changeicon

COLLABORATORS

	TITLE : changeicon	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 27, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

changeicon

1.1 Inhaltsverzeichnis

```
     ### #                               ###
    #    #                                #
   #     ####    #### # ###   ### #  ###   #   ###   ###  # ###
   #     #   # #   #  #   # #   #  #   #  #  #   # #   #  #   #
   #     #   # #   #  #   # #   #  #####  #  #     #   #  #   #
    #    #   # #   #  #   # #   #  #      #  #   # #   #  #   #
     ### #   #  ### # #   #  ###    ###  ###  ###   ###   #   #  v1.0
                                #
                             ###
```

Contents:

1.2 Introduction

ChangeIcon is a program which installs itself as a » Workbench-application. The user can drop Icons onto ChangeIcon which will be replaced by default icons (depending on the type of the files). This makes more sense if files do not have any Icons yet. These files can also be dropped onto ChangeIcon if you select "Show -> All Files" from the Workbench's "Window"-menu.

Surely, there are already some programs of that sort, but non of them fitted my needs. On the other hand, I wanted to learn something about the locale.library and the whatis.library. Latter is able to detect the type of files and let's the user easily add own types to its internal list.

You soon won't miss ChangeIcon when creating diskettes that shall be sent to friends as you save a lot of time. Just omit the Icons at first and then let ChangeIcon do the work!

1.3 Copyright

Commodore is a registered trademark of Commodore Electronics Ltd. Amiga and Amiga Workbench are registered trademarks of Commodore-Amiga, Inc. AmigaGuide and Installer appear under the (C)opyright of Commodore-Amiga, Inc.

ChangeIcon is Freeware. I, Hanns Holger Rutz, have the Copyright for it. The following conditions must be followed if you want to (re)distribute the program (otherwise you're breaking the law, breaking the law):

- ChangeIcon may only be distributed in whole. The following files have to be kept together unaltered in the shown directory-structure:

```
ChangeIcon (dir)
     Catalogs (dir)
          deutsch (dir)
            changeicon.catalog          ; German language-file
          français (dir)
            changeicon.catalog          ; French language-file
       ChangeIcon.cd                    ; CatalogDescription-file
       ChangeIcon.ct                    ; CatalogTranslation-file
     Docs (dir)
       Deutsch                          ; German documentation
       Deutsch.info                     ; related icon
       English                          ; English documentation
       English.info                     ; related icon
       Français                         ; French documentation
       Français.info                    ; related icon
    *Icons (dir)                        ; contains sample-icons
     Install (dir)
       Deutsch.info                     ; German Installer-icon
       English.info                     ; English Installer-icon
       Français.info                    ; French Installer-icon
       Install.script                   ; Installer-script
     S (dir)
       FileTypes                        ; filetypes example-file
  ChangeIcon                            ; the program itself
  ChangeIcon.info                       ; ChangeIcon's icon
 *ChangeIcon_Source.lha                 ; the sourcecode, packed
  Docs.info                             ; (any) drawer-icon
  Install.info                          ; (any) drawer-icon
  ReadMe                                ; contents depend on FD-series
  ReadMe.info                           ; related icon
```

  Files and directories preceded by a asterisk can be copied along optionally. Packing of the whole directory (e. g. in order to put it in a BBS) is permitted, of course.

- the ChangeIcon-package may only be redistributed on FD-series if the maker of the particular series has got a signed license from me. Commerical sale

of the package is prohibited. That’s because there are still enough companies that sale FD-programs for too much money - the author gets no penny, of course.
Fred Fish is allowed to distribute the package on his AmigaLibDisks and his CD-ROMs.

- the ChangeIcon-package may be transferred via electronic networks. In this case, no license is neccessary.

- the sourcecode (if distributed) may only be used for learning purposes. Reassembled programs or modified sourcecode-files may not be spread.

IMPORTANT: THE AUTHOR, HANNS HOLGER RUTZ, IS NOT RESPONSIBLE FOR ANY DAMAGE CAUSED BY THE USE OF CHANGEICON. THE USER TAKES THE RISK HIM- OR HERSELF!

Of course, I’m happy to hear bugreports or suggestions (» Author’s address)!

1.4 Requirements

ChangeIcon requires at least OS2.04. In order to use ChangeIcon in an other language than English, you need the locale.library v38 which can be found on the Workbench v2.1 and higher.
At this point, I want to mention that it would be very kind of you if you translated the CatalogTranslation-file ("ChangeIcon.ct") into a new language that hasn’t got a language-file yet. Please read » New translation for more information on that subject.

