DzDir

COLLABORATORS			
	TITLE : DzDir		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 6, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

DzDir

1.1   DzDir.guide

                      The legalities: I accept no responsibility for anything this ←
                           program does! You may
copy it as long as you charge less than $5. Fred Fish may distribute it on his ←
   Fresh
Fish Roms. Everyone has a legal section, so I thought that I'd do it too!

  DzDesktop is a collection of utilities to enhance AmigaDos and speed its opera-
tion. It offers a collection of tools to help the user to manipulate their files, ←
   view and
edit them. DzDesktop's programs are object oriented programs and are all hopefully
intuitive enough for the novice to easily use. To gain help at any stage while ←
   using
DzDesktop place the mouse above an abject (eg: a list, button or gadget) and tap
the help key.

  At present the only part of DzDesktop that has been implemented is DzDir, which
is a directory utility. Other programs are in the pipeline, but I thought that ←
   after about
a year in development I should release it.

The documentation is designed to be used on-line, but here is a list of all
the nodes that you may view;

               Features

               Filelists

               Buttons

               Input methods

               Instant Filetype Recognition

               ASL

               Hotkeys \ AppIcon

               Maincfg

1.2 DzDir.guide/Filelist

  The filelists work with all input methods (eg. Appicons and buttons) on a last
active filelist basis. If you highlight the window then that window's filelist is ←
   the active
filelist. I have not used the DirWork left\right mouse button method, not because ←
   I
don't see it as a viable method, but so that my dock window may have a menu.
  You may change the way the filelists are sorted either by the lists menu or by
returning the internal function "sort by ...". This method then becomes the new ←
   pre-
ference. Which is permanent provided you save that configuration by quiting DzDir
before you shut down.
   Both filelists are susceptible to appevents and will load the contents of the di ←
   -
rectory or file that is dropped on it. (If a file is dropped on the filelist then ←
   the file's di-
rectory is loaded, eg. "work:finalwriter2/fw" will load "work:finalwriter")
  The list's attributes are mainly configurable by the MUI-Prefs program. The
Font, colour, scrolling style, etc are at your disposal provided you have ←
   registered the
MUI system.
  The mouse or keyboard can be used in conjunction with DzDesktop's instant
filetype recognition to run specified programs on types of files.
  These filelists are usually the source of filenames for executables. There are
two exceptions; If you have checked the "No Filenames", or if the directory ←
   windows
aren't open.
   Information about files and directories is displayed at the top of each ←
     directory
window. This is in the format "selected dirs/total dirs selected files/total ←
   files se-
lected amount/total amount". The scale for the amounts varies for ease of reading, ←
    it
can be in bytes(B), kiloBytes (kB), MegaBytes(MB) or GigaBytes(GB). This size in ←
   -
formation will not include directories, to find the size of directories simply ←
   highlight
the one(s) you want, and return the internal "File size" function, or select " ←
   Lists/File
size" from the dock window's menu.

1.3   DzDir.guide/Features

                        DzDir's current features include;

*Uses MUI for maximum future benefit
*Command line size dynamically allocated from 0-1024 bytes (DW 1.62 40, DirOpus
4.11 200)
*Unlimited AppIcon support (only limit is set by your RAM)
*Unlimited Hotkey support (only limit is set by your RAM)
*Up to 100 configurable buttons
*These buttons may be configured simultaneously if desired
*Multiformat Date/Time display
*Multiposition RAM gauges
*May open without the Directory windows (ie. Dock only)
*System friendly capture of output (no console windows)
*Ability to View a text file from within a Lha archive (provided you have Lha V1 ←
    .32+ in
your path). To do this just double click a filename in the output list, to return ←
    to
the archive just double click in the list. To get the output list to contain the
filenames of a lha, simply configure the LHArc datatype to "lha v", with async
and output to list checked, and double click a Lha.
*Ability to Extract\Delete file(s) from within a Lha archive (provided you have ←
    Lha
V1.32+ in your path). To do this hit "x" to extract and "d" to delete. This works
when viewing a lha, much the same as viewing a text file from within a lha,
except press "x" to extract.
*Many objects respond to appevents for ease of configuration (eg. if a button has ←
    no
Text/Icon you can drop a file on it and it will auto config the button)
*Instant file type recognition (via the WhatIs.Library)
*Ability to add new filetypes as they evolve
*Default icons may be set up so DzDir will add the correct icon for filetypes
*Hotkeys may be setup for buttons, appicons, or just as hotkeys
*Limited Arexx Support
*On-line Help (Via AmigaGuide)
*Information strings showing the size/amount selected for files
*Icon edit window like Workbench's "Icons/Information..." window
*Proper use of the ASL.library and MUI for all Graphic effects (eg. Buttons, ←
    windows)
*Ability to put either an icons image in a button, text, or an Iff image
*Left and Right mouse buttons may be configured to do different things to a ←
    filetype.
(eg. view/extract a Lha)
*Appevents on the filetype list will select the type of file dropped on it.
*A wide array of
                    Internal
                     functions.
*Ability to move buttons horizontally and vertically, Add a new button to the left ←
    /right,
copy the button to the left/right or delete any button.
*Auto-tileing of buttons.
*Internal Copy command, complete with internal progress gauge.
*You decide if the source(src) or destination(dst) filelists are to be updated ←
    after all
actions.

