

DzDir

COLLABORATORS

	<i>TITLE :</i> DzDir		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 16, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DzDir	1
1.1	DzDir.guide	1
1.2	DzDir.guide/Editing Dock Windows	2
1.3	DzDir.guide/Legal	3
1.4	DzDir.guide/DzDesktop	3
1.5	DzDir.guide/Default Configuration File	4
1.6	DzDir.guide/Filelist	4
1.7	DzDir.guide/Features	6
1.8	DzDir.guide/Dock Windows	7
1.9	DzDir.guide/Buttons	8
1.10	DzDir.guide/Editing Buttons	8
1.11	DzDir.guide/Input Methods	9
1.12	DzDir.guide/Instant Filetype Recognition	12
1.13	DzDir.guide/Editing Instant Filetype Recognition	12
1.14	DzDir.guide/Instant Filetype Creation	13
1.15	DzDir.guide/ASL	13
1.16	DzDir.guide/Object Edit Window	14
1.17	DzDir.guide/Object Edit Window	14
1.18	DzDir.guide/Main Configuration window	14
1.19	DzDir.guide/Arexx	16
1.20	DzDir.guide/Requirements	17
1.21	DzDir.guide/Filtration and Searching	17
1.22	DzDir.guide/Registration	18
1.23	DzDir.guide/Future	18
1.24	DzDir.guide/Internal	19
1.25	DzDir.guide/Grep	22
1.26	DzDir.guide/History	22
1.27	DzDir.guide/Concept	23
1.28	DzDir.guide/Installation	23

Chapter 1

DzDir

1.1 DzDir.guide

What is DzDir?

Simply: A really good directory utility.

Is it crippled?

NO!

The shareware version of DzDir is **Fully** usable and configureable. However you are encouraged to register if you want to see new versions.

Very important information:

All input methods that use files, use their own filelist in a new window. This is much like the concept of having temporary sets with databases. It also makes all actions able to be 100% async, while retaining the familiarity of the two window directory utility. And you get to see what the children tasks are doing. Perhaps an example; when you grep files, the active list's entries are cloned to another window, you can then use the original filelist for other action(s). This new window belongs to another MUI application, therefore you could minimise this window before continuing with the rest of DzDir. The grep application will continue greping its files while minimised.

Introduction

Concept

Features

System requirements

Installation

Default configuration

Editing DzDir's configuration

- Dock Windows
- Main configuration
- Instant filetype recognition
- Objects
- Buttons

Using DzDir

- Arexx
- Buttons
- Filelists
- Grep
- Objects
- Input methods
- Instant filetype recognition
- Instant filetype creation
- Internal functions

Discussion

- Future
- History
- ASL

Legal

- Registration
- Warranty and Distribution

Happy file management!

1.2 DzDir.guide/Editing Dock Windows

Name

The title of this dock. It appears in the window's title string. You may open or highlight this dock by telling dzdir its name. If you give a string of length=0 `↔` then the current time and date is displayed in this dock's title.

Buttons

How many buttons you want in this window.

Rows/Columns

How many rows or columns you want this docks buttons to make.

Window Frame

Should this dock have a window frame.

Button Frame

Should this dock's buttons have a frame.

Virtual group

Put this dock's buttons in a virtual group with scroll bars to move around it.

If there are more buttons in the window than could fit on the screen then some will not be viewed. Dock's do remember where you were when saved and every consequent load the dock will appear scrolled to that position.

Always Open/Follow FileLists/Openable on Request

How should this dock be open, Always, if and only if the filelist docks are open or only loaded when asked for by name.

You can choose to have a dock window toggle its open state with the filelist windows. ie. If you close the filelists, that dock will also close. If you open the filelists that window will open too. This is done by the "Follow Filelists" option.

Save

Save off this dock. If this is a filelist dock then both filelist docks share one template file so you have uniform filelist docks.

Cancel

Forget the changes.

1.3 DzDir.guide/Legal

The legalities:

No warranty is implied for the accuracy of this program or documentation, no responsibility will be accepted for anything this program does either directly or indirectly. You may copy the shareware version as long as you charge less than \$5 for it & the archive is intact. Fred Fish may distribute the shareware version on his seedy Roms, and any pd/shareware cd-roms may include the shareware version too. You are permitted to try this product for up to one month then you must register it to continue use.

