UltraClip
The Virtual Clipboard
© Doug Overmyer 1992-1994

Function
UltraClip creates and manages a **virtual clipboard** for the Windows environment. It lets you

* create mini-clipboards that store snapshots of the Windows clipboard
* create OLE objects with embedded or linked data
* view graphical thumbnails for quick retrieval
* track the contents of the Windows clipboard in an iconic window
* hotkey to autosequence through miniclips without switching to UltraClip
* print proof sheets of the thumbnails or graphics from a disk file
* save and retrieve the mini-clipboard objects as .CLP files
* directly create mini-clips from .BMP, .PCX, and .WMF (Aldus) images stored on disk
* save and restore the current UltraClip desktop and all its objects
* edit and manipulate mini-clips with text objects < 55K
* edit OLE objects by double-clicking
* create libraries of graphics and text for various projects

Prerequisites
UltraClip requires Windows 3.1.

Constraints
UltraClip stores everything in memory - you'll need sufficient memory to support whatever you wish to store. Large color bitmaps can consume several megabytes, so beware. UltraClip ignores owner-display formats. Not compatible with the original release of Norton Desktop for Windows 1.0. Printing requires a print and video driver that supports StretchDIBits().

Installation
Installing UltraClip involves two operations:

* Copy UC.EXE, UC.HLP and UCHOOK.DLL to any suitable directory.

*_Carefully_ Install the CTL3D.DLL in the windows system directory using the following steps:

1. Check your Windows system directory for CTL3D.DLL, usually c:\windows\system.
2. If you don't find it, copy the new one into the system directory
3. If you find an existing copy, try using UltraClip with it - if it works, stick with it.
4. If it doesn't work, rename the old version to CTL3D.OLD, copy the new version into
   the system directory, and test it.
5. If other programs have problems (this shouldn't happen), delete CTL3D.DLL, and
rename CTL3D.OLD to CTL3D.DLL. UltraClip is not compatible with your system.

Launch UltraClip from the Program Manager or your favorite menu system. UC does not have any special path or directory requirements.

Upgrades from VERSIONS 1.6 and Earlier

☞ Unfortunately, UltraClip 2.0 does not read the desktop save files used by versions 1.6 and earlier. A slight change in format was needed to accommodate the OLE data streams. The best strategy would be to export the most needed miniclips as .CLP files, then load them into UC individually.

Introduction
A Few Terms
> **Clipboard** - refers to the Windows clipboard
> **miniclip** - refers to the child windows in UltraClip which contain mini-clipboards

The UltraClip Display Window
UC is an MDI application with a button and status bar along the top of the window area, and separate windows for each mini-clipboard.

The status bar is arranged as follows:

Buttons(left to right):
> Cut to Clipboard - copy currently active window to clipboard , then delete image window
> Copy to Clipboard - copy currently active window to clipboard
> Paste from Clipboard - create new image window from clipboard
> Delete Window - remove image window from UC
> Clear Clipboard - delete current contents of clipboard
> Array - tile the image windows in a grid for easy viewing

In the middle :
> a small iconic view of the current contents of the clipboard if displayable.

To the right :
> several 3D boxes giving various pieces of information about the active mini-clip window and system configuration:
>
> top row:
> > Format sizes associated with the active mini-clip
>
> bottom row:
> > Status of Auto Paste mode(P)
> > Status of AutoTrack mode(T)
> > Status of AutoSave mode(S)
> > Free Global memory
> > Module/program name of source of data

In the open area are **miniclips** which display thumbnails of the objects currently stored in UltraClip. Each miniclip has its own window. You can drag these windows around in the display area, iconize them, tile them or use the other options on the Window menu. If you zoom a miniclip, the object is displayed full size. If you resize the main window, the miniclips are retiled automatically.

The Windows Clipboard & OLE
The Clipboard is largely a convention that lets programs share data directly via system memory. When you "copy" to the clipboard, your application formats various blocks of global memory and turns them over to the Clipboard (the Windows system) to manage. Any application can retrieve those blocks from the system, make copies, and interpret them as best as it can. An application can use both predefined formats and custom formats when creating the data for the clipboard. Since the various formats usually refer to the same general item , we can collectively call them a clipboard 'object' which has various formats. When you copy to the clipboard or empty it, you remove the existing object and replace it with another.