*DzDir can be in either the source or destination directory, great for extraction ←
    of lha
files into the destination directory.
*Cloning and swapping of directories.
*Very descriptive file information display so you too shall know what DzDir is ←
    doing.
*Quick Access Group (QAG). This keeps a list of previous directories for reloading
later.
*Ability to create softlinks.
*Ability to put an IFF image in a button, remaping is performed if you have V3+ of
AmigaDos.
*Ability to put an icons image in a button if so desired.
*Qualifier + key to find a file in the filelists.
*Close progress window after actions if so desired.
*Filtration based on filename, a range of size, a range of days old, exact size, ←
    or ex-
act age, or filetype (Via the WhatIs.Library)
*Infinate support for new filetypes.
*Uses WbInfo() to display an icon's information. Like Workbenches
"Icon/Information...".
*Uses IoErr() for a more descriptive error output
*Can search multiple directories recursively for any of the Filtration parameters ←
    and
display a list of the matching files.

  I also feel that I should point out a inferiority;
*has become "apparently" slow. For this you must also consider the MUIMas-
ter.library and all of the classes. The first time you run a MUI application then ←
    all of
these classes that are required must be loaded from disk as well as the MUIMas-
ter.library. On my system the startup only takes 2.4 seconds the second time. ←
    Having
images in your buttons slows DzDir down, but I think appicons are the worst for ←
    time
consumption.

1.4 DzDir.guide/Buttons

                    DzDir allows you to have up to one hundred configurable buttons. ←
                      These are
either at the base of the directory windows or in the dock window.
  A button may contain text, an icon, or an IFF depending on its configuration.
  All buttons, except those that return an internal function, are susceptible to
appevents (dropping files on the button). This will not work for internal ←
    functions, and
may produce an adverse result if attempted.
  A button can either execute an AmigaDos command, return an internal func-
tion, or load a desired path into the active directory window. This is dependant ←
    on the
way you have configured that button.
  To configure the buttons choose "Edit" from the dock window's menu, then
simply press the button(s) you wish to configure. When a button is pressed with ←
    edit

mode on, it will pop a configuration window up for itself. You can edit all of the ↩
    but-
tons simultaneously if you wish. For any buttons new attributes to take effect you
must close that button's edit window.
See
                    Input methods
                     For the specifics of the attributes.

1.5   DzDir.guide/Input Methods

   You will find that all of the input methods are configurable in mainly the same
fashion. You will simply find that attributes not applicable to that method are ↩
    missing.

The attributes are;

Name:
·You may give a alias to certain events. This is done so you can easily find the ↩
    event
you desire. The name field of an appicon is the text below the appicon.

Either button name/icon/IFF name rollo:
·This rollo controls the button type of text/icon/IFF, if text you receive a ↩
    string gadget,
if icon or IFF you also get a ASL-Pop object. Just put the text you want in here
or the name of the icon/IFF you want. In the case of the "Other" configuration
you may only choose an Icon, This will be the appicons image.

Desired action rollo:
·This tells DzDesktop which type of command you wish for this button to have. It ↩
    is in
the form Executable/Internal/Path. If you choose Executable then you will get
a string gadget and a ASL-Pop object. If you wish for a internal command
DzDir will have a Pop-Object with a list of all the internal commands you may
choose from. And if you want a drive button, simply choose path. If a com-
mand name is comprised of characters including a space you must put quotes
around your command (eg. "Work:FinalWriter/My stuff is in here/my file"). This
is done like this so you may put "lha -v" in the command string, and Lha will
take -v as a parameter. (given that the quotes aren't there). For AmigaDos
commands you can choose where to put the filenames by putting %s where
you want them, if there is no %s in your line the names will be put at the end.
Dont put more than one %s here. The one %s is equil to all the filenames if
stack is checked.