ChangeIcon also needs the whatis.library (version 2 and higher). The Copyright for the WhatIs-package is owned by Sylvain Rougier and Pierre Carette. Note that, if you have received WhatIs along with ChangeIcon, WhatIs is distributed under its own conditions!

For speed reasons, a harddisk is recommended.

1.5 Installation

Normally, you’ll want to use the Installer-script in the "Install" directory. Then you need to have the Commodore-Installer in a directory accessable by your Workbench (e. g. "C:" or "SYS:Utilities/").

Keep these things in mind when installing ChangeIcon yourself:

- the whatis.library (if distributed) has to be put in the "LIBS:" directory.

- the filetypes example, "S/FileTypes", has to be copied to "S:" if no file of that name exists yet.

- if you want to run ChangeIcon in a particular language other than English, use the Catalog-files placed in "Catalogs/". Put the desired file in "LOCALE:Catalogs/<Language>/" and choose your preferred language with the

"Locale" preferences program.

1.6 Programstart

ChangeIcon has to be started from the Workbench since I was too lazy to code the shell-startup :-). These » ToolTypes are supported:

MODE=ICON|MENU
decides how ChangeIcon reacts when its window is closed. In ICON-mode an AppIcon appears, in MENU-mode an AppMenuItem is attached to the Workbench's "Tools"-menu. Default: ICON. (» Workbench applications)

POPUP=YES|NO
decides if ChangeIcon opens its window upon startup or appears as AppIcon/AppMenuItem. YES means that the window shall be opened. Defaults to NO.

EDITOR=<Path>
ChangeIcon offers the feature to edit the file that describes the filetypes. <Path> is the complete path of your favourite editor ChangeIcon shall call. If this ToolType is omitted ChangeIcon will try to read the environment variable "EDITOR". If latter is absent either, "C:Ed" will be used.

ICONDIR=<Directory>
this ToolType is very important. With <Directory> you specify the place where all the icons are stored which represent the different filetypes. Example: "Work:Icons/" More on that subject in » Operation.

ICONNAME=<Path>|<Name>
<Name> is the text that is placed beneath the AppIcon. If you want to replace the icon imagery itself either, you can give a complete path to an iconfile, e. g. "Work:Icons/ChangeIcon_AppIcon". Note that the text below the AppIcon will be "ChangeIcon_AppIcon" in that case.
Default: "ChangeIcon".
This ToolType is ignored in MENU-mode.

ICON_X=<Value>
ICON_Y=<Value>
ICON_X and ICON_Y specify the position in the Workbench window where the AppIcon should appear. The values are given in pixels (ICON_X) and rasterrows (ICON_Y). If these ToolTypes are not given, a free place in the Workbench window is chosen.
These ToolTypes are ignored in MENU-mode.

MENUNAME=<Name>
<Name> is the text that will appear in the "Tools"-menu of the Workbench (only used in MENU-mode). Defaults to "ChangeIcon".

DONOTWAIT

When you drop ChangeIcon in your "WBStartup" Drawer you usually give this ToolType in order to run ChangeIcon asynchronously.

When you are done with setting your personal ToolTypes, you can start ChangeIcon by simply doubleclicking its icon. Depending on the MODE- and POPUP-ToolTypes an (App)window, an AppIcon or an AppMenuItem will appear. If an error occurs ChangeIcon will let you know. In certain cases you are able to retry the startup.
(» Operation)

1.7 Operation

To make the explantion easier, I assume that you have set the » POPUP ToolType to "YES". In case of successful startup you can see ChangeIcon's window under your mousepointer. The interface offers the following gadgets (keyboard-equivalents are shown as underlined chars):

Auto type

When this gadget is checkmarked, ChangeIcon (i. e. the whatis.library) decides itself what the filetypes of the icons that the user drops into the window are. The possible types are listed in the listview at the left.

Types

this list contains all currently available filetypes. If you select one by clicking it, the "Auto type" gadgets gets switched off. Any icons dropped into the window will now be replaced by the icon that represents the selected filetype.
The types can also be selected with the Cursor-keys, Shift+Cursor skips on page, Alt+Cursor jumps to the top/bottom of the list.

DefaultTool

In "Auto type" mode this gadget is disabled. It becomes enabled when you select a filetype from the list. If you checkmark the DefaultTool-gadget, the String-gadget to the right is activated. Any text typed in will replace the DefaultTool of the default icon associated with the select filetype. E. g., if your default Text-icon contains the DefaultTool ":C/Look" but you want to create texticons with let's say ":C/MuchMore" as DefaultTool then you would select the text-filetype, checkmark the DefaultTool-gadget and type ":C/MuchMore" in the String-gadget.
The String-gadget can also be activated with the spacebar.