UltraClip allows you to take a snapshot of the clipboard and preserve it as a miniclip. These snapshots are also stored in global memory, just like the original, but have a longer life. Together they form a virtual clipboard where objects persist after a cut or copy, where you can pick and choose one for quick retrieval, and even manipulate its text. You can also save the miniclips to disk for later retrieval as .CLP files.

OLE is an advanced protocol for sharing data among Windows applications. One approach is to copy OLE data to the clipboard for pasting into a client application. Ultraclip supports OLE 1.0 in two ways. You can save all the data on the clipboard as a regular miniclip (and restore it later), or you can create an OLE miniclip which actually contains an OLE object. Both linked and embedded objects are supported. There is much more system support for these OLE objects than for simple clipboard data - the OLE libraries manage the binary data, read and write it to disk, and even control the display. One unique feature of an OLE object is that you can double-click on it and edit the data in the server application.

UltraClip supports the OLE 2.0 clipboard formats which allow you to save and restore the OLE data for use in other programs. Linking and Embeding within UltraClip using the OLE 2.0 protocoll is not available.

Operation
Using UltraClip
Once UC is running, just use the paste icon or menu option to paste the current contents of the Clipboard into UC. If there is an object on the Clipboard, UC makes a copy, creates a miniclip, and displays it. You can store as many miniclips as memory and resources allow. If OLE data is present, you can also create an OLE miniclip - select **Edit/Link** or **Edit/Embed** as desired.

To restore a miniclip to the Windows Clipboard so that other applications can use it, highlight the window you wish to access, and either click the **Copy** button, choose **Edit/Copy** from the menus, or **right-click** the window's title bar. This will copy the object to the Clipboard, replacing whatever was there before. You can also choose **Edit/Cut** (or the **Cut Button**) to copy the object to the Clipboard and delete the miniclip.

To selectively copy one or more formats (in a particular miniclip) to the Clipboard, right-click (secondary button) anywhere inside the miniclip. A dialog box with the available formats will appear. Click on the formats you are interested in, then **OK** to copy the selected formats to the Clipboard. The miniclip retains a copy of all the formats until you close it, or exit the program. If you chose the wrong format the first time, just repeat the copy.

To cycle thru the available miniclips, putting each in turn on the clipboard, use **Ctrl + the UC hot key**. Be sure that one of the miniclip windows has been selected (it does not have to be active), then hit **Ctrl+hotkey**. You can even do this while running another application. UltraClip calls this 'Auto Sequencing'.

You can configure several options for UltraClip using the File/Configure menu option. Choose the startup mode you desire and it will take effect the next time you start UltraClip.

Clicking the array button arranges the miniclips in a grid much like a slide-sorter for your camera. The default sizes the thumbnails to 125 pixels square and fits as many as possible in the current window area. See the configuration section for changing the default.

You can clear the current contents of the Clipboard using the Edit/Clear menu or the Clear button. This may free up some memory if the Clipboard is currently storing a large image.

When you depress an icon button, the status display box temporarily displays a brief explanation of the action.

You can edit and manipulate the text in a miniclip if the text size is < 55 K.The Edit/Modify menu option opens an edit window where you can change, add, move, append to Clipboard, and delete text. Double clicking with the left (primary) mouse button will also open an edit window.

You can edit an OLE miniclip by double-clicking the object. The host application will start where you can use all its facilities in editing. Before exiting choose **Update** to transfer the data back to UltraClip.

The Edit/Append menu choice appends the text object in the active miniclip to whatever text is on the clipboard.   You can use these options repeatedly to combine text in nearly any fashion.

You can launch the UltraClip help file from the **Help** menu or with the **F1** hotkey.

AutoPaste, AutoTrack and AutoSave Modes
AutoPaste - UltraClip has an Auto Paste mode that will automatically paste every new object placed on the Clipboard (except those copied from UC). To activate AutoPaste choose the Edit/Start AutoPaste menu option or click the status box with the right mouse button. The status box displays P or _ depending on the current status of AutoPaste.

AutoTrack - UltraClip will display an iconic miniature of the current contents of the Clipboard if you configure AutoTrack to be active. This iconic display will also appear when UltraClip itself is iconized, and will be updated whenever the Clipboard changes.