Hotkey:
·The keyboard short-cut for this button

Hotkey Sample:
·pressing this button will cause DzDir to sample the keyboard for input, and place
any acceptable hotkeys received into the Hotkey string. To abort this loop
simply press 'esc' or close the window. (The hotkey is in the form of "lalt
lamiga lshift d")

Run Async:

·If this box is checked then DzDir will run the command asynchronously (not wait ←
    for
the command to return) otherwise DzDesktop will wait for the command to re-
turn before DzDir can do anything else. (syncronous is handy for making ar-
chives)

Ask:
·checking this item will cause DzDir to ask if a command string is ok before execu ←
    -
tion. You may manipulate this string before it is executed if you wish. The ask
procedure is much the same as the parse one. The only real difference is that
here the options are preset by you, with parsing, the options are defaults
made by me.

Output to list:
·Capture the output of an execution and put it in a list. (great for lha and ←
    picture view-
ers etc.). This feature requires that you have mounted "PIPE:". The output list
uses a monospaced font (lets you view columned text properly eg. an lha) and
does formatting of tabs etc.

Stack args:
·If this option is checked then DzDir will stack all the filenames one after the ←
    other
and call your command once, otherwise your command is called for each file
that is highlighted.

No Filenames:
·If you wish to use a button sheerly to run a command and don't care what the ←
    filel-
ists are doing when you run it, this is for you. (eg snake, screenblanker, syndi-
cate). This attribute is not applicable to an instant filetype Recognition as it ←
    is
an impossible situation.

CD Dest:
·If you wish for DzDir to be in the destination directory when your command is run ←
    ,
choose this option. If this is not checked, DzDir will change directories into
that of the source directory window. I added this so you can extract lha ar-
chives into the destination directory.

Read Src:
·Tells DzDir to reread the source directory after execution of a command. You must
set this for internal commands too.

Read Dst:
·Tells DzDir to reread the destination directory after execution of a command. You
must set this for internal commands too.

Positioning:

Move:
·Move the current button in the direction of the arrow.

Add:

·Add a new button on the side of the arrow.

Copy:
·Copy the button to the button in the direction of the arrow.

Delete:
·Delete this button.

X and Y:
·These strings contain the position of the icon, this data is saved to your icon's ←
    data
so it is advisable to use copies, as the position in the Workbench window will
alter.

Lists:
·This contains all the current aliases/filetypes, with the last entry "<New Object ←
    >"
there if you wish for a new appicon/hotkey. When clicked on, another appi-
con/hotkey appears at the bottom of the list, this process will repeat until you
run out of ram.

Delete:
·This button will delete the active entry.

1.6 DzDir.guide/Instant Filetype Recognition

                   This feature requires that you have the "WhatIs.Library" ←
                       installed on your sys-
tem, and is used for if you double-click the left mouse button on a file, or click ←
    the
right button. The "Filetypes" file in your S: directory can contain additional ←
   filetype
information, if you find a new datatype.
  Both the whatis.library and filetypes would have been presented to you at in-
stall time.

  This window contains all the attributes for the filetypes.
See
             Input methods
              For the specifics of the attributes.

1.7 DzDir.guide/ASL

ASL
  This library is responsible for the standardisation of AmigaDos requesters. It
was invented by Commodore to standardise any requests made by programs. As it
has been the standard form V2.01 I choose to use it. You may patch it to Req-
Tools.library or MagicFileRequester if you wish, but I use ASL for the good of ←
   future
compatibility.

1.8   DzDir.guide/Other Edit Window

                        This is where you setup your appicons and hotkeys. You may give ←
                          this appi-
con or hotkey an alias which will be used in the list object.
  To generate an Hotkey just don't specify an image. Hotkeys range from a "f"
(without the quotes) to a "control lshift ralt p". You may create one in the ←
   hotkey
string or by pressing the sample button DzDir will sample your keyboard for accept ←
  -
able hotkeys, you may end this by pressing "esc" or by closing the Hotkey window.

