Contents

DLL Master

Overview

DLL Master is a tool that lists all the modules currently loaded by Microsoft Windows and allows you a certain degree of control over the loading and unloading of specific DLLs.   Here is a brief summary of the features of DLL Master:

• Displays a list of all modules currently loaded in Windows' memory.

• DLLs can be loaded or unloaded from memory, or their use count altered.

• Specific versions of DLLs can be loaded from disk.

• Specific DLLs can be pre-loaded at Windows startup.

• The display list can be rebuilt at any time, such as before or after loading another application.

• The last several lists are saved, and can be viewed each in its own window.

• Any two lists can be compared to produce a list of changes.

• Lists can be printed, saved to disk, and restored from disk.

• Version information can be displayed for any executable containing a version resource.

These capabilities can be extremely helpful to Windows developers who may be working with multiple versions of programs and DLLs.   You might get lucky and find it possible to unload a DLL which was orphaned in memory when an application died in testing.   One can certainly raise the use count of a DLL so that it doesn't unload.   Network administrators and others also find DLL Master useful in debugging potential DLL conflicts on networked PCs.   The common question "Which version of this DLL got loaded, anyway?" is quickly answered by DLL Master.

(DLL Master is a trademark of Shaftel Software.   Windows is a trademark of Microsoft Corporation.)

Current List Window

When you start DLL Master, the main or Current Lists window appears.   This window contains the list of modules currently loaded in memory, along with descriptive information including usage count, full pathname, filesize, timestamp and internal version number.   You can view full version information by double-clicking on a module in the list, compare the list to others, print or save it.   The list may also be sorted by any of the columns, by double-clicking on the column header.

There is space below the list where you can enter a short description for your own purposes (say, if you will be refreshing the list several times and want to keep track of which list is which).

There is only one Current List window, although DLL Master will save up to nine prior lists in similar windows, referred to as Prior List windows.   Closing the Current List window via its System Menu merely causes it to minimize.

The Current List window is where you load or unload modules.   As you may know, "loading" a DLL that is already in memory merely increases the use count of the loaded copy.   You should also be aware that unloading or decrementing the use count of a DLL may cause Windows to die a sudden and painful death.   Do so at your own risk!

NOTE: loading, unloading or decrementing the use count of a DLL does NOT cause the list to be saved in a Prior List window, even though the list is rebuilt so as to properly reflect all changes (i.e. unloading one DLL may cause others to be released as well).   Issuing the Refresh List command is what causes the existing list to be copied to a Prior List window and replaced in the Current List window.

Prior List Window

When the Current List is refreshed, using the Action/Refresh List menu item or toolbar button, the existing contents of the Current List window are first copied to a Prior List window.   This window will appear minimized, and can be opened and viewed.   A Prior List window contains the same module list along with the same descriptive information.

You cannot load or unload modules from this list since it is no longer assumed to reflect the current contents of memory.   You can, however, view version information about the modules, compare the list to others, print or save it.   The list may also be sorted by any of the columns, by double-clicking on the column header.   You may delete a Prior List window by closing it via its System Menu.

Up to ten lists (including the Current List) can be loaded in DLL Master at any given time.   When this maximum is reached, each Refresh causes the oldest list to be deleted.   (Any number of lists may be saved to disk and later restored for viewing, comparing or printing.)

Lists that are saved to disk and later restored are restored into a Prior List window.

Comparing Lists

Any two loaded lists may be compared by selecting the Action/Compare Lists menu item or toolbar button.

The results of the comparison are displayed in a list in a third window.   The list has three sections:

- **<u>Added</u> section**

    Modules which appear in the later list but not the earlier one.

- **<u>Removed</u> section**

    Modules in the earlier list which do not appear in the later one.

- **<u>Use Count Changed</u> section**

    Modules whose use count changed.

This function is very helpful in determining the effect on the system of loading a particular application. Want to know why Windows starts swapping to disk when you load App X?   Compare DLL Master before and after lists to see just how many megabytes of EXE, DLL, FON, DRV and other files get loaded when App X is run!

NOTE: Actions taken within DLL Master itself will sometimes alter the use count of one or more Windows modules.   This appears to be unavoidable.   Also, DLL Master requires certain DLLs, and will load them if they are not already resident; these are DLLMSTRD.DLL, GRID.VBX, THREED.VBX and CMDIALOG.VBX.

Viewing Version and Header Info

This function displays all sorts of interesting internal information from the version resource (if one exists) and the New Executable Header of the executable file as it is stored on disk.

A version resource is a block of descriptive information about the executable which Microsoft recommends including.   It contains such things as the vendor's internal version number, name, target operating system, internal flags, comment, description, copyright and trademark notices.   The string information can be repeated in any of the languages supported by Windows (although we have found only English blocks in the U.S. version of Windows we use).   Not all vendors place a version resource in their executables, nor do they always fill all the fields when they do.   Non-standard version resources also exist--DLL Master will try to detect them but may also blow up while trying to read one.