AppIcon /
AppMenuItem

Selecting this gadget will cause ChangeIcon's window to disappear. Depending on the chosen » MODE ToolType an AppIcon appears or an AppMenuItem is attached to the "Tools"-menu of the Workbench. ChangeIcon still reacts as in the Window-mode, except that you can't change any options, of course. You can change icons by dropping them onto the AppIcon or selecting them and choosing "ChangeIcon" from the "Tools"-menu. Later, I'll explain where the default icons are

searched.
ChangeIcon’s window can be popped up again by double-clicking the AppIcon or activating the AppMenuItem without having chosen any icons on the Workbench.

Quit

Selfexplanatory. (no security-requester)

Ok. But what exactly happens when you drop icons into the AppWindow or onto the AppIcon? ChangeIcon uses the whatis.library. The whatis.library uses a file called "S:FileTypes" which defines all known filetypes (the most important ones are implemented in the library). To edit the types or create new ones simply start a texteditor and edit the file "S:FileTypes". Starting the editor can be done from ChangeIcon’s menustrip (see later).
I’ll explain the structure of filetypes by looking at the AmigaGuide-type. Please refer to the WhatIsLibrary.doc for more information.

```
   # AmigaGuide-files
   TYPE "Guide"
     SUBTYPE        Text
     ICONNAME       "def_Guide"
     OPTNAMEPATTERN "#?.guide"
     SEARCHPATTERN  "@DATABASE"
   ENDTYPE
```

The first line starts with a comment. These are introduced with the number-char ’#’. "TYPE" introduces a new filetype, in this case named "Guide". "SUBTYPE" is optional. It says that the new type is based on an old one (AmigaGuide-files are Text-files). "ICONNAME" is essential for ChangeIcon. It tells the program what the name of the default icon for that type is. The directory where all the icons are stored may be specified with the » ICONDIR ToolType. "def_Guide" may, for example, be created with the IconEdit program. See » Integrated types for the internally implemented filetypes and their iconnames. "OPTNAMEPATTERN" is unimportant to ChangeIcon but may be used by other programs that use the whatis.library. The last keyword "SEARCHPATTERN" is important when using "Auto type". In this case ChangeIcon has to ask the whatis.library what type the files are you pass to ChangeIcon. To determine if it’s a Guide-type, the whatis.library searches the first bytes of the file for the text "@DATABASE" (case is not significant). There are more keywords to give type-specific information, these are described in the WhatIs documentation.

Back to ChangeIcon. Now, when you drop one or several icons onto the AppIcon or into the AppWindow, ChangeIcon identifies the filetype or takes the one you selected in the listview. In the next step ChangeIcon tries to load the icon associated with the filetype (given with the "ICONNAME" keyword) and replaces the icons dropped onto ChangeIcon. Default icons can be stored in:

- the directory specified with the » ICONDIR ToolType
- the current directory
- "ENV:Sys/"
- "ENVARC:Sys/"

If the filetype is unknown or the default icon could not be loaded, the screen flashes. Note that you can also create icons for files that do not have icons yet. Just choose "Show -> All Files" from the "Icons"-menu in the Workbench window, select the files you want ChangeIcon to create icons for and drop them as usually onto ChangeIcon. Another trick is to create filetypes that can never be used in "Auto type" mode, e. g. when you want to have an alternative docfile icon. Here is an example for such a type definition in "S:FileTypes":

```
   TYPE "Text2"
     ICONNAME    "def_Text2"
     NAMEPATTERN "~(#?)"         # no file can ever have such a name :-)
   ENDTYPE
```

Due to a bug in the whatis.library you always have to put such a "Fill"-type at the end of the typefile - otherwise the last "real" type doesn't get displayed in the listview. Thus, one should add these lines to "S:FileTypes":

```
   TYPE "Foo"
     NAMEPATTERN "~(#?)"
   ENDTYPE
```

And finally the description of ChangeIcon's menuitems:

Reparse types — If you've changed the "S:FileTypes" file, changes in the typelist become visible after choosing this item. You'll get a screenflash if the reparsing could not be done (usually because another program using the whatis.library is still running).

Edit types — Calls a texteditor in order to let you change the file "S:FileTypes". The editor's name is taken from the » EDITOR ToolType if given. Otherwise the environment variable "EDITOR" or - if it is absent, too - "C:Ed" is taken. Wenn you quit the editor, ChangeIcon automatically tries to reparse the filetypes.

AppIcon / AppMenuItem — See gadget description.

About... — Shows information on the program.

Quit — Guess!