See
                Input methods
                 For the specifics of the attributes.
  You may use a appicon as a hotkey if you dont specify a
Icon.

1.9   DzDir.guide/Main Configuration window

  This should be the first window that you look at, as it allows great control ←
     over
your buttons. The Dirwin(0/1) sliders are how many buttons you want at the base of
each directory window. The Row(0/1/2) are for your dock window's buttons. These
sliders represent how many buttons are to be place horizontally in that row.
  Mem Gauges specifies where you wish to have your memory gauges (C=chip
F=fast A=all) This may be None, Dock window, or Path window (the path window is
the one with the volume list in it)
  Date Style, choose from 4 date formats. If you desire a format that isn't there,
just tell me.
  QAG can be used to change the amount of ram devoted to the Quick Access
Group, if it is 0 then the QAG list will disappear.
  The Qualifier cycle gadget tells DzDir what qualifier to use when you want to
find a file. This is used when you are looking at a filelist and you want to see ←
   file
"Startup-sequence" but you have 1001 files in the s: directory, simply hold the ←
   quali-
fier down and slap a "s" key. Automatically the first file begining with "s" is ←
   high-
lighted and brought into view.
  Close progress win, tells DzDir if it should close the progress window after an
action.
  And last but not least, The Open Dirwins at startup, which if unchecked will
make DzDir act like a dock.

1.10   DzDir.guide/Arexx

Arexx support:
  At this stage DzDir only offers a very limited arexx interface. This is because
of my limited knowledge of the language itself. If you see a way to improve this ←
   area,
please tell me.

Current commands are;
· dirwinsopen: Open the directory windows.
· dirwinsclose: Close the directory windows.
· fileypeswinopen: Open the Instant Filetype Configuration window.
· iconinfo: bring up the icon view window.
· edit: toggle edit mode on/off.
· hotkeywinopen: Open the edit hotkey window.
· about: About DzDir, registration cost etc.
· appiconwinopen: Open the Appicons configuration window.
· filesize: Obtain the size of the highlighted material.
· selectall: select all the files
· selectnone: select none of the files
· selecttoggle: toggle the selected files.
· sortname: Sort the directory window's contents by their name.
· sortsize: Sort the directory window's contents by their size.
· sortdate: Sort the directory window's contents by their creation date.
· dirsfirst: Sort the directory window's contents with directories always first.
· dirslast: Sort the directory window's contents with directories always last.
· dirsmixed: Sort the directory window's contents with directories intermixed
with the files.

1.11 DzDir.guide/Path window

   This is where all of your paths/volumes appear. If you put a disk in "DF0:"
then it will appear hear as well as on the Workbench. You may get the contents of
the root directory of that disk in the filelists simply by either dropping that ←
   file in the
window you want or by double clicking the volume you want from the volumes list.
   The QAG (Quick Access Group) list is also in this window, if the memory as-
signed to the QAG in the main configuration section is not 0. To load one of these
simply double click on it.

1.12 DzDir.guide/Requirements

  V2+ of AmigaDos
  1+ megs of ram (I haven't tested DzDir but I have estimated this value, may
    need 1.5+)

These requirements are subject to change as MUI and DzDir enter later versions.
The recommended system is

  V3+ of AmigaDos
  2+ megs of ram
  Hard Drive
  68020+ CPU
  24 Bit (grin)

1.13 DzDir.guide/Filtration and Searching

You may filter in four main ways:

Files & Directories: This is only valid with Filters, and tells DzDir which, if ↩
    any, file-
type you wish to apply this Filter to.

Name: just choose a preset name, or put in an AmigaDos style wildcard, eg. F#?. ↩
    info

Size: You may choose a minimum and maximum size to Filter/search on or an exact
value, this size is in Bytes

Date: You may choose a minimum and maximum date to Filter/search on or an exact
value, This age is in Days relative to today Ie. 1 is yesterday, 6 is today last
week

Datatype: You may wish to filter out all files that are not IFF's but they dont ↩
    have a
.IFF tail. Just pick IFF and thats all the list will accept

1.14 DzDir.guide/Registration

Simply send Aust$45 to:
    Ben Martin
    Po box 2131
    Southport Queensland
    Australia 4214

  This money can be either cash or a money order. It is up to you to weigh up
the risk of sending cash against the additional price of a money order. Yes folks ↩
    you
are looking at a product from another student, and if I had a job and a big wad of
cash I would probably have given DzDir away, but I require money (SAS/C 6.51,
more Hard disk, etc!!). I won't be changing the fee to $0 or even lowering it (↩
    those of
you worried about losing out). The only variation you may see is an upward one.