The New Executable Header is a block of information placed by the linker at the head of every Windows executable file (just after the old DOS header for modules compiled for DOS-based Windows).   It contains a lot of very obscure technical stuff required by Windows when it loads the module into memory. DLL Master displays three tables of information from the header.   The Module Reference Table lists the names of other modules that are referenced by this module.   The Resident Name Table lists procedure names exported by the module, and has as its first entry the name of the module.   The Non-Resident Name Table lists procedure names exported by the module, and begins with a module description string. The difference between the two is that the name strings in the Resident Name Table are kept in memory to help speed up dynamic linking.   The name strings in the Non-Resident Name table are not kept in memory.

Filtering The Displayed List

You can limit the modules displayed in a list to particular extensions, using the checkboxes on the toolbar. Depending on which boxes you check, all or some of the following module types will be displayed: DLL, EXE, FON, DRV, VBX, and Other.

Filtering out modules using the checkboxes only affects which modules are actually displayed.   The complete list is still available should you decide to change the checkbox selections.

Unloading Modules

You can unload a DLL from Windows' memory using the Actions/Unload DLL menu item.   The Decrement Usage menu item will also cause an unload if the DLL's usage count is decremented to zero.

DLL Master does not actually unload a module--Windows does.   This is a logical consequence of the shared-code nature of dynamic link libraries.   A DLL is loaded by Windows when 1) another module issues a LoadLibrary API call to load it, presumably with the intention of utilization its functionality, or 2) another program or DLL is loaded which has the DLL named in its new executable header (see Viewing Version and Header Info), again presumably because it utilizes services provided by the DLL.   Once in memory, if a second module "loads" and uses the DLL by either of the two methods, it is given addressability to the same, single copy of the DLL; only the usage count of the DLL is increased for each "load".

Conversely, the usage count is decreased with each FreeLibrary API call, or by the termination of a task or unloading of a module which caused an implicit load as in #2 above.   This incrementing and decrementing of the usage count is the means by which Windows prevents the DLL from being unloaded while there are still callers around that expect it to be available, and yet is able to free it when all callers are done with it.   In theory, each load will be offset by an eventual unload, and the usage count will at some point reach zero.   It doesn't have to happen this way--obviously many modules will be kept busy and loaded the whole time Windows is running--but certainly for application-specific DLLs, it should.

Any error that occurs which prevents the usage count from being decremented can prevent the usage count from droppng to zero, and the DLL will be stuck in memory.   This is a real bother if it's a DLL you are in the process of developing: a bug in the DLL causes your test app to blow up, the DLL gets stuck in memory, and you can't replace it with a fixed version without restarting Windows first.   DLL Master and other programs of its kind attempt to take care of this by reducing the usage count to zero, at which point Windows will unload the DLL.   Of course, things may be so bad at this point that death will occur anyway....

Now, some modules in the DLL Master list are actually application executables (EXEs) and are associated with a Windows task.   Thus, if you load Notepad, NOTEPAD.EXE does not have the same status in the system as a DLL.   If DLL Master were to use a FreeLibrary call to get rid of NOTEPAD.EXE, an instant GPF would result.   So if DLL Master discovers that the module you have told it to unload is an app, it will display a dialog listing all tasks active in the system, and highlight the one it thinks matches the module.   If you click the Terminate button on the dialog, DLL Master will kill the task using the TerminateApp API call.

TerminateApp is effective but does not give the victim application a chance to clean up after itself, so memory or other resource losses can result.   Close apps normally if you can, use DLL Master if you can't.   And keep in mind that messing around with Windows this way always has its risks.

Configuring DLL Master

You have some control over how DLL Master looks when it starts up.   The Options/Configuration menu item opens a dialog window with several options for having the program remember how it looked when last run:

- **Remember module type checkbox states**

    This checkbox controls whether the toolbar checkbox states will be reestablished.   These checkboxes determine which module types (by file extension) will be displayed in the list.   See Filtering The Displayed List

- **Remember version info inclusion toggle**

    This checkbox controls whether the checked/unchecked state of the Include Version In New Lists menu item (on the Options menu) will be reestablished.   If this menu item if toggled on (checked), DLL Master will read the version resource of each module that has one and display the version number in the rightmost column of that module's row in the list.   It take somewhat longer to build a new list with this option checked.

- **Remember main window position**

    This checkbox controls whether DLL Master will open in the same state (NORMAL or MAXIMIZED) it was in when last exited.   If the state was NORMAL, then it will appear in the same position on the desktop.

