

# **NewIcons**

Nicola Salmoria Vedovatti

**COLLABORATORS**

	<i>TITLE :</i> NewIcons		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Nicola Salmoria Vedovatti	January 17, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>NewIcons</b>	<b>1</b>
1.1	NewIcons . . . . .	1
1.2	Introduction . . . . .	2
1.3	NewIcon Features . . . . .	3
1.4	Installation . . . . .	5
1.5	NewIcons . . . . .	6
1.6	DefIcons . . . . .	8
1.7	CopyNewIcon . . . . .	9
1.8	CreateDefaultIcon . . . . .	9
1.9	InjectBrush . . . . .	10
1.10	KillNewIcon . . . . .	11
1.11	PatchOpenWB . . . . .	11
1.12	Tips And Tricks . . . . .	12
1.13	Authors . . . . .	15
1.14	Acknowledgements . . . . .	15
1.15	Distribution . . . . .	16
1.16	History . . . . .	16
1.17	Future . . . . .	17

---

# Chapter 1

## NewIcons

### 1.1 NewIcons

```

                                The NewIcons Package
                                Version 2.0
                                by
~Nicola~Salmoria~
```

---

```
~Introduction~
```

```
~~~Features~~~
```

```
~Installation~
```

The Main Programs:

```
~NewIcons~
```

```
~DefIcons~
```

The Support Files:

```
~~~~CopyNewIcon~~~~
```

```
~~~~InjectBrush~~~~
```

```
~CreateDefaultIcon~
```

```
~~~~KillNewIcon~~~~
```

```
~~~~PatchOpenWb~~~~
```

```
~~Tips~and~Tricks~~
```

```
~~~~~Authors~~~~~
```

~Acknowledgements~~

~~~Distribution~~~~

~Version~History~~~

## 1.2 Introduction

This is the long awaited update to the original NewIcons GUI system. It offers several enhancements to the original set of programs, as well as several bug fixes. The history file is not entirely complete, but covers most of the enhancements which have been added to the package. ←

This, unfortunately will very likely be the last update of NewIcons by Nicola Salmoria. The package has been extensively beta tested by myself (Phil Vedovatti) and others and I have not found any bugs in the past 5 months of use. Fortunately, there is still the possibility of further enhancement to the package, and you can find out here.

Nicola has authorized me to release this package as an update to the original. ←

I have updated to documentation to reflect the changes in the package, as well as the installation script. I have also updated the Tips and Tricks section to answer more commonly asked questions.

For new users of this package, the introduction from the original documentation is here below. If you have never used NewIcons before, please read the documentation carefully...it will save you from any trouble you should encounter.

Introduction from the original NewIcons package:

This archive is not "just another set of icons". This is a completely new concept, which I hope will revolutionize, and push in the right direction, the idea of GUI on the Amiga.

Kickstart 3.0 was Commodore's first step towards ReTargetable Graphics. There isn't a device independent graphic engine yet, but there is a new pen arbitration system which allows applications to obtain a given color, or the one which is closest to it if there are not enough free pens.

It's simply ridiculous that, on a system capable of displaying a 256 color Workbench screen, and with a pen arbitration system like the one described above, icons are limited to 8 colors. Various solutions have been proposed, ranging from "use some colors and hope they are right" to "allocate some pens and reserve them for icon display", but they all miss the point, which is:

Why should I, the user, have to care about the icon palette?

Why should I set the system palette in a way decided by someone else

---

(the icon designer) and not the way `_I_` like it?

When you display a picture on the Workbench screen using MultiView, you don't have to care about the palette: the system chooses it for you. So why should one care about the palette used by icons? The system should take care of it, not the user.

Unfortunately, icons don't carry palette information. Therefore, I had to define a new standard for icon files, and write a system patch to make the system recognize them. Several goals had to be reached:

- The new standard should be backward compatible, that is users not running the patch program should see the standard 4-color icons.
- Manipulation of icons by users not running the patch program should not cause the new icon information to be lost.
- Remapping of the icons to the Workbench palette should be as fast as possible.
- Data should be compressed in order to reduce disk usage.
- The patch should work even on Kickstart 2.0 systems, with limited features.

All of the above goal have been reached. The new icon information is stored in the ToolTypes, so it is completely backwards compatible. A library, `newicon.library`, takes care of reading and writing the imagery in the ToolTypes; it also hides away the special ToolTypes so that when one asks for Information about an icon only the standard ToolTypes are shown. A patch program, `NewIcons`, intercepts calls to `icon.library`, redirecting them to `newicon.library`, so that Workbench and all other applications transparently get the new icon imagery.

## 1.3 NewIcon Features

### NewIcons

is a revolutionary system patch which enhances your Workbench and gives the user much more power and configurability over the appearance of the GUI (Graphical User Interface). The `NewIcons` system works transparently, and provides for a standard icon system that will look the same on any system, no matter what the palette is set to.

### DefIcons


complements the `Newicons` system. It works by recognizing a file's type, and instantly displaying the appropriate icon image for files without icons attached to them.

---

The system's features include:

- Allows up to 256 color icons on an AGA machine. (!)
  - Icon colors are color correct on an AGA system no matter what your palette is set to. They automatically appear in perfect color.
  - Default icons for dozens of file types automatically appear for files without icons (FAST!). Recognizes file types in a very efficient manner. If you click on a "fake" icon, your appropriate user-defined default tool will appear in the "Execute command" text field automatically!
