EasyWB

COLLABORATORS			

	TITLE : EasyWB		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 18, 2023	

REVISION HISTORY			

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

EasyWB

1.1 EasyWB guide

```
                          EasyWB – Version 1.4
             Copyright © 1996 Arian T. Kulp
             All Rights Reserved


        What is EasyWB

      How does it work?

    System Requirements

       Installation

          Usage

       To Do/Bugs

         Thanks to...

      About the author

          History

         Copyright
         EasyWB is EMailware.  Please Email
   me
        if you get this.  I
 don't care if you hate it, or love it just let me know what you think.
```

1.2 What is EasyWB?

```
          What is EasyWB?
```

EasyWB is the result of many hours of my toiling at the computer, trying to
make a simple way to handle an arbitrary file. It finally works!

With EasyWB, just define a few things, then you can drop any file (even if the
file has no icon of its own) onto the
                  AppIcon
                   to execute the
appropriate program.

For example: just drop a picture on the icon and EasyWB will tell ViewTek to
view it. Drop a text file on it and your favorite text editor or viewer will
handle the rest. File recognition is based on either standard AmigaDOS file

                  pattern-matching
                  , file size, or up to the first 12 bytes of the file.

Note: EasyWB is now a commodity. This means that when it is started, it's
name will appear in Exchange and it's GUI can appear or disappear by clicking
the appropriate button in Exchange. By default, it's hotkey for popping up is
alt shift 8 (alt-*).

1.3 How does it work?

How does it work?

In my mind, what makes EasyWB easy to use, is its object-oriented approach
to handling files. When you drop a file on its AppIcon, EasyWB scans
through a list of filetypes that you define. When it matches the file up
with one of the filetypes you created, it uses the program object you
assigned to that filetype.

So you first make a list of program objects i.e. ViewTek, AmigaGuide, more,
then you make a list of filetypes i.e. IFF picture, AmigaGuide document,
etc., then link them together.

This may not sound easy to use at this point, but once you try it out, it
won't seem so bad.

1.4 System requirements

System Requirements:

Any Amiga with OS2.0 or greater.

EasyWB will use ReqTools (reqtools.library), if present, for the optional
string requestor used in arguments. This library is not required, but without
it you cannot use the "String Requestor" argument in your program objects.
As most people have ReqTools, this should not be a problem.

Also, EasyWB consumes about 29k plus memory allocated for your preferences.
The more objects defined, the more memory used (though it uses very little).

No other files or libraries are needed.

1.5   Installation

```
Installation:

Installation is simple.  Double-click on EZInstall, and a script will ask
if you would like EasyWB copied to your WBStartup drawer.  This is the best
place for it, as it makes it available to you all the time.  It also copies
a sample preferences file which defines basic filetypes and program types.

To use the supplied AmigaGuide format documentation, simply copy it wherever
you'd like.

Note: Once installed, dropping files onto the AppIcon will appear to have
no effect.  This is because filenames need to be given for the applications
so EasyWB knows what to use (i.e. c:more, sys:utilities/wdisplay. etc.).
```

1.6   Usage

```
                Usage:

For the most part, EasyWB is intuitive.  But until you understand it's basic
principles, you may need a little help.

First of all, EasyWB has two different methods of accesss.  The first is its

                GUI
                  (Graphic User Interface).  This is the only way to configure  ←
                        EasyWB.
The second is its
                CLI
                  interface.

Please note: Only one copy of EasyWB can be run at one time.  This made sense
to me, as why would you need two AppIcons which do the exact same thing?
Trying to start a new EasyWB will simply bring up the requestor from the pre-
existing EasyWB.  This is transparent and will not look any different than if
you had simply double-clicked on the EasyWB AppIcon to begin with.

Also: I have done a few things to make EasyWB a good
                MultiView
                 replacement
for pre OS3.0 users.
```