1.4 DzDir.guide/DzDesktop

DzDesktop is a collection of utilities put out by Digital Zenith to enhance Amiga-Dos and speed its operation. It's collection is aimed to help the user to manipulate

their files, view and edit them, and in general slip stream this process. ↵
 DzDesktop's
 programs are object oriented 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 object in an active window (e.g. a list, button or gadget) and tap the help key ↵
 .
 There is also short help setup for many objects.

At present the only parts of DzDesktop that have been implemented are DzDir,
 which is a directory utility, & DzPim, which is a personal data manager. Other pro ↵
 -
 grams are in the pipeline, but after about many years in development, DzDir was
 ready to grace the public.

1.5 DzDir.guide/Default Configuration File

The default configurations supplied sway very heavily toward a 3.x machine
 (as it turns out you need AmyDos 3.0+). The main area of concern is in the instant
 filetype recognition, where most files are set to "Sys:Utilities/Multiview", which ↵
 obvi-
 ously isn't the best configuration for a personalised version of DzDir. Once ↵
 installed,
 you can easily select particular filetypes and alter them (via edit/Datatypes in ↵
 the
 menu).

If you are a novice we strongly recommend that you use the "text button" con-
 figuration. For the easier interpretation.

We will just finish by saying that these config files provide just a taste of ↵
 what
 you can achieve with DzDir, you will have to take control of personalising your ↵
 copy.

1.6 DzDir.guide/Filelist

The filelists work with all input methods (eg. appicons, hotkeys, buttons) on a
 last active filelist basis. If you highlight a filelist's window then that window' ↵
 s filelist is
 the active filelist, and thus the source of filenames. There is a small button ↵
 above the
 filelist that is selected when that list is the active one.

You can configure a qualifier (in menu select Edit/Main), then when you have
 a filelist active (and its window) if you hold that qualifier down and type an "S" ↵
 (no
 quotes) then the first file beginning with and "S" will be shown in the filelist. ↵
 If you keep
 the qualifier down and type a "Z" then the first file beginning with "SZ" is shown. ↵
 You
 can keep concatenating letters for a more defined selection until you deselect the
 qualifier, then the process is repeated.

You may change the way the filelists are sorted either by the lists menu or by
 returning the internal function "sort by ..." (this can be done by either setting ↵
 up a but-

ton, hotkey or appicon). This method then becomes the new preference. Which is permanent provided you save that configuration.

Both filelists are susceptible to appevents and will load the contents of the directory or file that is dropped on it. If a file is dropped on the filelist then the file's directory is loaded, eg. "work:finalwriter/fw" will load "work:finalwriter".

We have also provided drop & drag between filelists, their paths, and buttons. To drag the selected files simply pick an entry, hold the left mouse button down, move the mouse to the destination, and release the left button to drop the files;

- If you release the left button over the other filelist then the selected files are copied there. If you hold down the shift key while doing this, DzDir will ask you what to do.
- If you release the left button over the other filelist's path string (not in the filelist, but in the string object above it) then the selected object's directory is loaded there. I say "object" because if you drop "Sys:Libs" from "Sys:" then "Sys:Libs" is the directory that will be read.
- If you drop these files on a button then that button is run on those files.
- And if you drop the files back in the same filelist then one of two things will happen, if you dropped them in the same directory then nothing will happen. But if the position you drop them at's entry directly above is a directory then these files will be either copied or parsed there depending on the shift key (as in first point).

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. There are two actions per filetype, a left and right action, they are invoked by double clicking either of the mouse buttons. These actions also work from thread filelists (eg the search output list, the files being greped).

Filelists are usually the source of filenames for executables. There are two exceptions; If you have checked the "Ignore arguments", or if the directory windows aren't open (eg. If you have setup a button to run a text editor then if the filelist's window is open you are passing files to be edited, whereas if these windows are closed the editor is run with no files to edit).

Information about files and directories is displayed at the top of each directory window, preceded by an "r" if reselect as available. This is in the format "selected dirs/total dirs selected files/total files selected amount/total amount". The scale for the amounts varies for ease of reading, it can be in bytes(B), kiloBytes (kB), Mega-Bytes(MB) or GigaBytes(GB). This size information 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 "File/File size" from a dock window's menu. The screen title for each filelist's window contains the free space on that disk, and information about any filter you may have applied.