  When I receive your registration, if a later DzDir exists I will race you off a
copy with a letter. If not, I will send you the same version. This is done so ↩
    everyone
gets a letter and a disk. I may include versions (possibly beta) of some other ↩
    things
that I am working on, these may be a grep program, a Pim(Personal Information
Manager, like a FiloFax), or a screenblanker. The blanker works ok, but it needs a
rather rare dongle to run (The Retina 24 Bit board).

Don't forget to tell me these facts;

your name/address
your occupation (if you wish)
your machine type
the version of DzDir that you are using. (Very important)
your interests (if you wish. This is mainly to see what type of people use Amigas
these days)

any criticism/praise or new ideas you have (perhaps the most important part)

1.15 DzDir.guide/Future

   The unsure future of DzDir lays...In your hands. I can't devote all my time to
its generation anymore. And I myself am quite happy with it, so I shan't be adding
much to it. I do have a few ideas but I'm almost out of 'em, for DzDir atleast.

Current Ideas;

Arexx "path" command that takes an AmigaDos path and loads it into the active win-
dow
Arexx "activate" command for you to set the active window.
Perhaps a picture viewer, but to compete with the ample Pd (public domain) ←
    software
would be hard and fruitless. However I have just writen a viewer that uses
MUI 2.1's Bitmap class, it only took two nights so it wasn't that hard. If you
complain enough (once) then I'll add it. I also have a version for the retina, but
it is very much the same as "RetinaDisplay" for speed. The viewer I have only
views IFF's with less depth than halfbright and doesn't view Ham properly.
(you try viewing a HAM on the workbench?!?)
If these is a relatively easy way to pick up a file from the filelist and drop it ←
    some-
where else then that will be added for sure. Simulation of Appevents doesn't
seem easy at all, but I shall keep looking.

   If you have any ideas that you think a "Directory" utility should have then
please tell me. And most importantly, REGISTER!

1.16 DzDir.guide/Internal

#Active>->Other:
  Copy the active filelist's directory to the other filelist.
#Active<->Other
  Exchange the directories of the filelists.
#Add Icon:
  Add the correctly oriented icon to the selected files.
#Configure DataTypes:
  Bring up the datatypes configuration window.
#Copy:
  If a file exists with the same name then you may rename it on the fly, this fol-
lows the rename function's parameters.
#Delete:
  Can convert protected files to non-protected ones on the fly.
#Directories first:
  Sort he filelists with directories first.
#Directories last:
  Sort he filelists with directories last.
#Directories mixed:
  Sort he filelists with directories placed to comply with your "sort by" field.
#Edit Icon:
  Change the tooltypes, view the icon image, etc. The same as the Workbench

```
  "Icon/Information..." command.
#File Size:
  Calculate the size of the selected files/directories.
#Left Action:
  The same as double clicking on a file.
#Makedir:
  Just asks for the new directory's name and makes it.
#Make softlink:
  Make a softlink from the selected file(s), DzDir will ask for the link's name.
#Move:
  If a file exists with the same name then you may rename it on the fly, this fol-
lows the rename function's parameters.
#New Shell:
  Open a shell in the directory of the source directory window. You will need a
file in S: called DzShell-Startup.
#Dir Windows:
  Open the directory windows.
#Parent:
  Display the parent directory to that which is in the current directory window.
#Parse Files:
  Ask for a command and execute it on all selected files.
#Pattern files:
  Filter the filelists according to set perameters.
#Quit:
  Exit the program.
#Rename:
  Will check if a file exists with the given name, if so it will reprompt you for  ←
      a
new name. DzDir will do this until a filename is reached or you choose to re-
place the file with that name.
#ReRead Directory:
  Update the active filelist.
#Right Action:
  The same as right clicking a file.
#Root:
  Display the root directory to that which is in the current directory window.
#Search:
  Search the selected directories for certan files.
#Select all:
  Highlight all of the files in the active directory window.
#Select none:
  Highlight none of the files in the active directory window.
#Select toggle:
  Highlight all unhighlighted files in the active directory window and unhighlight
the ones that are currently highlighted.
#View Text:
  View a selected file with the output capturing method of DzDir.
```