1.7   Things to do/Bugs in EasyWB

```
                Bugs:
    One thing I can't seem to track down is a small memory loss.  If you open
    any windows, it is about 208 bytes, without using windows, it's about
    112 bytes.

    Also, currently when you drop an unrecognized file onto the AppIcon (only
    if DEFAULT is not defined), a window comes up saying so.  Previously,
```

```
    the file add window popped up, autofilled to add a new file type.  For
    some reason, this now causes a crash every time the window is open,
    and I can't figure it out.  This is getting frustrating!
```

To do:
```
    I am considering making it possible to define the same filetype multiple
    times pointing to different program objects.  This would be done so you
    could, for example, define a picture twice, then have it point to a
    viewer, and an editor (maybe DPaint).  Then when you drop a picture on
    the AppIcon, a requestor would come up like this:
```

```
                    --------------------------------
                    |@|EasyWB                       |
                    --------------------------------
                    |   The file you have dropped:  |
                    |          Pic.IFF              |
                    |   is defined more than once.  |
                    |                               |
                    | Please choose program to use: |
                    |                               |
                    |     --------- ----------      |
                    |     |DPaint | |ViewTek |      |
                    |     --------- ----------      |
                    --------------------------------
```

```
    Please tell me if this is something I should spend my time on.

    In addition, I have no support for drawers being dropped on the AppIcon at
    this time.  I finally added batch processing, but you may only drop files
    for now.

    Finally, though all gadgets seem to have keyboard equivalents, they
    actually don't.  I can't seem to figure this one out, but I will try to get
    it working.  I hope this doesn't cause too many problems for anyone.
    If anyone knows why I can't seem to receive VanillaKey messages from these
    windows (yes, I set the right IDCMP flags!), let me know.
```

```
If you find any bugs, I won't be surprised ;) and I want to know about them.
Get in touch with
                me
                 and I'll do my best.
```

```
Also, I am open to any suggestions.  Please let me know if you have any.
```

1.8 Thanks to

```
Thanks go to:

Anthony Moringello.  For his assistance with many C questions, and example
                     source from an AppIcon program he wrote (not very
                     recognizable anymore!).

Jess Sosnoski.       For acting as a guinea pig, and offering much
                     constructive criticism during the development of EasyWB.
```

Jan van den Baard.   For the wonderful program GadToolsBox.

Nico Francois.   For the easy-to-use but so powerful ReqTools.

Commodore-Amiga.   For writing an operating system which is just so fun to
                     program!

1.9 About the author

                    About the author.

Hmmm, not much to say. I've been writing C programs for my Amiga for several
years now. This is the only one which I've actually felt comfortable enough
with the idea of distributing. It works! Consistently!!

I also enjoy music, reading, and writing. I love my Amiga, but not to the
point of fanatacism, and I love programming on it. I enjoy talking to others
that are the same way.

As for other work I've done, if anyone has the Xetec American Heritage
Illustrated Encyclopedic Dictionary CDROM and cannot run it (I can't on
my
                system
                ), I wrote a pretty good program for viewing words and definitions ↩
                    .
It's WorkBench-friendly, saves to the clipboard or another file, prints,
and has a GUI, or shell based interface in the same program for about 20k!

To get in touch,

My email address is: akulp@blue.weeg.uiowa.edu

or USPS address: Arian T. Kulp
                 1019 N. Dodge St.
                 Iowa City, IA 52245

and my phone number is: 319/351-0622

I love to hear from other Amiga owners, especially programmers.

1.10 EasyWB History

Not much here yet.

v1.1   Beta release.
        Fixed problem of crashing if a requestor was brought up without an
        open window.