UltraClip uses 3 special icons when there are no public graphic formats on the clipboard:

    X - Clipboard is empty
    T - Clipboard has only Text and non-displayable formats
    01 - Clipboard has only binary (non-displayable) formats

AutoSave - You can have UltraClip automatically save the UC desktop every *n* minutes. You set this option in the **File/Configuration** dialog. When you first start UltraClip, it checks the AutoSave time period. If the period is > 0 it turns on AutoSave. Then, every *n* minutes UC checks to see if there are any miniclips on the desktop. If so, it puts up a dialog asking if you really want to overwrite the desktop save file. You have three choices:

    Yes: autosave will now run automatically for the rest of the session
    No: start over counting the minutes and repeat the query process until you answer Yes or Cancel
    Cancel: autosave is suspended for the session or until you reactivate it

Thus you are queried at the start of each session to help insure that the desktop save file is not accidentally overwritten.

The Miniclip Windows
Each miniclip displays a thumbnail of the object it is 'storing' - either a graphic or an unformatted dump of text. When you paste an object into UC, it stores all the available formats as well as building a thumbnail image if a graphic format is found.

If you create an OLE miniclip, the display is controlled by the OLE libraries as are the available formats. Regretfully these objects display rather more slowly than the UC thumbnails which are optimized for speed. The available formats for restoring to the clipboard are also determined by the OLE system.

If the object contains no displayable formats, the miniclip displays the program name that put the data on the Clipboard, and the formats that are available for copying. You may have to zoom the window to see all the text if there are many formats.

The title bar of the miniclip window displays the origin of the image - the file name if read from disk, or the module that put the object on the Clipboard (if available).

If you **zoom** a miniclip, the graphic is redrawn full size. Depending on the size and complexity of the object, this can take a while the first time. The delay is caused by the time needed to create the full image -

initially, only the thumbnail is built. The full graphic is cached in memory, though, so you don't need to wait a second time.

Zoomed miniclips have scroll bars that let you move around the image or scroll through text. Please note that only ten screens of text are maintained for scrolling - this limitation is currently needed for performance reasons. Graphics are available for scrolling regardless of the image size.

Status Windows
UltraClip displays several pieces of information to help manage the environment. There are status boxes for:

    *AutoPaste status
    *AutoTrack status
    *AutoSave status
    *free global memory
    *source of data in the clip window
    *major formats and size stored in the clip object
    *iconic clipboard view

You can right click on several of these status displays and UC will pop-up an expanded view of the status:

    *Free memory -->Free memory + free system resources
    *Formats --> all formats available + global size
    *iconic view--> all formats on the clipboard

Auto Sequencing

This feature lets you sequentially restore the available miniclips to the clipboard by using the Auto Sequence hot key. You define this key using the **File/Configure** dialog. Just make sure that one of the miniclip windows has been selected, and use **{Ctrl + Auto Sequence hot key }** to place each miniclip in turn on the clipboard. You can do this while any application is active so long as UC is running. It also works if UC is iconized.

You can reconfigure the hotkey at any time - it takes effect immediately. If you configure a hotkey that is also used by another application, UC may never see the keystroke (this fascist behavior by the other application is deplorable, but there is no recourse). The best solution is to choose another hotkey. You should choose a key from 'A' to 'Z' or '0' to '9'.

If you wish to control the order that the miniclips are copied, just click the **title bars** of the miniclips in the order that you wish them copied. If you iconize UC, you can see the various miniclips as an icon when you cycle through the miniclips.

Drag & Drop Interface

There are several ways of getting work done in UltraClip. The menu choices are the most familiar. In addition there are several situations where you can right click on a window to start a particular action - see Mouse Work for a description.

A third approach uses the drag and drop object metaphor. The idea is that you select an object and 'drop' it on an icon or area that symbolizes what you want to accomplish. The cursor changes to an up-arrow when over an item that will accept a drop. In UltraClip two objects are 'droppable': **miniclips** and the iconic **clipboard window**. Select a miniclip, then click inside it and drag - the cursor changes to an up arrow. Here are the players for drag and drop:

Source	*Target*	*Action*
miniclip	Cut button	copy to clipboard and delete
	Copy button	copy to clipboard

	Delete button	delete miniclip
	Array button	array the open miniclips
	Clipboard icon	copy miniclip to clipboard
clipboard	Paste button	create a miniclip from clipboard
(icon)	Clear button	empty clipboard contents
	Client area	create a miniclip from clipboard

If you have many palette-intensive miniclips, you may have to wait a second when you select a new miniclip - Windows is at work updating the image displays, and will miss the mouse drag if you start too soon.

Edit windows for text also use drag and drop - see next topic.