  - Make a brush, convert it to an icon. It will appear on your workbench in the exact proper colors you created it in, no matter what your palette is set to. Imagine making gorgeous icons from your digitized pics, in more than 16 colors!
  - Icon images are compressed to maximize disk space. They are uncompressed faster than you can say "NewIcons"!
  - Also KS 2.04 compatible (16 color icon limitation).
  - Fix that ugly Cross-DOS icon to look like a standard NewIcon.
  - Unique icons for RAM-Disks and CD-ROM's that don't have icons.
  - A complete set of beautiful, standardized Workbench icons, drawn by artist Roger McVey.
  - Comprehensive set of utilities to allow you to manipulate your NewIcons.
  - FREEWARE! Why would anyone want to pay cash for icons?
  - Very stable. Beta Tested/Updated for over a year.
  - NEW: Dithering of icon images is now selectable for users with fewer available Workbench colors. This is turned off by default.
  - NEW: Icons can be given a single tooltype, which will tell NewIcons to display a NewIcon image from elsewhere on the system. This feature has some caveats, but introduces a huge number of neat possibilities for configuring your Workbench, and saves on disk space and chip RAM. Imagine having your Workbench icons appear randomly different on EACH reboot!
  - NEW: The source for deficons.prefs has been included, so any enterprising programmer can add their own file identification types to the system!
  - NEW: Many bugfixes to the NewIcons/Deficons system.
-

## 1.4 Installation

To install the package, just double click on the "   
Install\_NewIcons"  
icon.

Make sure you have some disk space on your SYS: partition.  
You should have at least 200K available if you install the entire  
package. If you don't have enough room on your partition and  
you still want to install the package, see the  
tips~and~tricks  
section

to learn how to make more room for the images. Alternatively, you  
can install

NewIcons  
without the  
DefIcons  
files.

You will be prompted as to which parts of the distribution you want  
to install. For maximum performance, it is suggested that you install  
both NewIcons and DefIcons.

The installer script will install the icon images on your system. The  
tooltypes of your icons will not be touched by the installation script,  
and the original icon images will not be altered. You will not see the  
new icon images until the NewIcons program is run. The installer script  
will launch NewIcons immediately, so you can view the new images right  
away. Since your original icon images are not actually deleted, it is  
easily possible to restore the icons to their original appearance, so  
don't be intimidated by the installation process.

The NewIcons program will be copied to your c: directory, and  
newicon.library will be put in your Libs: directory. You'll  
be given the option of where you would like to put the other icon  
utilities. If you choose to install DefIcons (highly recommended),  
the program will be copied to your c: directory and the default icons  
and brainfile will be put in ENVARC:.

The program

PatchOpenWB  
will be copied to c:.

Two short commands will be added to your s:user-startup file to launch  
NewIcons and DefIcons whenever you boot your system.

Kickstart 3.x users! IMPORTANT:

-----

You need to add a line to your s:startup-sequence to fix a bug in  
Kickstart 3.x. Add the following line to your startup-sequence,  
before IPrefs and after SetPatch:

C:PatchOpenWB

Users of 2.04 need not worry about this step.

---



-----

Once the installation is complete, reboot your computer and you're ready to have the best looking GUI on the block!

The installation program only installs images for the standard WB, so some files will likely not be "NewIconized" by the installer script. Don't worry; you can add images to these icons later using the

CopyNewIcon  
utility supplied in this archive.

NewIcons works by installing compressed images of the new icons into tooltype lines of the .info files. Once NewIcons is launched, you will not see these tooltypes, so they will not interfere with your work. If for some reason you are not happy with NewIcons, you can use the utility

KillNewIcon  
to remove these images from the .info  
files, restoring them to their original appearance.

The big advantage of this system is that when the NewIcons program is run, you see the new icon images. When NewIcons has not been started, you will see the boring old icon you viewed before you received this package. Therefore, icons that are transferred to users not running NewIcons will still be visible as a plain, old-style icon.

## 1.5 NewIcons

This program is the heart of the package, and is launched from your user-startup. It accepts two parameters, and its template is as follows:

```
PRECISION/N,DITHER/S
```

The lower the PRECISION parameter, the greater the precision when picking colors to remap the icons. The default value is 16. Zero is very high precision, -1 is the highest.

NOTE: If you are running Kickstart 2.0, this parameter does nothing.

In this version of NewIcons, dithering is now, by default, off. This generally results in better looking icons for AGA systems, since the icons appear exactly as they were created. For users with a limited number of available colors on their workbench (such as those with backdrops with lots of colors, or a 2.04 system), you can try the DITHER option. For most systems, keeping the dithering off will usually be the best choice.

Examples:

```
newicons          (sets the precision to 16, no dithering)
newicons -1 DITHER (sets the precision to highest, and will dither)
newicons DITHER   (sets the precision to 16, will dither)
```

The idea behind NewIcons is that we shouldn't need a fixed color scheme for icons anymore. NewIcons will automatically remap icon images using the colors available on the Workbench screen. A specific palette is therefore not required (icons look reasonable even on a 2-color Workbench) but it is suggested to obtain best results. If you use a Workbench with  $\geq 64$  colors, you can ignore the palette completely. The file "palette.preset" is the palette I use on my Workbench. Note I have changed the pens used by bright and dark edges to have smoother button gadgets, and that color 3 is the same as color 0 so that the shell cursor is of the correct color. This palette uses 7 out of the standard 14 colors used by the NewIcons. The remaining 7 are dynamically allocated at run time.

Workbench 2.0 users:

If you are running Kickstart 2.0, you will have to set up a suitable palette. The colors used by the supplied icons are:

|             | Dec |     |     |
|-------------|-----|-----|-----|
|             | R   | G   | B   |
|             | --- | --- | --- |
| Black       | 0   | 0   | 0   |
| Gray1       | 2   | 2   | 2   |
| Gray2       | 5   | 5   | 5   |
| Gray3       | 8   | 8   | 8   |
| Gray4       | 10  | 10  | 10  |
| Gray5       | 12  | 12  | 12  |
| White       | 15  | 15  | 15  |
| Green       | 2   | 8   | 2   |
| Yellow      | 13  | 11  | 3   |
| Red         | 13  | 2   | 2   |
| Purple      | 9   | 4   | 10  |
| Light Brown | 12  | 9   | 6   |
| Dark Blue   | 0   | 5   | 9   |
| Light Blue  | 5   | 8   | 11  |

Of course you can sort them anyway you like, since NewIcons will pick the correct colors from the above palette for you.

A palette for 2.0 is supplied in the Extras drawer of this distribution.

In this version, there is a new tooltype feature. Any icon that does not have NewIcon image data in it can be given a single tooltype to cause NewIcons to "borrow" an image from another NewIcon icon. To use this feature, you simply add the following tooltype to the icon:

```
DEFAULTIMAGE=<location of image to "borrow">
```

For example, you could direct all your drawer icons to a single image on your harddisk or RAM: disk. Once this is done, you can change the image of all your drawers by changing the single "loaner" icon. You can read more about using this feature

here

## 1.6 DefIcons

The DefIcons patch will magically transform Workbench's silly fake icons (displayed when you pick the menu item Window/Show/All Files) into more meaningful ones. The program not only recognizes files such as projects and tools, but identifies various other file types:

- IFF, JPEG, ANIM, GIF and PNG files
- 8SVX, SMUS, DCMS, Med, and PT/ST mods
- Handlers, Kickfiles, and Filesystems
- ASCII, TeX, AmigaGuide and Postscript files
- Source codes of various languages
- Keymaps, Datatypes, Libraries, and Printer Drivers
- Various crunched filetypes
- Bitmap and Outline Fonts

...and several more.

Just like any "fake" icon, you can't double click on fake icons to start the associated tool immediately, but if you have designated a default tool for the deficon (located in ENVARC:sys), the "Execute Command" text field will automatically have the default tool entered for you. This feature is very handy for things like text files and picture files that don't have icons attached to them.

Additionally, you can transform a fake icon into a real icon using the

```
CreateDefaultIcon
utility.
```

DefIcons has a brainfile called deficons.prefs, which is copied to ENVARC: by the install\_NewIcons script. Unfortunately, it is currently not possible to easily edit the brainfile, so you can't add your filetypes, but the source code for deficons.prefs is included. With a little programming knowledge, you can create your own deficons classes. When you install NewIcons, the Installer script copies the default def\_XXX icon images to ENVARC:. These icons are used by DefIcons.

Apart from fake icons, DefIcons has another wonderful feature: if a volume doesn't have a disk.info icon, it will replace the standard def\_disk.info icon used by Workbench with a user-defined one. For example, it will use def\_CD0disk.info for a CD in CD0:, and def\_RAMdisk.info for the Ram Disk. If you did the old 'copy ramdisk.info to ram:disk.info' trick, you can remove that command from your startup-sequence.

DefIcons works well with multiple file systems and MFS. When a disk without disk.info is inserted in DF0:, it first searches for an icon named def\_DF0disk.info; if it can't find it, it looks for an icon named after the file system (def\_DOS0disk.info, def\_MSD0disk.info, def\_PFS0disk.info...) so you can have different icons for different

file systems. Latest versions of CrossDos use an icon of their own for MS-DOS disks. Since the icon is really ugly, you may want to use your `def_MSDDisk.info`. To do that, load `L:crossdosfilesystem` into a hex editor, search for the string `"disk.info"` and replace it with something else, for example `"dusk.info"`.

The disk trick will also work for non-DOS disks. In this case, the basename is the one displayed by Workbench under the icon. For example if Workbench shows `"DF0:NDOS"`, the icon named `def_NDOSdisk.info` will be used. `DF0:????` is a special case: `def_BADdisk.info` will be used.

DefIcons uses a tree structure when identifying files. For a JPEG picture, for example, DefIcons will search in order first for `def_jpeg.info`, `def_picture.info` and `def_project.info`. IFF files are special, and DefIcons will search for `def_XXXX.info`, where `XXXX` is the FORM type of the file. If it can't be found, `def_iff.info` is used.

For archived files there are three families: `def_archive` (`lha`, `zip`, ...), `def_filearchive` (for `uuencode`, `compress`, `gzip`, ...) and `def_diskarchive` (DMS). These families can have sons (i.e, `def_lha`, `def_dms`, etc) if you create and add them to your `ENVARC:sys` directory.

## 1.7 CopyNewIcon

`CopyNewIcon` will likely be the most used of the support utilities.

This command is used to copy the NewIcon image from one icon to another. You can use it to add NewIcon images to your other programs/files. ONLY the NewIcon image is copied. The standard image and the other properties of the destination icons are not affected.

You'll need this command to use the icons contained in the Icons/Misc drawer. DO NOT USE THE USUAL 'COPY' COMMAND TO COPY THOSE ICONS. You should consider the Icons drawer as a sort of "catalog" of images, which you can pick IMAGES, and not true icons, from.

From CLI:

```
Copynewicon srcicon desticon1 [desticon2...]
```

From Workbench:

```
click on CopyNewIcon
shift-click the icon you want to copy from
shift-click the icons you want to copy to
shift-double click the last icon
```

You can also add this command to the Tools menu using ToolManager or ToolsPrefs.

## 1.8 CreateDefaultIcon

This will convert a fake icon to a real icon. Of course it is ←  
much more  
useful if  
DefIcons  
is running.

It also replaces real icons with standard icons, that is with the ones which would be used as fake icons if there wasn't a real icon. I hope I've been convoluted enough to confuse you ;-)

Anyway, the interesting part is that if you use this command on a real icon, the original properties will not be lost (position, drawer dimensions, tooltypes, default tool - unless the def\_icon has a default tool by itself). This command is VERY handy to quickly replace all the drawer icons in your hard disk.

Usage is the same as  
KillNewIcon  
:

Click on CreateDefaultIcon, then shift-click on the icons you want to make into a real icon. Double-click on the last icon selected. If you have a standard default tool for certain files (like Multiview for your AmigaGuide® files), you can add that tooltype to your default icon (Envarc:sys/def\_guide.info) and any AmigaGuide® files you use CreateDefaultIcon on will have the correct default tool associated with it. This technique is useful for picture and sound files, and ASCII text files as well.

## 1.9 InjectBrush

This command is used to create a NewIcon. It can be used only from CLI. It takes the name of the icon to NewIcon-ize and the names of two ILBM brushes. The brushes should be 36x40 in size, and use the standard palette. In the Extras/BUILDING\_BLOCKS directory you find some standard elements to use in creating NewIcon images.

Template:

```
ICON/A,NORMALIMAGE/A,SELECTEDIMAGE,FORCE/S
```

Example: `injectbrush myicon.info brush1.iff brush2.iff`

You can leave off the ".info" extension if you like. Injectbrush is smart enough to know you mean the icon file, not the file itself.

Note that the original image is not removed from the icon. It stays there so that users not running NewIcons can see an old-style icon.

Color 0 of the brush used to create a NewIcon is always considered the transparent color. That is, it will always be remapped to color 0 of the screen.

The newicon.library does not optimize the color data. Thus, if you use

---

an 8 bitplane brush to create an icon, and it uses only 16 colors of the 256 available, the icon created will be larger than if you used a 4 bitplane brush. The best thing to do is to always work on a 16 color screen, using the `building_blocks/platform` provided to start from.

`Injectbrush` expects the brushes to be of the standard size and color palette. You will be prompted if the brushes are not 36x40, or if the palette colors are non-standard. This feature can be overridden by using the `Force` option.

## 1.10 KillNewIcon

This is the complementary of `InjectBrush`. It removes the `NewIcon` and restores the icon to its original image.

From CLI:

```
Killnewicon icon1 [icon2...]
```

From Workbench:

```
click on KillNewIcon
shift-click the icons you want to remove the NewIcon from
shift-double click the last icon
```

You can also add this command to the Tools menu using `ToolManager` or `ToolsPrefs`.

## 1.11 PatchOpenWB

Kickstart 3.x has a serious bug concerning Workbench screen opening. If the Workbench screen is closed and a program causes it to be opened, either by calling `OpenWorkBench()`, or `OpenWindow()`, or `LockPubScreen()`, the system may hang. Specifically:

- If the program has priority less than 0, it will always hang.
- If the priority is greater than 0, it will never hang.
- If the priority is 0, it may or may not hang; it depends on when task switching happens.

This program is a workaround for the bug. It will intercept `OpenWorkBench()`, `OpenWindow()` and `LockPubScreen()` calls and temporarily raise the priority to 1, thus avoiding the deadlock. This patch is necessary when running `NewIcons`, otherwise the system will always hang when the Workbench screen is closed and reopened.

Users of Workbench 2.0 do not need to use this command.

To install the patch, just execute the program `PatchOpenWb` in your startup-sequence, preferably before `IPrefs`, and after `SetPatch`.

---

## 1.12 Tips And Tricks

Here are some tips to help you make the most of the NewIcons system:

Q: When I ran the installer script, it didn't convert every single icon on my hard drive. Why not?

A: Since everyone's system can be configured differently, the installer can't possibly configure for all setups. The installer is designed to install the typical Workbench System file icons. To convert the rest of your icons, use the CopyNewIcon utility to copy images to other icons.

Q: I used CreateDefaultIcon to convert my icons into NewIcons. Now my tooltypes and default tool settings are missing from the icons. Help!

A: Go back and read the documentation. If the default icon in Env:sym associated with the icon you created has a default tool setting in it, it will be used, as well as the tooltype settings. For general use in converting existing icons to NewIcons, use CopyNewIcon instead. CreateDefaultIcon is for adding icons to files that don't have .info files!

Q: With 2 images in each icon, my icon sizes are a bit bigger than standard icons. Is there anything I can do about this?

A: If you decide that you really like the NewIcons system, and make it a permanent part of your Workbench, you can make the standard icon images very small, i.e. a few pixels large. This is done quite easily with the included utility "Killoldicon", by Michael Barsoom. This changes the old icon images permanently, so don't do this procedure unless you are sure you won't need them back. To use, simply shift-click on killoldicon, then shift-click on the icons you wish to convert. While holding shift, double-click on the last icon to convert. Since NewIcon images are compressed, the resulting final icon can be considerably smaller than a standard icon with the equivalent number of colors. One drawback of this method, however is that if you share these icons with people not running NewIcons, they will only see the tiny pixel icon.

Q: Can I run Deficons without running NewIcons?

A: Absolutely, if you only want to use the default icon feature. In this case you will only have the standard icon images to view, however. NewIcons can be run without Deficons also, although you won't get default icon images for "fake" icons.

Q: Sometimes when I change screenmodes, the colors of my NewIcon appicons don't look right. What's going on?

A: Intuition doesn't reset the colors properly for appicons, unfortunately. You'll have to switch back to the screenmode you were using, or restart

the program that created the appicon. Sorry, this is a fault of the OS.

Q: My icon colors don't look the right colors.

A: The NewIcons system dynamically allocates colors as needed when an icon appears on the Workbench. Most likely if you are having a problem with the icon colors, you don't have enough free colors available on your Workbench screen. Unless you are using a back-drop with a large number of colors, 32 or 64 colors is usually plenty with the supplied icons. Increasing the number of bitplanes in the screenmode settings should fix this problem.

Q: I want to make my own icons that are different sizes and colors from the supplied icons. How can I get injectbrush to work with the brushes I use to make the images?

A: All the supplied icons use the same 14 color palette to preserve the number of available colors on the Workbench. If you choose, you are certainly not limited to these colors and you can make icons as complex as your heart desires. Icons also do not have to be the 36 x 40 pixel size either. Using the "FORCE" option in injectbrush will allow you to use nonstandard palettes and sizes (up to 93 x 93 pixels). Make sure your brushes are the same size and palette, or injectbrush won't allow you to make them into icons.

Note: Although you can create up to 256 color icons, this is not really recommended, since you will very likely lock all the available colors for the workbench, and programs like Multiview will not be able to dynamically allocate the correct colors for viewing pictures and so forth. In addition, icons with large palettes take up a larger amount of disk space.

Q: IconEdit doesn't work with my NewIcon images. What gives?

A: Do NOT use standard icon editing tools to manipulate NewIcon images. Some programs could cause serious damage to the icon data, saving the NewIcon image as an old-style image, therefore losing palette information. The best way to create NewIcon images is to use a paint program; create a brush, and then use injectbrush; there are NewIcon editors out there as well, such as Iconian.

Q: I prefer the magic workbench style of icons. Can I get the benefits of using NewIcons with mwb?

A: Absolutely. Just convert the images to brushes and change them into NewIcons with injectbrush. From now on you'll never have to worry about what your palette is set to. There is also a utility to convert mwb icons to NewIcons available on Aminet.

Q: If I want to make my own default disk icons, do I have to make them a disk type icon?



A: No. Deficons can use drawer type icons for disks or drawers in most cases. It will also use any type of icon for the "kick" type icons. Therefore, if you make new default icons for your ENVARC:sys drawer, you don't usually have to convert them to a disk or kick type icon. This applies to the custom icons only, such as def\_RAMdisk, etc. The standard def\_disk.info and def\_kick.info icons still must be of the proper type.

Q: When I double click on an iconless file, sometimes the "Execute Command" gadget doesn't get filled in with the default tool like it's supposed to.

A: This is due to Intuition, not Deficons. If you open a drawer and click on an iconless Amigaguide file for example, Multiview should be selected in the "Execute Command" field. However, if you open a second drawer or click on other icons in a different drawer than the first, then try to open that same AmigaGuide file, the default tool will not appear. Sorry, there is no easy solution to this problem; it's a fault of the OS.

Q: I really don't want a great looking Workbench...how can I deinstall it?

A: There is a script in the archive that will deinstall the package for you. To remove NewIcon images from icons by hand, just run KillNewIcon on them.

Q: How do I use the DEFAULTIMAGE tooltype?

A: Use it with icons that have NO NewIcon image data in them. For example, to make all your drawers use the image of an icon called Work:MyDrawer.info, add this tooltype to your non-NewIconized drawers:

```
DEFAULTIMAGE=Work:MyDrawer.info
```

Now, you can change the look of ALL your system drawers just by changing the MyDrawer icon's image. Pretty cool! You could even set up your user-startup to randomly copy a different image at each bootup to Env:sys, and make your Workbench look different every time you start your computer!

Using this feature slows down loading time slightly, because two icons have to be loaded (the main one and the one whose image is to be borrowed), but has several advantages, especially with icons which are used frequently, like drawer icons:

- 1) It saves disk space, because the image is stored only once.
- 2) Saves memory when the icons are on screen, because the Tooltypes containing the compressed image are loaded, decompressed, and freed, whereas in the normal method they stay in memory until the icon is freed.
- 3) If you get bored of an image, just replace the default one, and all the others will change automatically!

Remember that the image stored in the icon has precedence. Therefore, you have to "killnewicon" a NewIcon to have the DEFAULTIMAGE ToolType work.

---

WARNING!!!: This trick is recursive, that is, if DEFAULTIMAGE points to an icon which has a DEFAULTIMAGE itself, a third icon will be loaded, and so on. Pay attention not to create loops which refer back to another icon. If you do, you will likely meet the big GURU. You have been warned.

Q: How can I make my NewIcon icons into brushes for editing?

A: There's a utility on Aminet (gfx/conv/newicontobrush) which will do this for you.

Q: Where can I get more icons in the isometric/McVey style?

A: There are loads of them on Aminet (well over 1000 images) in the pix/icon directory. Some of the best ones are McVey's NewIconsXTRA and NI-V2 packages, and the AESicons series by Phil Vedovatti (IMHO!).

Q: Are there any backdrop sets for NewIcons?

A: Yes! Look in the pix/icon directory on Aminet. Roger McVey has uploaded literally hundreds of NewIcons compatible backdrops. They look great!

## 1.13 Authors

The programs were written by

Nicola Salmoria  
Via Piemonte 11  
53100 Siena ITALY

E-Mail: MC6489@mclink.it

The icon artwork was done by

Roger McVey

E-Mail: r.mcvey@genie.geis.com

The installer script, documentation, and the archive layout was done by

Philip A. Vedovatti

E-mail: vedovatt@u.washington.edu

## 1.14 Acknowledgements

---

Thanks to Chris Green, whose chunky to planar routine was based upon. (Chris based his routine on one by Richard Addison).

Thanks to Federico Giannici who pushed me to make the color remapping faster, and faster, and faster...

Thanks also to Michael Barsoom, who kindly allowed the inclusion of his program, "killoldicon" in the package.

A special thanks to our friends who transcribed the installer scripts:

Eric Sauvageau for the French version, Giacomo Magnini for the Italian version, Volker Schleifstein for the German version.

And thanks to all the NewIcons fans out there!

## 1.15 Distribution

This software is FreeWare. It is freely distributable as long as the archive remains intact, and only a nominal fee is charged for its distribution.

This software is provided "AS IS" without warranty of any kind, either expressed or implied. By using it, you agree to accept the entire risk as to the quality and performance of the program.

## 1.16 History

Version History:

V1.0:

Original public release.

V2.0:

In the previous version of the docs I forgot to mention that the chunky to planar routine was based on a routine by Chris Green (who based his routine on one by Richard Addison).

fixed bug in DefIcons, was causing "plase insert volume dir c" after doing execute command - dir c:

fixed bug in DefIcons, would hang if it couldn't find deficons.prefs

fixed bug in DefIcons, was causing a disk icon to appear for files called "disk".

fixed bug in DefIcons, would show a drawer as disk icon for disks containing a directory called "disk" (which is a bad idea anyway).

The GetDiskObjectNew() patch now returns NULL if the given name doesn't exist

---

(like the standard function)

now does dithering (optional)

some file types added to deficons.prefs.

bug fix: createdefaulticon now preserves the tooltypes of tool icons and of project icons whose default icon have no default tool.

bug fix: newicons would check the argarray after FreeArgs(). Thanks to Nicola Soggia for discovering it and not reporting it. Thanks to Moreno Magnani who did report it.

bugfix, arj files were not recognized

bug fix: DefIcons now uses Stricmp() to avoid problems with international sorting

DEFAULTIMAGE special ToolType added

Fixed installer: no longer copies over the original def\_#? icons in Envarc:sys. It now just copies them to a safe place.

Localized installer and documentation files added

Replaced "ReplaceImage" utility with "KillOldIcon", since ReplaceImage required NewIcons not to be running to use.

Changed a few of the icons from the original distribution archive

## 1.17 Future

First, the bad news. Nicola has pretty much left the Amiga platform. Although his Amiga currently sits on his desk, his programming endeavors are now concentrated on the PC. As a result, this release version of the executables will very likely be the last one from him. The Amiga community has benefited greatly from Nicola's programs (Hex, MFS, NickPrefs, StatRAM, NewIcons/DefIcons, etc., etc.) and his programming efforts will surely be missed by many. I don't want this to sound like a eulogy, since he is surely alive and well, and I wish him the very best. It's been a real pleasure being a part of the NewIcons effort, and I thank Nicola for allowing me to be involved in it.

Now for the good news. Nicola has released to me the source code for deficons.prefs, and I have included it in this archive for your programming pleasure. This should allow further expansion of the DefIcons portion of the package in recognizing future filetypes, such as lzx packed files, more sound and graphic types, and so on. If you create an updated brainfile, please send it to me; or if some people out there are interested in forming a collaborative effort, I'd be happy to help coordinate things.

Also in this archive are the includes for the newicon.library, which opens up many other possibilities for NewIcons compatible programs.

---

You will see future icon releases as well. My own AESicons packages are now approaching 500 icons, and there are several other icon image packages out there as well. Roger McVey has recently released another great set, as well as a huge collection of NewIcons compatible backdrops. NewIcons is by no means dead. Like the Amiga, NewIcons lives on.

Enjoy!

Sincerely,

Phil Vedovatti

---