To change a files protection bits you can simply double click on that bit in the filelist and it will be toggled.

1.7 DzDir.guide/Features

DzDir's current features include;

- * Uses MUI for maximum future benefit, and user enjoyment/configurability.
- * All execution is done 100% Asynchronously from the main program, including internal functions!
- * Async reading of directories and lha files.
- * Command line size dynamically allocated from 0-1024 bytes (DW 1.62 40b, DirOpus 4.11 200b).
- * Unlimited AppIcon support (only limit is set by your RAM).
- * Unlimited Hotkey support (only limit is set by your RAM).
- * Unlimited configurable buttons. (only limit is set by your RAM)
- * All of these buttons may be configured simultaneously if desired.
- * Multifformat Date/Time display.
- * Unlimited RAM/CPU gauges and or levelmeters.
- * May open without the Directory windows (ie. Dock(s) only).
- * System friendly capture of output (no console windows).
- * Ability to view any file(s) from within a Lha archive (provided you have Lha V1.32+ in your path, have mounted PIPE:, and assigned T: somewhere).
When you double click on a lha with the left mouse button the archive is expanded as if it were a directory. You can view an Lha from within an Lha, with no limit as to how nested you can get. It is strongly recommended that you only read from nested Lha files. (as they are files in "T:" and autodeleted)
- * Ability to use most standard internal file manipulation routines on files within an Lha. You can Delete, Copy, Move etc. As though these files actually existed!
- * 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).
- * Custom drag and drop from/to many objects.
- * Instant file type recognition (via the WhatIs.Library).
- * Instant file type creation (only limit is set by your RAM).
- * 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.
- * Arexx Support.
- * On-line Help (via AmigaGuide).
- * Information strings showing the size/amount selected for files.
- * Uses Workbench's "Icons/Information..." Window to edit icons.
- * Ability to put either an icons image, text, gauge, levelmeter, imported 8 bit icon, or an IFF image in a button.
- * Left, middle, Right mouse buttons may be configured to do different things to a filetype. (eg. view/extract a Lha)
- * A wide array of
 Internal
 functions.

- * Ability to move buttons horizontally and vertically, Add a new button to the left/right, copy the button anywhere, swap the button with another, clone the button, and delete any button.
- * Drop And Drag button configuration.
- * Auto-tiling of buttons.
- * Internal Copy command, complete with internal progress gauge. (settle down...)
- * You decide if the source(src) or destination(dst) filelists are to be updated after all actions. (power!)
- * DzDir can be in either the source or destination directory, great for extraction of lha files into the destination directory.
- * Cloning and swapping of the filelists contents.
- * Very descriptive information displays so you too shall know what DzDir and its tasks are doing.
- * Quick Access Group (QAG) list. This keeps a list of previous directories for reloading later. Also displays all parents to root, and any default paths you desire.
- * Ability to create softlinks.
- * Qualifier + key to find a file in the filelists. (currently unavailable)
- * Filtration based on filename, a range of size, a range of days old, exact size, or exact age, or filetype (Via the WhatIs.Library) can easily be implemented.
- * Can search multiple directories recursively for any of the Filtration parameters and display a list of the matching files.
- * Can recursively scan directories
 - Greping files for regular expressions.
- * Most Configuration occurs in realtime.
- * Popup Docks may be setup on certain buttons (those buttons that aren't in a popup dock themself) containing many buttons. These docks are activated by holding the mouse down over a button for longer than 0.2 secs. If there is no popup dock a message will appear in the screen title informing you. You can also tell DzDir to dealloc this popup dock so ram usage is kept down. When a popup dock is required, it is loaded quickly from disk.
- * Dock windows can remain uncreated and unallocated until requested via a hotkey, button, appicon, or arexx command.
- * Rows of buttons can be put in custom virtual classes so huge amounts of buttons can be made on a 640x200 screen.
- * CPU usage monitoring.
- * If you double click a file or directory's attribute bits they are toggled on or off.
- * You can predict a DzDir spawned task's priority.
- * New Skip all option, for eg. when copying, files exist, skip all that do.
- * And many more things that we are so used to that we have temporarily forgotten.
- * Ability to reselect files that you acidently unselected.
- * Jump to a file in filelist by selection string on the fly (eg. Hitting qual + "ea" would scroll to the first file begining with "ea")
- * Oh yeh... And a map of the World!