v1.2   File requestors added.
        Internal hex recognition added.
        File size recognition added.
        Improved text entry gadget handling by not requiring the user to

```
          hit enter or return to make EasyWB notice changes.
          Fixed file pattern matching to be case-insensitive.


v1.3      Allowed batch processing of files.
          Added Use/Save/Cancel buttons on windows, instead of ambiguous
            close gadget handling.  Should be more intuitive now.
          I actually lock onto the program to call now, so if nothing is
            there, you won't experience a lockup.
          I also do a test now to see if the file dropped on the AppIcon
            actually exists.  This could be important!
          Fixed a bug so you can now specify EasyWB as default tool in
            project icons so you can just double-click on a project.
            This works great if you create a link to EasyWB as Multiview
            so files from archives won't need tooltypes editted.


v1.4      Handles extend-select from Workbench.  Now you can select EasyWB
            and another file(s) and double-click to act upon them.
          Now it's a commodity.
          Rewrote preferences to be IFF standard now.  No odd size chunks!
          Added configurable console window which can be used for program
            objects.  Only opens window if toggled in object and if program
            actually needs it.
          Created a nicer, more style-compliant icon.
          General speed-ups/optimizations.
```

1.11 Copyright

```
   The package "EasyWB - Version 1.4" is Copyright © 1996 by
   Arian T. Kulp. All Rights Reserved.

   This package can be freely distributed as long as:

   1. It is not sold; only a reasonable charge for copying and storage
      medium is allowed.

   2. All of the following files are included in their original form
      without modification of any kind.

      EasyWB
      EasyWB.info
      EasyWB.guide
      EasyWB.guide.info
      EasyWB.prefs
      EZInstall
      EZInstall.info

   3. No crunching of executable allowed.

   Permission is hereby granted to include EasyWB in PD compilations such
   as Fred Fish or Aminet CD.

   This software is provided as-is, without warranty either expressed
   or implied.  In no event will the author be liable for direct,
   indirect, incidental or consequential damages or loss of data
   resulting from the use of this software. The risk as to results and
```

performance of this software is assumed entirely by the user.

1.12   Usage from GUI

From the GUI:

First of all, when you first start EasyWB, you won't see much. After a few
seconds, an AppIcon with a gray background and an asterisk (yes, that's
what it's supposed to be -- because you can drop any file on it) appears on
your WorkBench. At this point (the first time anyway), there isn't anything
you can do with it except double-click on it.

Upon double-clicking, a window should appear as follows:

```
          ---------------------------------
          | EasyWB Preferences          #|
          ---------------------------------
          |                               |
          |  Add  @| File Definitions  |  |
          |       ----------------------  |
          | Del  |       L              |  |
          |      |          i           |  |
          |      |          V  s        |  |
          | Edit |          i   t       |  |
          |      |           e          |  |
          |  Up  |              w       |  |
          |      |--------------------|  |
          | Down |_____|  |
          |                               |
          |                               |
          |    Use     Save    Cancel    |
          ---------------------------------
```

We'll address each gadget one at a time.

Add: The Add gadget brings up a new window allowing you to define a new
    object. Which type of object depends on your current list.

Del: The Del gadget deletes the current object. No effect if there is
    no current object.

Edit: The edit gadget brings up a window to edit the current object.

Up: Moves the current object up the list one position (unless it is at
    the top).

Down: Moves the current object down the list one position (unless it is
    at the bottom).

Use: Saves the current settings to env: and hides the window. These
    settings will stay in effect until the next reboot.

Save: Saves the current settings to envarc: and hides the window. These
    settings will survive a reboot.

Cancel: Hides the window. No changes that you made without saving will

remain.

The gadget showing File Definitions in the example above is a cycle gadget.
By pressing this gadget, you toggle between the two lists: "File Definitions,"
and "Applications." When you select this gadget, the listview below it
updates itself to reflect the items in the corresponding list.

The Listview gadget shows all objects in the current list (as defined by the
cycle gadget above it). The space underneath the listview is a string
gadget which you can use to edit the object name without double-clicking.

For info on adding or editing File Definitions click here:

            For info on adding or editing Applications click here:

            EasyWB also has a
            menu
             and the icon has
            tool types
            .

