by the client computer to connect to the server. The "ServerIP" signifies the IP address of the server. "ServerPort" signifies the Port no. being used by the server. The "ServerPassCode" signifies the passcode assigned by the server. And the "Name" signifies the name associated with client connection in the server side. An event "ClientRequest" is triggered whenever a client has successfully connected with the server. And an Index no is given as one of its parameter, signifying the connection made. If the connection was unsuccessful an event "ConnectionFailed" will be issued together with an error code (refer to error codes section) as its parameter. For the return value, please refer to the Error Codes section.

e.g.

```
ret=ConnectToServer("202.54.117.67",4000,"AstroBoy","Client1"); //set this computer as client named as Client1
```

5. **string GetInfo (long Index)**

-this is used to get the name of the computer at the remote side, by specifying the index no. of the connection. The "Index" is the assigned no. for the connection to the remote side. When the computer is acting as server, the event "ClientRequest" is issued together with the Index no as its parameter whenever a client connects to our computer. When the computer acts as client, the event "Connected" is issued together with the Index no as its parameter whenever we have a successful connection with the server. The name of the remote computer will be given, otherwise a "NO CONNECTION" value will be returned.

e.g.

```
name=GetInfo(1); //get the name of the remote computer using the index no of the connection
```

6. **long GetActiveConnections ()**

-this is used to report the no. of active connections we have with other computers. This is regardless of the state of the computer(Server, Client or both Server and Client).

e.g.

```
active_connections=GetActiveConnections();
```

7. **long SendData (long Index, string Data)**

-this is used to send data to the remote side of the connection. The "Index" signifies the index no. associated with a connection to a remote side. The "Data" signifies the data to be sent to the remote side. An event "DataReceived" will be triggered whenever a data is received from the remote side. For the return value, please refer to the Error Codes section.

e.g.

```
ret=SendData(0,"Hello World"); //send data to the computer connected as index 0
```

8. **long CloseServer ()**

-this is used to disable the computer to act as a server. All client connections to the computer will be automatically cut off.

e.g.

```
ret=CloseServer(); //stop being the server
```

9. **long CloseConnection (long Index)**

-this method acts like a "kick" function. It will terminate any connection made to a remote computer by specifying the index no. of the connection. For the return value, please refer to the Error Codes section.

e.g.

```
ret=CloseConnection(2); //kick the remote computer designated as connection no. 2
```

II. Events:

1. **ClientRequest (long Index)**

-This event is only applicable if the computer is acting as Server or as both Server and Client. This is an event triggered whenever a remote computer(client) has made a successful connection with our computer(acting as server). The "Index" parameter signifies the unique no. that is used to identify the connection. To get the name of the computer on the remote side, use the "GetInfo" function.

2. **Connected (long Index)**

-This event is only applicable if the computer is acting as Client or as both Client and Server. This is an event triggered whenever the computer(acting as client) has made a successful connection to a server. The "Index" parameter signifies the unique no. that is used to identify the connection. To get the name of the computer on the remote side, use the "GetInfo" function.

3. ConnectionFailed (long ErrorCode)

-This event is only applicable if the computer is acting as Client or as both Client and Server. This is an event triggered whenever the computer(acting as client) is unsuccessful to connect to a server. The "ErrorCode" describes the reason of failure. Please refer to Error Codes.

4. DataRecieved (long Index, string Data)

-This event is triggered whenever data is recieved from the remote side. The "Index" describes from which connection the data came. The "Data" is the actual data recieved from the remote side.

5. Disconnected (long Index)

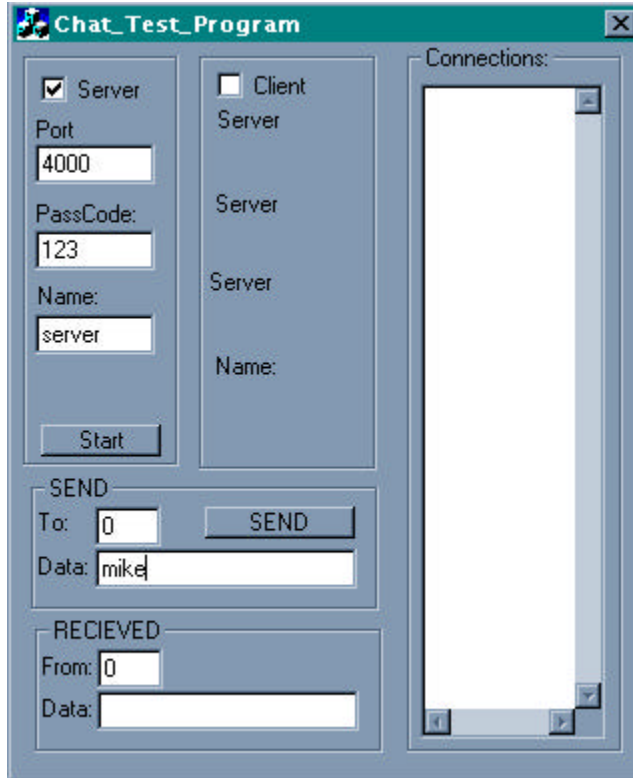
-This event is triggered whenever a connection from a remote side has been terminated. The "Index" signifies which connection has been terminated.

Error Codes:

- 0** - No Problem / Connected
- 1** - Error on Connection
- 2** - Connection is Closed
- 3** - Wrong Passcode
- 4** - Time Out
- 5** - PortNo already in use
- 6** - No Connection
- 10** - Server Already active
- 11** - No more available connections on client side
- 12** - No more available connections on server side

Demo Chat Operation Manual

Server Application:



1. To initialize Server application, click the Server Check Box on the top of the Demo Chat Applet.
2. Provide the necessary data to start Server Application like Server Port, Passcode, and Name in the Server window then click "Start" to implement settings.
3. To send data, type the data on the Data window of the Server afterwards click send for data transmission to occur.

Client Application

The screenshot shows a Java applet window titled "Chat_Test_Program". It contains several input fields and controls for configuring a client application. On the left, there are checkboxes for "Server" (unchecked) and "Client" (checked). Below these are labels for "Port", "PassCode:", and "Name:". To the right of these labels are corresponding text input fields. The "Server" field contains "127.0.0.1", the "Port" field contains "4000", the "PassCode:" field contains "123", and the "Name:" field contains "client". Below these fields is a "Start" button. At the bottom left, there are two sections: "SEND" and "RECEIVED". The "SEND" section has a "To:" field with "0" and a "SEND" button, and a "Data:" text area. The "RECEIVED" section has a "From:" field with "0" and a "Data:" text area. On the right side of the window is a large, empty text area labeled "Connections:".

1. To initialize Client Application for Demo Chat click on the Client Check Box in the Demo Chat Applet.
2. Provide the necessary data required like IP Address of Server on the first window, Server Port on the second, Passcode on the third, and the "Name" you wish to be recognized on the network.
3. To send data, type the data on the Data window of the Client afterwards click send for data transmission to occur.