# UUCODER.VBX

Please take a moment to read this document before using UUCODER.VBX. It explains the properties and events supported by this Control.
Included in the UUCODER.EXE file is UUDEMO.MAK which demonstrates how to use UUCODER.VBX.

**Overview.**
UU Coding is a common techniqus used on the internet to post binary files. The contents of a binary file are 'encoded' up into text characters that can be handled by all mail handlers. A UU Decoder is simply a program that takes the encoded data and recreates the original file. This control supports both UU and XX Encoding\Decoding.
This product is shareware. You can register in two ways.

1. GO SWREG on Compuserve.
2. Send 10$ + 2$ S&H to:

Steve Bridges
76 Nash St 2nd Floor
New Haven CT 06511.

Make checks payable to Steve Bridges.

Registered users receive the latest version plus the source code.

If you have any questions or problems you can contact me at:

71507,1033@compuserve.com
STEVEB3019@aol.com

**Properties.**

**InFile.**

Specifies the name of the file to process. For an encoding operation this is the name of the file to be encoded. For a decoding operation this is the name of the file containing the UU or XX coded data.

Example.

```
Form.Control.InFile = "C:\TEMP\INPUT.FIL"
```

**OutFile.**

Specifies the name of the file that will hold the encoded data. This property is only used for an encoding operation. If this property is left blank then the control will default the output file name to the input file name with an extension of .UUE or .XXE depending on the encoding method chosen.

Example.

```
Form.Control.OutFIle = "C:\TEMP\INPUT.UUE"

Form.Control.OutFile = "C:\TEMP\"      ' specify a directory only. The control
                                       ' will default the file name and put the
                                       ' encoded files here.
```

**Type.**

Specifies the type of encoded file to create or decode.

Values:

0 Use UU Encoding\Decoding
1 Use XX Encoding\Decoding
2 Auto Detect based on file contents.

For encoding operations a type of 2 will default to creating a UU encoded file. For decoding operations a type of two will analyze the input data and attempt to determine the coding method.

Example.

```
Form.Control.Type = 1
```

**Open.**

This property is only relevant for decoding operation. If it is set to true then the control will attempt to execute the application associated with the type of output file created. (i.e NOTEPAD.EXE for .TXT files).

Note.
If the input data contained multiple output files then the associated application will be invoked for each file created.

Example.

```
Form.Control.Open = true
```

**Action.**

Setting this property will cause the control to either encode or decode a file.

Values.
0 Decode a file back into it's original
1 Encode a file

Example.

```
Form.Control.Action = 1
```

**LinesPerFile.**

This property is used for encoding only. It determines the maximum lines of encoded data to write to an output file. The control will create multiple output files each holding up to this propertys values lines of data. The default seting is 950. A setting of zero prevents the control from splitting the encoded file up into smaller files.
The output files are sequentially numbered.

Example.

```
Form.Control.LinesPerFile = 10
Form.Control.InFile = "INFILE.DAT"
Form.Control.Action = 1
```

Assuming that the data in the input file would cause three output files to be created then the output files would be named:

INFIL1.UUE
INFIL2.UUE
INFIL3.UUE

**Result.**

Specifies the result of the encoding or decoding operation. This is a read only property.

Values
20000 - Invalid Action supplied
20001 - An error occurred opening an input or output file.
20002 - No valid encoded data was found during a decode action.
20003 - The operation completed normally.
20004 - The specified file was not found.
20005 - No input file was supplied.

**Events.**

The control supports one event. This event is passed a message number and a string. Some of the messages are informational and can be ignored, others require some action. The messages are detailed below.

sub ctlname_Message (MsgNum as integer, Msg as string)

MsgNum 1 (OUTPUT_FILE)
Msg File name of file being created from the encoded data. If the input file contains multiple encoded files then you will receive this message for each file decoded. This message is sent for each 'begin nnn filename.exe' that is encountered. This is an informational message.

MsgNum 2 (GET_NEXTFILE)
Msg Indicates that the next part of a multi file encoded file could not be found. For a multi part file the control builds the name of the next input file based on the name of the current input file being processed. It does this by assigning numbers to the end of the input file name. i.e Control.Form.InFile = "UUFILE1.UUE". The next file that the control will attempt to use is: UUFILE2.UUE. If the control cannot find this file it trigger this message passing the name of the file it tried to open. Your code must supply the name of the next file to be used.

MsgNum 3 (WRITE_FILE)
Msg specifies the next file name that will be used for a multiple output file encoding operation. Multiple output filenames are generated by adding a number to the end of the filename. E.g

Form.Control.InFile = "UUFILE.DAT"

If this encoded to three output files the control would default them to:

UUFIL1.UUE
UUFIL2.UUE
UUFIL3.UUE

You can use this event to modify the name that will be used. To do this simply change the Msg to the file name you wish to use.

| | |
|---|---|
| MsgNum | 4 (DECODING_FILE) |
| Msg | Specifies the next file that the control will use in a multi file decoding operation. This is an informational message only. Any modifications to the message are ignored. |