Manipulating Text

You can edit and manipulate text objects which are less then 55K in size. Two options are available on the Edit menu:

* Append to Clipboard: takes the text object in the active miniclip and appends it to the text on the Clipboard. The resulting new text object replaces the old one on the Clipboard.

* Modify Text: opens an editing window where you can rearrange, add and delete text.

You can also open an edit window by double clicking inside the miniclip with the left (primary) mouse button. AutoPaste is suspended whenever an edit window is open.

The edit window has several features:

* Drag & Drop Editing:
Select a block of text, then move the cursor back over the text and press. The cursor changes to an up arrow, and you can 'drag' the text to any new position in the text. If you reach the edge of the edit window, the window will automatically scroll the remaining text into view. You can spot your 'drop' by following the caret(a light gray bar) which shows the current insertion point.

If you hold down the CTRL key when you drop the text, the selected text will be *copied* to the new spot without deleting the source. Otherwise, the text is *moved* and the source is deleted.
    !!If you attempt to increase the amount of text beyond 55K, you will find that the additional text is not accepted and should receive a warning. Save your work, and export the text to the clipboard for use in a work processor such as Write which supports large text blocks.

* Speed Keys:
The delete and backspace keys delete and remove text as you would expect. You can use Ctrl-Home and Ctrl-End to jump to top/bottom of text respectively. Shift-Ctrl-Home & Shift-Ctrl-End jump and select text from the current insertion point to the top or bottom of the text. Ctrl-Ins copies selected text to the clipboard, while Shift-Ins pastes from the clipboard.

* Adding Text:
You can add text by clicking an insertion point in the existing text, and typing. Hitting the Enter key ends one line and starts the next.

* Copy & Paste Buttons:
These buttons copy selected text to the Clipboard, and paste text from the Clipboard respectively. You can also use the speed keys for copy and paste mentioned above. If there are insufficient local resources for the paste operation, a warning message is put up to alert you that the paste failed.

* Append Button:
Select a block of text and append it to whatever text is already on the Clipboard. The total size of the new text must be less then 64K. You can use this feature repeatedly to rearrange text in almost any fashion.

UltraClip will let you open multiple edit windows, but only one for a given miniclip. This feature would, though, let you transfer text fragments from one text object to another (via the clipboard) with good control over where the text goes in the second window.

Files
Using .CLP Files
Ultraclip can save and read .CLP files that are compatible with the Clipboard Viewer application that comes with Windows. To save a regular miniclip as a .CLP file, highlight it by clicking on the title bar, and choose File/Save from the file menu. Supply a suitable file name, and your clip object is saved to disk. You can load one or more .clp files by choosing File/Open, selecting the CLP file type, and selecting the file/files you wish to load. Each clip object will be loaded into its own window with a thumbnail . You can copy it to the clipboard or hold for later retrieval (or print a thumbnail sheet).

To save an OLE miniclip as a .CLP file, copy it first to the clipboard and paste it as a regular miniclip which you can then save to disk.

Using .BMP, .PCX, and .WMF Files
UltraClip can read Win 3.0 style bitmap files(.BMP), .PCX version 5 files, as well as Aldus-style metafiles (the kind produced by Corel, Arts & Letters, and other draw programs). Choose File/Open, select the file type you are interested in, and highlight the filenames you wish to load.
In this situation, UltraClip does a bit more than just load the format into a miniclip. For .BMP files, both a memory DIB image and a bitmap & palette object are created. You can copy one or all to the clipboard as needed. PCX files are translated into a comparable DIB image, and a corresponding bitmap and palette are constructed. For metafiles, UC creates a Picture file as well as a DIB image and bitmap/palette pair. Thus you can use UC to rasterize metafiles and create bitmap objects from DIB and PCX images if needed.

Text Files
UltraClip will read text files off the disk and create a miniclip. File size is limited by available memory but editing capacity is much more limited. A text file can contain only printable characters plus CR, LF, Tab and EOF characters.

The Desktop
The menu options **Desktop/Save** and **Restore** let you save the contents of all existing miniclips to disk and restore them later. This includes OLE miniclips. You can specify the drive and save file name using the **Configure** option. Please note that the desktop save file can be very large if you have a number of large objects in memory - it consumes roughly as much disk space as the in-memory images. For example, if UltraClip has reduced your free global memory from 8MB to 5MB, you can expect the desktop save file to use roughly 3 MB of disk space.

