

INITLoader

By Ian Hendry

New for 7.0b4 1) A bug was fixed where I did not initialize managers which broke auto-mounting on 7.0b2 and later . 2) Double clicking on a folder in the folder list, opens that Finder window . I like this feature, it lets you really see from which folders you will be loading extensions. 3) Help Manager resources have been added. 4) If an extension uses too much high memory (by moving bufPtr down), INITLoader will be blamed. INITLoader uses NO high memory . Sometime I may fix it later so the correct extension is blamed, but not for now .

What INITLoader does INITLoader has two main functions: 1) INITLoader makes aliases to extensions (also called "startup documents" or "INIT s"), cdevs and rdevs work as expected. System 7.0 does not (at this time) support running aliases to these items (the "Magic System Folder" does not even route them to the correct folders). 2) INITLoader lets you choose additional folders from which to load extensions.

Why INITLoader does Aliases to extensions and folders give you all the advantages of aliases. One of the coolest things is auto-mounting. This means you can specify a folder on a floppy and the system will ask you to mount the floppy when it resolves the alias. The same is true of network volumes. If you have (your extensions folder) an alias to an extension on an AppleShare® or FileShare™ volume, then at startup time you will get a password dialog (unless you created the alias as a guest) and that server will be automatically mounted and the extension will run. INITLoader can also automatically unmount any volumes that it mounts. This cleans up so you do not have a whole lot of extra volumes around when you get to the Finder . For those of you saying "neat hack, so what?" Here are examples. Suppose you are a network administrator and you always want your team to have the latest version of some extension. It is a headache for you to always have to find all the people and convince them to upgrade (particularly if they all have to do it at once). With INITLoader there are two solutions: you can distribute an alias to the extension on a server or you can distribute a copy of INITLoader with an alias to a folder on your server in it. The second solution allows you to add whatever extensions you want onto you team. For another example, suppose you are developing an extension and you want to test it on various machines. INITLoader allows you to load your extension from the network with out having to install on all those machines.

Notes INITLoader will not load multiple copies of extensions. INITLoader will not load any extension from a file with the same type/creator as one it

has already seen. If you have several copies of "Programmer 's key" for example, only the first one will load. INITLoader first examines the normal places in the System Folder from which extensions are loaded. After all aliases to extensions have been resolved and run, then INITLoader looks though folders that you selected. This means that extensions found in the System Folder take precedence (because INITLoader will load the each extension only once) over those in folders you select. INITLoader can run other copies of itself. If one INITLoader loads a second INITLoader then the second INITLoader will run and will load any extensions not loaded by the first one (**ie** those in folders specified to be loaded in the second INITLoader). When the second copy is done, the first INITLoader will regain control and continue loading other extensions. This only works if the second INITLoader is loaded by the INITLoader (that is, NOT by the normal extension loading done by the System). If you have an alias to an extension, you cannot stop it from loading by moving the extension itself (the alias will find it anyway).

T esting Use INITLoader at you own risk. I have run INITLoader on everything from a Macintosh™ Plus to a Macintosh IIfx with 20 Meg real RAM. It works with VM (I tried it a little) and in 32- Bit mode. I did not run it on a huge variety of extensions. Most extensions tended to fall into two categories: worked fine and didn't work with 7.0 at all.

Cautions and Things that go bump in the hard disk There are some extensions that do not like being moved around. I hope with System 7.0 developers begin to remove location dependence from their code. I figure the Soviets have abandoned socialism, Germany has reunited, why not go for broke and hope for good code from extensions? INITLoader depends heavily on System 7.0 features and will not work under System 6.0.x. Some extensions have support files. These extensions should support alias files via "ResolveAliasFile" for their data files. Until they do, you will have to have copies of their data files on the various machines. I wrote an extension to support alias files transparently and there were some problems with doing that solution so now we have ResolveFinderAlias. ResolveFinderAlias' predecessor appeared on the first System 7.0 Developer's CD and was later added to the System as a new toolbox call. Read about it in the Finder Interface chapter of Inside Macintosh V olume VI, use it in all your code that does not use Standard File. Make sure AppleShare loads before INITLoader (which normally happens due to their names). Otherwise aliases to items on network volumes may not load correctly . Some extensions try to re-open themselves later . Several extensions actually try to open themselves by name in the System Folder . Looking for yourself if kind of dangerous and just looking in the System Folder doesn't work with System 7.0 at all (much less INITLoader). If you really need to find your file again, create an alias to it under 7.0 and keep that around.

Future enhancements ***Support of cut, copy and pasting aliases between copies of INITLoader . More feedback on what INIT s are loading. Duplicate extension warnings (right now INITLoader just skips the duplicates -- it would be nice have an option that put up a dialog). Supporting arrow keys in the list. Watch for bufPtr violators.***

Technical Info Resource hunting you will find there are several changes you can make. First there is the list of folders that INITLoader will search for INIT s on you boot volume. The 'SLst' resource is a list of specialty folder types passed to FindFolder . INITLoader first searches the folders returned from that set of FindFolder calls. The 'TLst' resource is a list of file types that INITLoader will search for extensions (note that INITLoader assumes that the regular System will pick up any non-aliased extensions in the System Folder so extending this list will not affect which extensions are run in the System Folder). The "gVar" resource specifies global information. One field is the auto-unmount feature of INITLoader and the other is a flag not to load aliases to extensions found in the system folder . You should only set this flag if newer versions of the System decide to support aliases to extensions. There is a field for whether or not to enable double-clicking an item in the list to open Finder windows. There is an unused field for selecting the method of sending open events (keep this zero for now). At this time the 6.0.x glue is not here for any of the 7.0 calls I use but a later version may support the 6.0 world to some extent.

Trademarks AppleShare, FileShare, Finder , Macintosh and System 7.0 are all trademarks of Apple Computer . I don't know if I have to put these here, but what the heck.

Using INITLoader

Aliases to extensions in the normal places (System, Extensions, Control Panels folders) are run automatically . Selecting more folders from which to run extensions is done through the control panel interface. Holding down the mouse button during startup stops INITLoader from running. To add a new folder onto the list, click the "add" button. You can then select folders through a standard file dialog. This dialog is one area I wish I could improve (and I will if I have time). The "Select this folder" button selects the folder whose contents you are currently seeing (this works the same way as the MPW set directory dialog). For example, clicking it in the dialog above would cause the folder "LateSys" to be selected NOT "INIT s". To select "INIT s" you hold down the option key and the "Open" changes to "Select" (as I did for this screen dump) and you can choose "INIT s". I'm sorry in advance for any trouble this may cause. The new folder is added to the list that will be searched. When you add folders they get a default name which made from the path to that folder . Double-clicking an item in the list will

open the corresponding folder 's window in the Finder . If you want you can change the name with the "Rename" button. Select any of the items and you want to rename and click the "Rename" button. You can then edit the name. Clicking the "Default" enters the name based on the path to the folder . You can also delete any items in the list with the delete button. Clicking the "Unmount auto- mounted volumes" check box tells INITLoader to unmount any volumes mounted while running extensions. INITLoader will only unmount volumes that have no open files on them.