1.13 Filetypes

            Adding/editing filetypes:

Upon pressing "edit" on an existing object, or "add" while in the file
objects list, a new window will appear:

```
            ---------------------------------
            | File Definition Edit        #|
            ---------------------------------
            |   Object Name  _____  |
            |                                |
            |     Object ID  _____  |
            |                                |
            |   Recognition  @|Internal-ASC  |
            |                   _____  |
            |      Program  |_____|  |
            |                                |
            |                                |
            |     Use        Cancel        |
            ---------------------------------
```

Object Name :   This string gadget is used for you to give the current file
                type a name. This way you can define a Gif picture and name
                it GIFPIC or something similar. This just makes it easier
                for you when you look in the list to add, edit, or delete.
                This is useful only to you, as EasyWB disregards it.

Object ID   :   This string gadget allows you to enter up to 12 characters
                for EasyWB to use in identifying this filetype. This can be
                a tough one to figure out for some files, so I have provided
                a few
                examples

in the included preferences
file.

The use of this gadget changes based on the Recognition
gadget.

For internal strings, case is significant, and question marks
can be substituted for values that are not always the same in
the same types of values. Example: IFF files usually start
with FORM, but the next four bytes reflect the size of the
file. Since most files have different sizes, use four question
marks here.

For filename strings, case is insignificant, and standard
AmigaDOS
pattern
 matching is used.

NOTE: This string has no use for ASCII and Default types.

Recognition :   This cycle gadget toggles between six ways EasyWB
                 recognizes files. The six ways, and how they work follow:

                 Internal-ASC
                 EasyWB compares the File ID String against the first 12
                 characters of the file (character by character).

                 Filename
                 EasyWB compares the File ID String against the actual filename.

                 Filesize
                 EasyWB compares the File ID String against the actual size of
                 the file in bytes. Note: with this option, the string must be
                 a number. (duh!)

                 Internal-Hex
                 This works like Internal-ASC, except is compares up to the
                 first twelve bytes of the file against the literal hex values
                 in the File ID String as follows:
                 typing 00 00 03 f3
                 would find the values 000003f3, not 30 30 30 30 30 33...

                 ASCII
                 EasyWB uses this type if no other filetype matches the dropped
                 file and the file is plain ASCII text. This should probably
                 call a text viewer, hex viewer, or text editor.

                 Default
                 EasyWB only calls this object if nothing else matches up. If
                 you have an ASCII object, it will catch plain text files, but
                 if the dropped file filters past all your objects, and it is
                 not ASCII, this object will be used. NOTE: You may not want
                 to use this type until you have defined all the objects you
                 need. By default, EasyWB will present you with a requestor
                 telling you about unrecognized files when it first sees them.
                 Once you have a "default" type, every file will be recognized
                 one way or another. (Sorry, auto-add not currently functioning).

```
Program      :    This gadget is used to actually assign a program object to
                  the current filetype.  A window with only a listview containing
                  all program objects will appear.  Simply select the object you
                  want.  This is where it is a good idea to have your program
                  objects named appropriately.

Use          :    Closes the edit/add window while retaining changes you have
                  made.

Cancel       :    Closes the edit/add window losing any changes made.  Note: if
                  you were adding an object, Cancel loses the object altogether.

Play around with these and experiment.  They only make sense as you use them!
```

1.14 Program objects

```
Adding/editing program objects:

Upon pressing "edit" on an existing object, or "add" while in the program
objects list, a new window will appear:


          --------------------------------
          | Application Editor        #|
          --------------------------------
          | Object Name _____ |
          |                              |
          | Output Win  [ ]              |
          |                              |
          |    Filename _____GET |
          |                              |
          |   Arguments _____ |
          |                              |
          | Argument 1  @|Filename       |
          | Argument 2  @|Filename       |
          | Argument 3  @|Filename       |
          |                              |
          |      Use      Cancel         |
          --------------------------------


Taking the gadgets in order:

Object Name  : This string gadget is used for you to give the current program
               a name.  This is useful only to you, and EasyWB never uses it.
               For my own config, I use Viewtek to view pictures and animations,
               and I name that object "PicView."  I use muchmore to display text
               files, and name that object "TextView."

Output Win   : This is a checkbox which tells EasyWB to open a console window
               if the program has any text output.  This window will only be
               opened if needed.

Filename     : This is where you tell EasyWB which program to use.  This must be
               a fully-qualified path name, or something in the system path list
               which identifies the program.  Examples would be:
```

```
                         SYS:utilities/more
                         WORK:viewers/vt
```

Get   : Calls an ASL file requestor to select a program filename.