You can also choose **Desktop/Save As...** and **Restore From...** to specify a new *desktop* file to save or restore. This allows you to build multiple catalogs of graphics and text, perhaps one for each project, and easily switch between them. Note that when you choose **Save As...** or **Restore From...** you are actually reconfiguring the save file name - thus it becomes the default for the next save or restore action.

Printing
Printing Thumbnails
UltraClip includes a simple print option that lets you print 'proof sheets' of the current thumbnails. In the Edit/Configure **options** dialog you can specify the grid used to position the printed thumbnails:

m Across by
n Down with a margin of
x Inches.

During printing, new thumbnails are created to optimally fill the available space with a caption built from the filename or module name.
Be patient when starting a print - creating a page with 24 or 30 thumbnails is quite system intensive considering the scaling and (in most cases) color conversions. Don't try a print without sufficient memory - the print manager can consume considerable disk and memory resources. The quality of the output depends somewhat on the printer driver, as does the speed of the operation.

Printing Graphics Directly from Disk
UltraClip will also directly print thumbnail sheets of graphics files residing on the disk without loading them simultaneously into memory. This feature allows you to print automatically thumbnails for a very large number of graphics images - such as might reside on a CDROM. Use the **File/Print from Disk...** menu option to choose the files you wish to print. You can select any bitmap, metafile, or .CLP file that UC can read. In addition to the file name, UC titles each page with the drive/subdirectory where the images were found and a sequential page number.

While much less memory intensive than printing the images from memory, you must still ensure that the system has sufficient resources.

General
Memory Checking
UltraClip will monitor the free global memory based on a limit you set in the **Edit/Configure** dialog. When the free memory drops below the specified minimum, you will be prompted before creating new miniclips. You can defeat this checking by setting the memory limit to 0.

Configuring UltraClip
You can configure UC by choosing **Configure** from the file menu, or right-clicking on the title bar with the mouse. In addition to setting the startup AutoPaste & AutoTrack mode, you can choose a new thumbnail size and printing setup and specify the desktop save file name. Your choices are written to **uc.ini** and takes effect with the next object pasted into UltraClip. Please note that existing thumbnails are not recalculated, so it is best to change that parameter at the beginning or end of a session.

You can temporarily toggle the AutoPaste and AutoTrack mode by right-clicking in the respective status windows. There are also menu options to change the current mode under the Edit menu.

The AutoSequence hook key should be a letter from 'A' to 'Z' or '0' to '9'. You can disable the hot key facility by deleting the existing entry and leaving it empty. This is not the same as putting a space in the field, which is itself a valid character.

OLE
UltraClip supports OLE 1.0 in two ways: regular and OLE miniclips. Regular miniclips neither embed nor link to objects, but let you store OLE data for embedding and linking in other applications. When you copy an OLE formats into UC, it captures the OLE data; when you restore them to the Clipboard, it places copies of the original data back on the clipboard in the same order and format. The goal is to transparently restore the state of the Clipboard. Of course, if you selectively choose formats to copy to the clipboard, the copy process will not match the original.

OLE miniclips actually create an OLE object managed and displayed by the OLE system libraries (and source application). Within UltraClip these OLE miniclips behave much like the other windows, although they are completely different internally. There are advantages and disadvantages to using OLE. The OLE miniclips display much move slowly, contain fewer formats and are tied to OLE applications. On the plus

side, they can be edited from within UltraClip, they require less disk space to store if linked, and they can be automatically updated by the source application.

UltraClip supports the OLE 2.0 clipboard formats which allow you to save and restore the OLE 2.0 data for use in other programs. UltraClip does not support OLE 2.0 miniclips - that is, if you embed an object in UC, it will be an OLE 1.0 style object regardless of the source program.

Manual Configuration Options
There exist several configuration options that you can use to override Ultra Clip's various safety warnings. You would enter one or more of the following lines in the **UC.INI** file (but not the braces):

```
[UC]
PromptOnClose=0                    {no nag exit from UltraClip}
PromptOnSave=0                     {no nag before overwriting desktop save file}
AutoRestore=1                      {automatically restore the desktop save file when UC starts}
```

