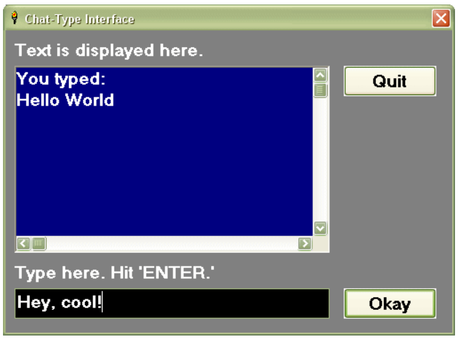
Chat Window Prototype

author

What is a chat window?

A chat window generally consists of a textbox where the user can type some text, and a texteditor that displays the text typed by the user(s) and scrolls as the chat progresses. When the user hits ENTER, the text he has typed into the textbox is copied to the texteditor and the textbox is cleared.



```
You typed:
Hello World
```

```
Hey, cool!
```

When is it used?

Online chatrooms use chat windows. Desktop chat programs and text-based games also use chat windows.

In a desktop chat program, you might display a question to a user, and when he types his response and presses ENTER, your program evaluates his response and gives him a message. The classic Eliza program worked this way. As an example, a user might type, "I feel good today." The program might respond, "I'm glad you feel good today." You can try an Eliza program online here: http://www.manifestation.com/neurotoys/eliza.php3

Text-based adventure games also use chat windows. The user might type a command, such as "north" and the program responds with a description of the new situation. One response might be, "You now enter a room with a window on the east wall and a table in the center."

Dialog Windows

Dialog windows are the best choice for a chat window. They cannot contain menus, but menus are rarely useful for these types of applications. The big advantage of dialog windows is their ability to have a default button. A button given the extension .default is activated whenever the user presses the ENTER key. You'll see why this is useful later in the article.

Getting input.

A simple textbox allows the user to type his messages. It is usually located near the bottom of a chat window.

```
textbox     #main.type, 10, 290, 355, 34
```

Next to the textbox, there is a button that has the caption OKAY. The button has the extension .default. Because it is in a dialog window, this button's event handler is activated whenever the user presses ENTER. This allows the user to keep his fingers on the keyboard without needing to pick up the mouse and click the OKAY button.

```
button      #main.default, "Okay",[okay],UL, 380, 290, 105, 34
```

The event handler for the default button must include code to retrieve the contents of the textbox. The !contents command does that. Remember to begin commands to textboxes with a ! character to signal to the textbox that this is a command. If you do not, the text command is displayed in the textbox. After the command is issued, the string variable contains the text from the textbox. This example uses a string varible called txt$.

```
#main.type "!contents? txt$"    'get text typed by user
```

Can the user type into the texteditor display?

The texteditor portion of the chat window is commonly used for display purposes only. To prevent the user from typing into this texteditor, issue a !disable command, like this:

```
#main.display "!disable"    'user cannot type into display box
```

Textbox vs Texteditor

Textboxes have a different set of commands than texteditors. They are far more limited. Use the proper command set for each control. Do not attempt to use texteditor commands in textboxes, and vice versa.

Handling user input.

Once the contents are contained in the string variable, they can be displayed in the texteditor quit simply. Printing any text, whether a literal string, or a string variable, causes the text to be added to the end of the text in the texteditor.

```
#main.display txt$              'put user text into texteditor
```

If you ever need to clear the texteditor, use the CLS command.

```
#main.display "!CLS"
```

Once the program has retrieved the text entered by the user, it should clear the textbox so that it is ready for the next user input. The textbox does not have a !CLS command. Any text printed to a textbox replaces the current contents, so to clear the contents, you can print an empty string to the textbox, like this:

```
#main.type ""                   'clear typing box
```

You must do one more thing after retrieving the user's text, displaying it in the texteditor, and clearing the textbox. You must !setfocus to the textbox, so that the next time the user types something, the keyboard input goes into the textbox.

```
#main.type "!setfocus"  'set focus so typing goes into textbox
```

Evaluating user input.

The code so far takes user input, displays it in the texteditor, and clears the textbox for more user input. The program needs to evaluate the user input in some way, and then give the user instructions or a response of some sort. The kind of response depends on the type of program. For this demo, the program simply gives the user a message that he has typed the following text, then displays the user's text.

```
    #main.display "You typed:"      'put message in texteditor
    #main.display txt$              'put user text into texteditor
```

In a real program, the user's input must be evaluated so that a correct response can be displayed by the program.

Challenge

Why not take this chat window template and create an interactive, artificially intelligent chat program like Eliza? Or you can create a text-based adventure game. You can add a graphicbox to display objects or a map. You can add statictext controls to indicate which objects a character has collected. There are many possibilities!

Demo

```
'** Created by LB Workshop - 2/9/2005 6:45:50 AM
'** Chat-Type Interface

    True = 1 : False = 0


[InitColors]
    'Please use default colors when possible.

'Use the following color statements only if necessary in the program.
    ForegroundColor$ = "white"
    BackgroundColor$ = "darkgray"
    TexteditorColor$ = "darkblue"
    TextboxColor$    = "black"
    'ComboboxColor$   = "White"
    'ListboxColor$    = "White"

[WindowSetup]
    NOMAINWIN
    WindowWidth = 510 : WindowHeight = 370
    UpperLeftX = INT((DisplayWidth-WindowWidth)/2)
    UpperLeftY = INT((DisplayHeight-WindowHeight)/2)

[ControlSetup]
statictext  #main.s1, "Type here. Hit 'ENTER.'", 10, 260, 369, 25
statictext  #main.s2, "Text is displayed here.", 10, 10, 369, 25
button      #main.default, "Okay",[okay],UL, 380, 290, 105, 34
button      #main.quit, "Quit",[quit],UL, 380, 40, 105, 34
textbox     #main.type, 10, 290, 355, 34
texteditor  #main.display, 10, 40, 355, 210

Open "Chat-Type Interface" for Dialog as #main
    #main "trapclose [quit]"
    #main "font ms_sans_serif 14 bold"
    #main.display "!disable"    'user cannot type into display box

[loop]
    Wait

[quit]
    close #main : END

[okay]
    #main.type "!contents? txt$"    'get text typed by user
    #main.display "You typed:"      'put message in texteditor
    #main.display txt$              'put user text into texteditor
    #main.type ""                   'clear typing box
    #main.type "!setfocus"  'set focus so typing goes into textbox
    wait
```

Chat Window Prototype | What is a chat window? | When is it used? | Dialog Windows | Getting input. | Can the user type into the texteditor display? | Textbox vs Texteditor | Handling user input. | Evaluating user input. | Challenge | Demo