Arguments   : If the program requires any arguments (most do), specify them
                  here. For example, the sample player oplay will play a
                  digitized sample, if invoked from a shell or CLI as follows:
                     oplay raiders.snd

                  In order to define this, you would enter oplay as Filename
                  (with a path if needed), then for File Arguments, enter %s.
                  %s is how you tell EasyWB to insert a word. EasyWB will
                  substitute, based on what is selected in Argument 1, a value
                  for %s. For many programs, it is a good idea to enclose %s
                  in quotes. This way, if the filename dropped on the AppIcon
                  has spaces, they will be handled properly.

Argument 1-3 : These cycle gadgets tell EasyWB how to replace the %s's in the
                  File Arguments string gadget. It defaults to Filename, which
                  is how most programs want it. Clicking on it once changes it
                  to Directory (replaces the %s with just the directory of the
                  file). Clicking on it once again, changes it to String
                  Requestor. The option requires reqtools.library to be
                  installed in your libs: directory. This option makes EasyWB
                  bring up a string requestor for you to manually enter that
                  argument each time a file is dropped on it. This could be
                  useful for a program like LhA. This way you could enter "x"
                  to uncompress the archive, or "v" to view the archive each
                  time a different archive was dropped onto the AppIcon.

Use   : Closes the edit/add window while retaining changes you have
                  made.

Cancel   : Closes the edit/add window losing any changes made. Note: if
                  you were adding an object, Cancel loses the object altogether.

Play around with these and experiment. They only make sense as you use them!

1.15 EasyWB menu

With the EasyWB prefs window open, an Intuition menu is available.
Two submenus are available: Project and Options:

```
    Project            Options
    -------            -------
    Open Prefs O       Output Win...      W
    Save As... A       ~~~~~~~~~~~~~~~~~~~~
    ~~~~~~~~~~~~       Reset to Defaults  D
    About      ?       Last Saved         L
    ~~~~~~~~~~~~
    Hide       H
    ~~~~~~~~~~~~
    Quit       Q
```

```
Open Prefs         : Opens a file requestor to select a prefs file to open.

Save As...         : Opens a file requestor to select a prefs file to which to
                     save the current settings.

About              : Brings up an informative requestor with a few facts about
EasyWB.

Hide               : Closes the prefs window while leaving the program running.
                     Does not save prefs anywhere.

Quit               : Hmmmmm.... (does not save anything, or ask if you're sure)

Output Win         : Opens a window you can move around and resize.  If you
                     specify an output window in a program object, it will use
                     this size and position.
                     These settings will be saved with the preferences file.

Reset to Defaults : Clears all current prefs.  This is internal only, and
                     will not affect any prefs files you have selected or
                     previously utilized.

Last Saved         : Loads the preferences previously saved using the Save icon
                     in the prefs window itself (not from Save As... in the menu).
```

1.16   Icon tool types

```
Tool types are accessed from the Workbench by selecting the icon (not
AppIcon), then going to the Workbench Icon menu and selecting
Information.

The following tool types are supported by EasyWB:

ICONX Use this to specify the X position of the AppIcon on WorkBench.

ICONY Ditto for Y.

OutputWinTop     \
OutputWinLeft     \  Specifies default settings for the output window.
OutputWinWidth   /   These can be changed through the Options menu.
OutputWinHeight /

QUIET   Normally, EasyWB will bring up a simple requestor when it is started.
        Use QUIET to suppress it.  You want it suppressed if it is in your
        WBStartup drawer.
```

1.17   Usage from CLI

```
From the CLI:

Using EasyWB from the CLI (or WorkBench Execute Command... option) has a few
peculiarities to get used to.  As noted before, you can only have one EasyWB
```

running at one time. I decided that this was a good idea for the following
reason.