Known Peculiarities
Occasionally AutoPaste may create multiple clip windows when only one is appropriate. Basically there are two situations: (1) the source application prematurely closes the clipboard before it is done copying all formats (2) you close the source application and it replaces the object on the clipboard with a new one. UltraClip tries very hard to avoid problem number 1, but a slow application can trip it up. This is frustrating - the only fix I know of is to turn off AutoPaste for that application or delete the unwanted (usually incomplete) clip windows. Problem 2 is unavoidable - the "duplicate" object is in fact slightly different from the first, although it may appear identical.
Micrografix Designer is unhappy with the MGX_... formats as restored by UltraClip, so they are suppressed even if present on the clipboard.
Frame by FrameMaker seems to use a non-standard clipboard format that fails to render properly in UC.
It is possible to create metafiles in advanced drawing applications (such as Corel Draw) that are simply too complex for Windows to render. Typically Windows crashes when trying to display these files. One possible workaround is to copy sections of the image to the clipboard and store them in UltraClip. If, for example, Corel warns you that a given image is 'too complex' for the clipboard, do not try to display the clipboard version - rather clear the clipboard and start again.
If you click a button and nothing happens, check to see that a clip window is highlighted. If not, click a title bar and try the button again.
If you save an object to disk which includes a 'bitmap' format, and then restore it to an environment running in a different graphics mode, the bitmap format will not display properly and may crash the program. This includes both .CLP files and desktop save files. This is unavoidable - the bitmap format is *always* device specific, and does not transfer between modes such as VGA and SuperVGA. The solution is to use the DIB format, which is device-independent and should display on any graphics adapter.
At least one version of the Diamond SuperVGA driver faults when executing StretchDIBits() with a 16x15 DIB (such as produced by WinWord 6.0 mini-icons). The only fix is to obtain an improved driver.
The UltraClip hotkey facility uses a system wide 'hook' that inspects every keystroke. There is a certain performance penalty with this kind of operation - although the rate of keystrokes is pretty slow in most situations. You can defeat the hotkey checking by configuring the hotkey to be a null - that is, highlight the current hotkey entry, and delete it with the delete key, then click OK. This will prevent UC from installing the keyboard hook.

Copyright
UltraClip is copyright Doug Overmyer, 1992 and 1994.

Distribution Terms
UltraClip is distributed as freeware - there is no registration fee. If you cannot obatin a copy from an electronic service, I will send one to you for $5 to cover shipping/media/handling.

   Doug Overmyer
   1817 San Pedro Avenue

Berkeley, CA 94707
USA

Disclaimer
UltraClip carries no warranties. You use it entirely at your own risk. You must determine its suitability for your system and needs.

Source Code
UltraClip is written in Borland Pascal and VC++.

Acknowledgements
I wish to thank the following whose many suggestions contributed materially to UltraClip.
Nord-Jan Vermeer
Nat Johnson
Christian Gerber
Earl Watt
Hartmut Bromkamp
Günter Herbig
I appreciate your help a great deal! Thanks also to Ron Burke (editor *Window/DOS Journal*) whose articles inspired drag and drop editing, and to Ron Gery (MS Developers Network) whose articles on palettes finally got through!

Problems, Suggestions
Please drop me a note on Compuserve (71021,2535) or Internet: overmyer@netcom.com

Author
Doug Overmyer
January 10, 1995
Notices:
Arts & Letters® is a registered trademark of Computer Support Corporation. CorelDRAW® is a registered trademark of Corel Systems Corporation. Micrografx is a registered trademark of Micrografx, Inc.
Notes:
Release 1.1 added OLE support, reworked entirely the storage of objects, display of metafiles, improved error recovery.
Release 1.2:implemented thumbnails, reworked owner-draw format , reworked large metafile display code, added various configuration options.
Release 1.3:added zooming of graphics, selective copy of formats, uniform tiling feature & better scrolling.
Release 1.4:reworked the array feature, added printing, reading .BMP and .WMF files, and reading/writing .CLP files and saving/restoring the desktop,revised autopaste logic.
Release 1.5:read .PCX helpfile;stay on top option;
Release 1.6:reworked status display;added autotrack clipboard;print from disk;scroll maxed windows,text manipulation functions,drag&drop interface;
Release 1.7:add real OLE client support;autosave option;more status windows; low memory check;
Release 1.7A:fix text file read for short files
Release 1.8: better sanity checking for bitmaps, error recovery; new desktop save/restore options; correct various anomalies;new ctl3d version;improved internal drag and drop response
Release 1.9:accept .wmf monochrome files;AutoSequencing and hook logic added;rev clipboard thumbnail code;F1 help accelerator;
Release 2.0:fix bugs, remove gif support, fix wmf palette code, text reads