1.8 DzDir.guide/Dock Windows

You may have an unlimited amount of dock windows. If you desire another then choose "Dock/New" from another dock window.

Popup dock windows are by default in vertical orientation in virtual mode. This allows you to have 256 buttons in a popup and page through them, handy for font selection etc.

See

Editing dock windows
for attributes you can edit.

1.9 DzDir.guide/Buttons

DzDir allows you to have any amount of configurable buttons. These are either at the side of the directory windows or in dock window(s).

If you drop a button over another one (not itself) then the two buttons are swapped, if you are holding down the shift key then the source button is copied there.

If you drop a button on itself then it is cloned next to itself, if you are holding down the shift key it is deleted. More advanced D&D is planned for this area, but as there

are no registered users this will be done in time.

These buttons are broken down into any amount of docks that have of up to 255 buttons, each of which can have 255 buttons in a popup dock!

A button may contain text, an icon, an imported 8 bit icon, a Levelmeter monitoring some resource, a gauge monitoring some resource, or an IFF image depending on its configuration.

All buttons, are susceptible to appevents (dropping files on the button from workbench).

A button can either execute an AmigaDos command, return an internal function, load a desired path into the active directory window, open a dock window, ex-

ecute a series of Arexx commands, open a special popup dock, or pass control to another object.

This is dependant on the way you have configured that button.

1.10 DzDir.guide/Editing Buttons

To configure a button simply right click the button(s) you wish to configure, It will pop a configuration window up for itself. You can edit all of the buttons simultaneously if you wish. For any buttons new attributes to take effect you must ok that button's edit window.

See

Input methods

For the specifics of the attributes in this edit window.

1.11 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 objects. This is done so you can easily find the object you desire. The name field of an appicon is also the text below the appicon. In DzDir V8.1+ this is also used in the filetypes configuration to see what to do to a file. As of V8.1 this filetypes are relational and you setup all actions in the object editor, then in the filetypes editor you simply choose from a list of objects which to do when you left/right/middle click a file. (more on later this node)

Either button Text/Icon/IFF/Import/Meter/Gauge name rollo

- This rollo controls the button type;
 - * If text you receive a string gadget for the name.
 - * If Icon or IFF you also get a ASL-Pop object. Select the name of the IFF/Icon for DzDir to load each time the button is needed.
 - * If Import give the name of an 8 bit icon to import, after this is done, leave the name string there, but the icon itself is no longer needed. DzDir stores the imagedata itself, so faster loading and less disk thrashing is achived.
 - * If meter or gauge then you simply choose what system resource you want DzDir to show you in this button.

Icon

- If you are editing objects you can give them an appicon, 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/Dock/Rexx/Popup/Object.
 - * If you choose Executable then you will get (from left to right) a string gadget a ASL-Pop object and a pop-list object. If a command 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 {F} where you want them, if there is no {F} in your line the names will be put at the end. The one {F} is equal to all the filenames if "Stack arguments" is checked. Such flags as {F}

may be obtained by the pop-list object. DzDir will try to lock your command file, if it can't then DzDir will alert you, from there you may either proceed or halt execution. If you choose to go ahead anyway then DzDir has done its best not to crash your machine. (Write to the future owners of AmigaDos and get them to improve "system()" if you need better execution).

If you select Recursive filtering for this object then my exe parser will call your program on matching files in any directories encountered, so for example you could show and files matching a filter of type=IFF in dir art:clipart/ with 'U:VT

*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.

*If you want a drive button, simply choose path. Then the string is what path is loaded into the active filelist when this button is pressed.

*If you wish for this object to open a "openable" dock window(s) then choose "Dock" and put the "name" of that window here, the popup will list all the windows you can choose from.

*And if you choose rexx, you can string a rexx script together to do many things. A leading comment is added automatically for you. An example is, address command; 'run >NIL:' 'U:priman';wait 1;address DZDIR.1;opendock "AMP" where U:Priman is run, then a one second wait before opening the dock "AMP" which could have buttons to control PriMan. All REXX scripts are run Async so you can interact with DzDir.