My biggest use for EasyWB is to drop files on, but I also use it from within
ToolManager, and from within the shell to handle files. Simply type:
  EasyWB <filename>
to let EasyWB figure out the file and act on it. If EasyWB is already
running, typing this will simply send a message to the already running
process (thereby avoiding loading in preferences a second time), then quit.
The pre-existing process handles the file.

If EasyWB is not already running, it will load in the preferences, act on the
file, then quit.

Please note: You can only invoke the EasyWB AppIcon/GUI from CLI if you
use the command:
  EasyWB
Notice no filename after it. This will start it up, but you will not get
your shell back unless you run it:
  run <>NIL: EasyWB

I feel the best way to use it is to keep EasyWB in WBStartup. Then you can
always drop files on it, or call it from a shell. It's pretty compact in
size, so you won't notice it much.

Options for running EasyWB from CLI are:
    FILENAME,CLIPBOARD/K/S,UNIT/S
What this means, is you can either use a filename:
    EasyWB sky.iff
or, view the clipboard:
    EasyWB CLIPBOARD
which would view clipboard unit 0.
To view other units, use:
    EasyWB CLIPBOARD UNIT [0..255]
Note: To use the UNIT keyword, you must have already used the CLIPBOARD
keyword.

1.18   Using EasyWB as MultiView

Under 3.0+ there is a program called MultiView which uses external libraries
to handle arbitrary files. You can view pictures, play sounds, read Amiga-
Guide documents, etc. all with this one program. Unfortunately, below 3.0
we have nothing like it.

For people like me, this creates a problem. I get a lot of new archives from
Aminet and BBS's, and many of them have MultiView set as the default tool.
Now instead of having to be forever editing the default tool string, just
rename EasyWB to MultiView, and as long as you have defined all of the file
types, it will work almost as well.

Even better, create a link from SYS:utilities/MultiView to EasyWB, and you
can keep track of EasyWB better.

I have also made the command line options look like MultiView by giving it

options to use the clipboard instead of a specific file.

1.19   What's an AppIcon?

```
AppIcon:
A gateway to a program, which utilizes an icon drawn directly onto the
WorkBench screen.  To access the program, just double-click on the AppIcon,
or drop a file onto it.
```


1.20   What's pattern-matching?

```
Pattern matching:
A method of specifying files from a list, by identifying similarities in the
desired files.


Example
Pattern : #?.txt
Result  : Any files with ".txt" as the last four characters of the filename.
          README.txt, BBSLIST.txt would show up, whereas README or txt.BBS
          would not.


Pattern : #?(~.info)
Result  : Every file in the directory, except icons (files with ".info" at the
          end.


Notice the tilde (~) character can be used to negate a pattern.  Also,
parentheses ()'s can be used to specify just a portion of the name.
```


1.21   My system

```
My system consists of an Amiga 2000 with A2620 card (screaming 14Mz 68020
card with a blazing 14Mz 68881 math co-processor), OS3.1, a whopping 2x speed
Reno Mediavision CDROM drive, Viva (sit down for this) 2400 modem
(with fax, of course), external Pyramid MIDI interface, 6 megs of Fast RAM,
1 complete meg of Chip RAM, and a 120M SCSI HD.  (Oh yeah, and 2 (two) low-
density floppy drives)
```


1.22   File ID String example values

```
File ID String example values:

Internal:
    FORM????ILBM     standard IFF picture
    FORM????ANIM     standard IFF animation
    FORM????8SVX     standard IFF sampled sound
    GIF87 or GIF89   GIF format picture
    ??????JFIF       JPEG picture
```

```
    ??-lh              LhArc, LZH archive
    @database          AmigaGuide database

Filename:
    #?.c               C source file
    #?.h               C header file
    #?.mod             many mod music files
    mod.#?             many more mod music files
    #?.FL[I|C]         IBM FLIck animations
```

Just a few examples. Most data files these days have some kind of ID string.
In most files, it's internal, but in a few (like most MOD's), you still must
use filename identifiers.