- **Remember current window position**

    Similarly, this checkbox controls whether the state and position of the current list window will be reestablished.

- **Remember version/header info window position**

    This checkbox controls whether the position of the version/header information window will be reestablished.   This window is not resizable: because it takes up so much room, it cannot be accomodated inside the parent main window in VGA mode and so is opened as a modal dialog. Someday, when 640x480 is history, this may be changed.

- **Remember list column widths**

    This checkbox controls whether the width of the list columns will be reestablished, if you have changed them from the default.   The changes must be made on the current list window in order to be remembered, though they will be reflected on any prior list windows that are subsequently opened.

- **Remember confirm unload/terminate-task toggle**

    This checkbox controls whether the checked/unchecked state of the Confirm Unloads menu item (on the Options menu) will be reestablished.   If this menu item if toggled on (checked), DLL Master will ask for confirmation when you try to unload a DLL, decrement the usage count of a DLL, or terminate a task.   If it is toggled off, the action will be taken without asking for confirmation; the Unload DLL and Decrement Usage toolbar buttons will be colored red as a reminder that confirmation is turned off.

Configuration information is stored in DLLMSTR.INI in the Windows directory.   To return all selections to their default states, just delete DLLMSTR.INI when DLL Master is not running.   It will be recreated with the defaults the next time DLL Master is run.

Loading DLLs At Windows Startup

You can instruct DLL Master to automatically load one or more DLLs when Windows starts up. Simply place a DLL Master program item in the Windows Startup group, and pass it a text file as a command line argument.   The text file consists of a list of files like this:

C:\CSTUFF\THIS.DLL

C:\CSTUFF\THAT.DLL

D:\TEST\THEOTHER.DLL

If this file were named LOADLIST.TXT, the command line property of the item in the Startup group would look like this:

DLLMSTR.EXE LOADLIST.TXT

You may want to do this to be sure a specific version loads, or to raise the use count so a DLL is not unloaded until you want it to be.

Note that this will not work with DLLs that are loaded into memory before the DLL Master loader routine gets control.

Menu Selections

File Menu
Actions Menu
Options Menu

File Menu

The File Menu contains the following items:

- **Save List**

  This allows you to save the contents of any list window or compare window to a disk file.

- **Restore Saved List**

  This allows you to restore a saved list from disk back into a Prior List window.

- **Print List**

  This allows you to print any list.

- **Print Setup**

  This brings up the standard Print Setup dialog.   The main reason for this is to change the printer orientation to Landscape since the printed list is too wide for Portrait mode..

- **Exit**

  Exits from DLL Master.

Actions Menu

The Actions Menu contains the following selections:

- **Refresh List**

  This will rebuild the Current List based on a new scan of memory.   The previous list will be copied from the Current List window to a Prior List window first.

- **Show Lists**

  This displays a dialog box from from which you can open or bring to the front one or more existing list windows.   You can also select them from the Windows menu, or just double-click on minimized list windows.   The advantage of Show Lists is that part of each list window description is shown, making it easier to find the list you want.

- **Compare Lists**

  This opens a dialog box from which any two of the existing lists can be compared.   The results will be displayed in a new window.   See <u>Comparing Lists</u>.

- **Version/Header Info**

  This allows you to display the full version resource information embedded in a module, and also selected information from the module's New Executable Header.   A file dialog first appears so you can select which file to examine.   Version/Header information for any module in a list can also be seen by double-clicking the module's row in the list.

- **Task List**

  This opens a dialog that lists the tasks (applications) running under Windows.   You may select and terminate any task from this window, which is the same one that opens automatically if you try to unload a module in the list and the module is running as a task rather than as a DLL.   This menu selection is only available on the Current List window.   See <u>Unloading Modules</u>.

- **Load DLL**

  This item opens a standard file dialog from which you can select a DLL file to load.   Keep in mind that if the DLL is already loaded in memory, its use count will just be increased--it will not be reloaded from disk.   Only available on the Current List window.

- **Decrement Usage**

  When a DLL is highlighted in a list, selecting this item will cause its use count to be reduced by 1. If the count becomes zero Windows will automatically remove the DLL from memory.   Only available on the Current List window.

- **Unload DLL**

  When a DLL is highlighted in a list, selecting this item will cause it to be removed from memory. Only available on the Current List window.   See <u>Unloading Modules</u>

Options Menu

The Options Menu contains the following items:

- **Include Version In New Lists**

  Toggling this item on causes the internal version number to be included for the modules in the list. This version number is found in the version resource compiled into the executable by the module's creator (not all programmers or vendors include one).   Since the version resource must be read from the module as it is stored on disk, toggling this item on does increase the time it takes to build a new list.

