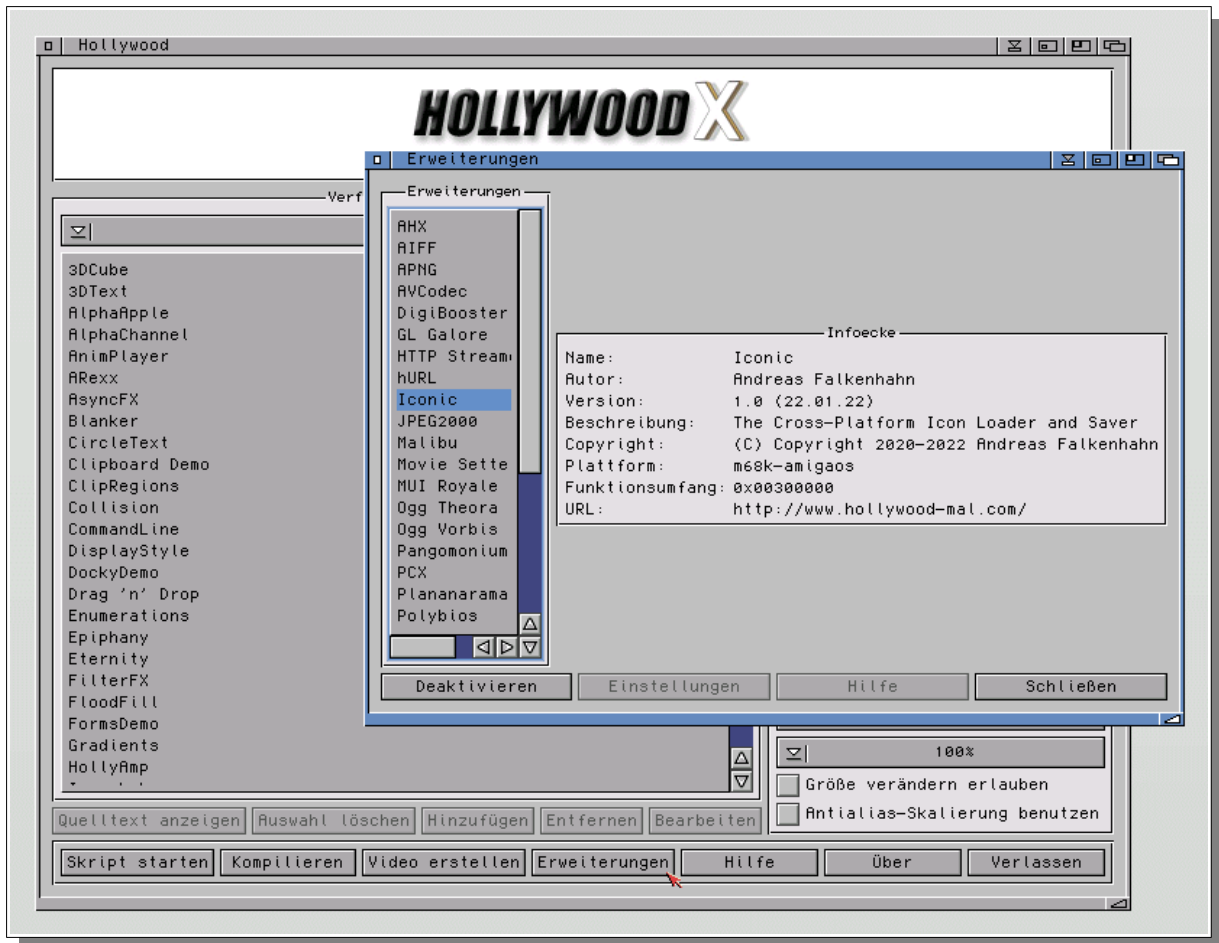


ANDREAS FALKENHAHN

Hollywood

ICONIC PLUGIN SHORT MANUAL



VERSION 1.0 (22-JAN-22)
FIRST RELEASE

Contents

| | |
|---|---|
| Abstract..... | 4 |
| Requirements..... | 4 |
| Loading icons..... | 5 |
| Example 1 – Display Icon..... | 5 |
| Example 2 – Get Icon States..... | 5 |
| Example 3 – Display Icon ToolTypes..... | 6 |
| Saving icons..... | 6 |
| Icon format information..... | 7 |
| History..... | 8 |
| Copyright..... | 8 |

Abstract

Iconic is the ultimate icon loader and saver plugin for Hollywood. It will make all your icon dreams come true because it can load and save a large variety of different icon formats. Currently, the following icon formats are supported by Iconic:

- x AmigaOS 1.x style icons
- x AmigaOS 2.x/3.x style icons
- x AmigaOS 3.5 icons (aka GlowIcons)
- x AmigaOS 4.0 icons
- x macOS icons (*.icns format)
- x MagicWB icons
- x MorphOS/PowerIcons icons (PNG)
- x NewIcons
- x Windows icons (*.ico format)

Of course, Iconic also supports loading and saving of the metadata found in the Amiga icon formats like tooltypes, icon position, drawer view mode, icon type and so on. Thanks to Hollywood 9's new support for palette images it is also possible to load the icon palette, set a transparent pen and access the raw palette pens used in the icon image data. Of course, Iconic doesn't only support palette-based icon images but also icons that contain true colour images with alpha channel. Several of the icon formats listed above also support palette and true colour images within the same icon file and several different icon sizes within the same icon file. All of this is supported by Iconic as well making it the ultimate icon loader and saver plugin for Hollywood.

Requirements

Iconic requires Hollywood 9.0 or better. It's also recommended to install the JPEG 2000 plugin for Hollywood because the macOS icon format (*.icns) stores some images in the JPEG 2000 format and Iconic will only be able to load them if the JPEG 2000 plugin is available.

Loading icons

To activate Iconic, simply add the following line at the top of your script:

```
@REQUIRE "iconic"
```

Then you can load and save icons using the functions from Hollywood's icon library. Here's how to load and show an icon:

```
LoadIcon(1, "Monkey_Island.info")
ConvertToBrush(#ICON, 1, 1)
DisplayBrush(1, #CENTER, #CENTER)
```

Note that icons can contain images in multiple sizes and color resolutions. To find out the number of images in an icon, just query the `#ATTRNUMENTRIES` attribute. The following code iterates over all images in an icon and draws them:

Example 1 – Display Icon

```
LoadIcon(1, "explorer.ico")

For Local k = 1 To GetAttribute(#ICON, 1, #ATTRNUMENTRIES)
    ConvertToBrush(#ICON, 1, 1, k)
    DisplayBrush(1, #CENTER, #CENTER)
    WaitLeftMouse
    Cls
Next
```

Furthermore, each icon entry can also have multiple frames. For example, Amiga icons typically contain two frames: The first frame is shown when the icon is not selected and the second frame is shown when it is selected. To get these different icon states, you have to query the `#ATTRNUMFRAMES` attribute. Here's how to extend the code from above to draw all icon entries and frames:

Example 2 – Get Icon States

```
LoadIcon(1, "Drawer.info")

For Local k = 1 To GetAttribute(#ICON, 1, #ATTRNUMENTRIES)
    For Local j = 1 To GetAttribute(#ICON, 1, #ATTRNUMFRAMES, k)
        ConvertToBrush(#ICON, 1, 1, k, j - 1)
        DisplayBrush(1, #CENTER, #CENTER)
        WaitLeftMouse
        Cls
    Next
Next
```

Amiga icons can also contain metadata stored as tooltypes. To get this data, you can use the `GetIconProperties()` function:

Example 3 – Display Icon ToolTypes

```
LoadIcon(1, "Hollywood.info")
t = GetIconProperties(1)

For Local k = 0 To ListItems(t.ToolTypes) - 1
  DebugPrint("Item:", k, "Key:", t.ToolTypes[k].key,
    "Value:", t.ToolTypes[k].value,
    "Enabled:", t.ToolTypes[k].enabled)
Next
```

`GetIconProperties()` also allows you to read other metadata that is stored in Amiga icons, like the icon type, its position on the Workbench, the size of the drawer window if the icon is a drawer icon etc.

Finally, if Iconic is activated, you can also use icons in formats supported by the plugin with the `@APPICON` preprocessor command. Thus, with Iconic installed you could also do something like this:

```
@APPICON "my_app.ico"
```

It's no longer necessary to use Hollywood's custom icon format or specify the individual icon images manually. With Iconic you can simply pass an icon in one of the common icon formats to `@APPICON`.

Saving icons

Icons can be saved using the `SaveIcon()` function from Hollywood's icon library. When Iconic is active, it will register the following new icon formats for use with `SaveIcon()`:

- `#ICNFMT_AMIGAOS`: classic AmigaOS icon
- `#ICNFMT_AMIGAOS35`: AmigaOS 3.5 icon (GlowIcon)
- `#ICNFMT_AMIGAOS40`: AmigaOS 4.0 icon
- `#ICNFMT_MACOS`: macOS icon (*.icns format)
- `#ICNFMT_NEWICON`: NewIcons icon
- `#ICNFMT_PNG`: MorphOS/PowerIcons icon in PNG format
- `#ICNFMT_WINDOWS`: Windows icon (*.ico format)

For example, to save icon 1 as a classic AmigaOS icon, you could use the following code:

```
SaveIcon(1, "test.info", #ICNFMT_AMIGAOS)
```

Note that all icon formats have their peculiarities. For example, classic AmigaOS icons must be

palette-based and the palette isn't stored in the icon file. So if you try to save an icon that only contains 32-bit true colour image data with alpha channel as a classic AmigaOS icon, Iconic will render the true colour pixel data down to 4 colours using the Workbench 1.x palette when saving the icon. This will probably lead to rather unsatisfying results, so for optimal results you should make sure that the icon you are about to save contains the pixel data exactly in the format as expected by the respective icon format. See the next chapter for an overview of pixel data formats supported by the individual icon formats supported by Iconic.