1.8 Der Rest

Some last words. I'd like to thank...

· Gérard Cornu for all the French translations
· André Voget and Tim Jöllenbeck for Betatesting
· John Harper for the ingenious JEd-editor
· Rune Gram-Madsen for ASM-One V1.02 which I used to write this program
·~Fred Fish for his fantastic FD-series
· Commodore for the Amiga, the Installer and AmigaGuide

And now the incredibly long, interesting history:

v1.0 (02.-27. of February 1994) - it worx

Well. I could add the facility to start ChangeIcon from the shell and to select files via ASL-requester (as in MegaView - really nice program)... Momentarely, I'm simply to lazy, but maybe you are able to convince me or you have other suggestions. Then read » Author's address.

So long, have fun with the program!!

1.9 Adresse des Autors

You have a bugreport, suggestions, a translated » CatalogTranslation-file or want something else? Then please write to:

Hanns Holger Rutz
Auf dem Godenstedter Berg 17
D-27404 Zeven
GERMANY

1.10 Workbench applications

As of OS2.0 there are three types of Workbench applications: AppIcons, AppMenuItems and AppWindows.

- AppIcons are icons temporarly installed by a program. One can doubleclick them or drop other icons on it. In either case, resulting actions depend on the program that installed the AppIcon.

- AppMenuItems are menuitems which appear in the "Tools"-menu of the Workbench. One can select them with or without having selected icons on the Workbench.

- AppWindows look just like normal windows. But you can drop icons into them and special operations on these icons are performed. One example for this is the IconEdit program which comes along with your Workbench-set.

1.11 Translating ChangeIcon

You can do me a great favour and translate ChangeIcon into another language. To do so, you need an ASCII-editor and the file "ChangeIcon.ct" provided in this package. "ChangeIcon.ct" contains the English texts. Example:

```
; ** Req-text, LineFeeds allowed
MSG_NO_MEM
?
; Not enough free memory!
```

The first line is a comment containing additional information on the particular text. In this case we get to know that the text appears in a requester and may contain LineFeeds ("\n"). Don't care about the second line - it contains information only important to me. The questionmark in the following line has to be replaced by the translated string which is placed in the last line (defined as a comment).

When you have done a translation, just send me the new .ct-file on diskette and you’ll receive a Catalog-file for that language from me.
(» Author’s address).

1.12 ToolTypes

ToolTypes are parameters that can be edited from the Workbench: select an icon and choose "Information..." from the "Icon"-menu of your Workbench. All ToolTypes appear in the list shown in the new window that is popped up. One can edit them now by selecting them or clicking the "New" and "Del" Gadgets.

A ToolType begins with a keyword optionally succeeded by an equal sign and an option or text. Example:

```
   EDITOR=SYS:Tools/MEmacs
```

EDITOR is the keyword (case insensitive). In this example, the equal sign is followed by the path to a text-editor. If there are several predefined options they are separated by a pipe-symbol ("|") in this documentation:

```
   POPUP=YES|NO
```

I. e. you can either type "YES" or "NO" after the keyword.

Changes to the ToolTypes can be saved by seleting the Gadget with the same name.

1.13 Integrated types

The following types are integrated into the whatis.library. The iconnames are "def_<TypeName>", e. g. "def_Text" for the "TEXT"-type.

8SVX	IFF/8SVX sound
ANIM	IFF/ANIM animation
Assign	Assign
Dir	Directory / Drawer; Note that the iconname is "def_Drawer"!
Exe	executable program; Note that the iconname is "def_Tool"!
FTXT	IFF/FTXT textfile
IFF	generic IFF-file
ILBM	IFF/ILBM picture
ILBM24	IFF/ILBM picture with 24 bitplanes (16,8 million colours)
Imp Data	datafile packed with "Imploder"
LHArc	archive packed with "LHArc" or "LHa"
Lib	a library (as found in "SYS:Libs")
Med MOD	piece of music created with the Freeware program "MED"
Object	object-module (used by programmers)
PP Data	datafile packed with PowerPacker
PP Exe	executable program packed with PowerPacker
PP30 Exe	executable program packed with PowerPacker v3.0
PP40 Exe	executable program packed with PowerPacker v4.0

PREFS	IFF/PREFS file as used by the programs found in "SYS:Prefs"
Pure Exe	executalbe program which may be made resident (PURE-bit set)
Script	batchfile for the shell (has the "S"-bit set)
SMUS	IFF/SMUS song
TERM	file created by "Term"
Text	bare textfile
Volume	Volume; Note that the iconname is "def_Disk"!
Zoo	archive packed with "Zoo"

1.14 Index

Too lazy... :-(