- **Confirm Unloads**

  Toggling this item on instructs DLL Master to prompt for confirmation when you try to unload a DLL, decrement the usage count of a DLL, or terminate a task.   If this item is toggled off, the Unload DLL and Decrement Usage toolbar buttons will be colored red as a reminder that no confirmation will be sought before going ahead.

- **Configuration**

  This item opens a the Configuration dialog, which lets you customize how DLL Master will look when you start it up.   See Configuring DLL Master

- **Autoloading**

  This item opens a text window that explains how to use DLL Master to load specific DLLs when Windows is started.   See Loading DLLs At Windows Startup.

Registering DLL Master

License Agreement

Thank you for using DLL Master--we hope you find it valuable and want to add it to your Microsoft Windows 3.1 toolkit.   DLL Master is a shareware program written in Microsoft Visual Basic and is provided at no charge to the user for evaluation.   Feel free to share it with your friends, but do not give it away altered or as part of another system.   The shareware version of DLL Master is distributed as DLLMST32.ZIP or some similarly named file. If you distribute DLL Master to friends, associates, or to a computer bulletin board system (BBS), please distribute the ZIP file rather than the individual files for DLL Master. The zip file does not contain the required Visual Basic 3.0 runtime module VBRUN300.DLL. This module is available on many BBS's including Compuserve, and is included with the registered version of DLL Master.

REGISTRATION

If you as an individual find this program useful and find that you are using DLL Master and continue to use DLL Master after a reasonable trial period, please make a registration payment (see Registration Fee) to Shaftel Software.   The registration fee will license one copy for use on any one computer at any one time.   If used on a network, one copy must be purchased for each computer that will be using DLL Master.

Business, government, institutional and other organizational users of DLL Master must register and pay for their copies of the program within 30 days of first use or their license is withdrawn.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND MANUAL ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED.   BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.   GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT.   THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM.   ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

DISTRIBUTORS

Anyone distributing the shareware version of DLL Master for any kind of remuneration must first contact Shaftel Software for authorization, with details including price to be charged.   This authorization will be automatically granted to distributors recognized by the Association of Shareware Professionals (ASP) as adhering to its guidelines for shareware distributors, and such distributors may begin offering DLL Master immediately.   (However Shaftel Software must still be advised so that the distributor can be kept up-to-date with the latest version of DLL Master.)

Distributor advertisements, catalogs, etc must explain the shareware concept, and state that this program is not free, but is copyrighted software that is provided to allow the user to evaluate it before paying.

Registration Fee

If you as an individual find this program useful and find that you are using DLL Master and continue to use DLL Master after a reasonable trial period, please make a registration payment to Shaftel Software.

Business, government, institutional and other organizational users of DLL Master must register and pay for their copies of the program within 30 days of first use or their license is withdrawn.

The registration fee will license one copy for use on any one computer at any one time.   If used on a network, one copy must be purchased for each computer that will be using DLL Master.   For full licensing details, please see License Agreement.

The registration fee for a single license is $25.00 (plus $3.00 shipping and handling outside North America).   California customers, please add $1.81 sales tax ($2.06 in L.A. County).

Payment must be made by check or credit card (VISA, Mastercard, American Express or Discover) in U.S. dollars at time of order unless prior arrangements are made.

PAYMENT BY CHECK

If payment is by check, send your order to:

>   Shaftel Software

>   P.O. Box 12485

>   La Crescenta, CA 91224-5485

PAYMENT BY CREDIT CARD

If payment is by credit card, order from Public (software) Library by calling 800-2424-PsL or 713-524-6394 or by FAX to 713-524-6398 or by CIS Email to 71355,470. You can also mail credit card orders to PsL at P.O.Box 35705, Houston, TX 77235-5705.

The above numbers are for orders only.   PsL does not handle inquiries or provide technical support.

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, etc, must be directed to Shaftel Software as described below.

SUPPORT

Shaftel Software can be contacted by fax for support or with questions.   The number is (818) 957-7992. We can also be reached on Compuserve (72540,1462).   Or, you may reach us by mail at the above address.

Comments and suggestions for improving DLL Master are always welcome.

DLL Master Order Form

Please register me (us) as a licensed owner of DLL Master:

Name: _____

Company: _____

Address: _____

                      _____

City: _____ State: _____ Zip: _____

Country: _____

Date: _____

Fee per copy	$	25
CA Tax ($1.81 or $2.06)	$_____	
S&H outside N. America $3	$_____	
Check Total (U.S. dollars)	$_____	

   ( ) 3.5" disk   ( ) 5.25" disk

First acquired or learned about DLL Master from:

   ( ) Friend   ( ) Coworker   ( ) Catalog   ( )BBS:_____

   ( ) Other_____

                                                          07-3.3.0

Send to: Shaftel Software

               P.O. Box 12485

               La Crescenta, CA 91224-5485