Some icon formats also support compression of the image and palette data. If an icon format supports compression, Iconic will compress the image and palette data. If you want to turn this off, set the table tag "*Compression*" to *FALSE* when you call `SaveIcon()`, e.g.

```
SaveIcon(1, "test.info", #ICNFMT_AMIGAOS35, {Compression = False})
```

To modify the metadata that is supported by AmigaOS icons, use the function `SetIconProperties()` from Hollywood's icon library. See the [Hollywood manual](#) for details.

Icon format information

For reference, here is some information about the icon formats supported by Iconic:

- ◆ **AmigaOS 1.x style icons:** This is a palette-based icon format which uses just a single image that should be in 4 colors. Note that the palette isn't stored in the icon itself but instead it is expected that the icon uses the standard Workbench 1.x colors.
- ◆ **AmigaOS 2.x/3.x style icons:** The same as AmigaOS 1.x style icons except that icons have two states now (normal and selected). Also, they support some additional metadata like tooltypes. Image data should be 4 colors and use the standard Workbench 2.x colors.
- ◆ **AmigaOS 3.5 icons (aka GlowIcons):** This is a palette-based icon format and in contrast to classic AmigaOS icons, the palette is stored inside the icon so that icons appear in the correct colors independent of the Workbench's palette. GlowIcons support a transparent pen, the maximum icon size is 256x256 pixels and the maximum color depth is 8-bit, i.e. 256 colors. GlowIcons also contain a 4-color fallback icon that will be shown on Amiga systems that don't support GlowIcons. Icons have two different states (normal and selected). Image and palette data can be stored with or without compression.
- ◆ **AmigaOS 4.0 icons:** This icon format is used on AmigaOS 4.x and supports true colour icons with alpha channel (32-bit). The maximum size is 256x256 pixels. AmigaOS 4.0 icons also contain a 4-color fallback icon that will be shown on classic Amiga systems that don't support AmigaOS 4.0 icons. Icons have two different states (normal and selected).
- ◆ **macOS icons (*.icns format):** Images inside the icon can't use arbitrary sizes like in the Amiga icon formats but can only use the following fixed pixel sizes: 16x16, 32x32, 48x48, 64x64, 128x128, 256x256, 512x512, 1024x1024. Sizes bigger than 128x128 pixels must only contain 32-bit alpha channel true colour images. Smaller sizes also support palette images but typically can contain 32-bit true colour data as well. Image data can be stored with or without compression.

- ◆ **MagicWB icons:** These are like AmigaOS 2.x/3.x icons but use 8 colors and a special palette known as the MagicWB palette. Like with classic AmigaOS icons the palette isn't stored inside the icon file so Iconic will treat all icons that use 8 colors as MagicWB icons.
- ◆ **MorphOS/PowerIcons icons (PNG):** This is an Amiga icon format based on the PNG format. It is used on MorphOS and on classic systems using the PowerIcons patch. Image data can be of an arbitrary size and must use 32-bit true colour pixel data with alpha channel. Icons only contain one image and one state.
- ◆ **NewIcons:** This is a palette-based icon format and in contrast to classic AmigaOS icons, the palette is stored inside the icon so that icons appear in the correct colors independent of the Workbench's palette. NewIcons support a transparent pen, the maximum icon size is 93x93 pixels and the maximum color depth is 8-bit, i.e. 256 colors. NewIcons also contain a 4-color fallback icon that will be shown on Amiga systems that don't support NewIcons. Icons have two different states (normal and selected).
- ◆ **Windows icons (*.ico format):** Images inside the icon can be up to 256x256 pixels. They can use image data that is palette-based or 32-bit true colour image data that contains an alpha channel. Image data can be stored with or without compression.

History

Version 1.0: (22-Jan-22)
- First release

Copyright

This plugin was written by Andreas Falkenhahn <andreas@airsoftsoftwair.de>
See the COPYING file in this package for conditions concerning distribution of this plugin. Visit <http://www.hollywood-mal.com> for more information on Hollywood and more plugins.