Hotkey

•The keyboard short-cut for this button. (The hotkey is in the form of "lalt lshift d" etc).

There is also a cycle gadget here, it toggles between a window and global style hotkey. A window hotkey only works when you have a dock window or filewindow active (good for delete, so you don't do it from DPaint!), where a global hotkey still works if you are playing a game (given that the game is OS friendly!).

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.

Flags

Asynchronous execution:

•If this is checked then DzDir's spawned execution task will run the command asynchronously (not wait for the command to return) otherwise this task will wait for the command to return before it terminates. Note, DzDir runs these "synchronous" operations in a thread anyway, but this is good for a batch job where you want the thread do files one at a time.

Ask before execution:

•checking this item will cause DzDir to ask if a execution is really fine before doing it.

You may manipulate this object's action 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 DzDir.

Output to Window:

•Capture the output of an execution and put it in a list. (great for picture viewers 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 arguments:

•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 once for each file that is given.

Ignore arguments:

•If you wish to use a button sheerly to run a command and don't care what the filelists are doing when you run it, this is for you. (eg snake, screenblanker, syndicate).

Cd into the destination:

•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. We added this so you can extract lha archives into the destination directory. Or if a program opens its requesters in the directory it was run from.

Rescan source:

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

Rescan destination:

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

Open Shell Window:

•Tells DzDir to execute your command from a shell. This attribute overrides both the "async" and "output to window" flags, as all shells multitask, and window output makes no sense with this flag. After execution the shell will remain open for a second and then automatically close.

Recursive filter:

•Tells DzDir to apply the current filter options to any recursive files found. Great for greping all '#?.C' files in Source/ where that dir has files like obj.c.old etc. Or for viewing each JPEG or GIF datatype made over the last 7 days that is less than 40K and has 'ee' in its name.

Open progress window:

•For this task do you want to see a file/dir progress in a seperate window. The progress window has two important buttons; one with a stop sign to halt that action, and the other is a toggle for recursive filtering.

Other things you might see...

Priority

•If a subtask is spawned, what should its priority be. This effects internals/exe/rexx

actions

Buttons

·How many buttons this object has in its popup dock.

Sort popup dock

·Sort this button's popup dock.

1.12 DzDir.guide/Instant Filetype Recognition

This feature requires that you have the "WhatIs.Library" installed on your system, and is used for if you double-click the left or right mouse button on a file. The "Filetypes" file in your S: directory can contain additional filetype information, for if you find a new datatype.

Both the whatis.library and filetypes file would have been presented to you at install time.

[Click here](#)
to see how to edit this section of DzDir.

1.13 DzDir.guide/Editing Instant Filetype Recognition

To edit the action taken on a filetype choose "settings/Filetypes..." from the menus. This will give you a list of available filetypes and the action that will result from a double click. To determine the filetype of a file just drop it in the list of filetypes.

Parameters of the FileTypes window...

Types list

This list has all the filetypes DzDir recognises.

Mouse clicks

The name of an object that dzdir will run when you double click a file, of the type selected in the type list, in any filelist. The pull down button will show a list of all your objects to make selection simple.

SendTo

You can add and delete a type specific list of objects that can later be accessed through a Popup/Sendto configuration in a button.

Save/Load

Save or load the filetypes database.

1.14 DzDir.guide/Instant Filetype Creation

To take advantage of this feature requires that you have the "WhatIs.Library" installed on your system, and, after implementation, is used for if you double-click the left or right mouse button on a file. The "Filetypes" file in your S: directory will contain the additional filetype information.

Both the whatis.library and filetypes would have been presented to you at install time.

To make a new filetype;

Pick a bunch of file(s) of this new alien type.

Return the "Make Filetype" internal action, or choose "File/Make type"

Choose "Test" to see a hexadecimal correlation list.

Choose a name for the new type, and enter it.

Choose "Make" to add the type.

The correlation is based on the selected files, simply, the more zeros you see the less correlation. You really only need, say, 4 sequential bytes or so for a fairly decent filetype. DzDir will however use all the first 480 bytes in its tests, and write out any correlation that exists where 4 bytes in a row are identical.

If you make a new Filetype with only one sample file you really need to know what you are doing. This type may be good, but it is not usually the best case. Generally the more files of the new type the better. However we have not restricted you to have to use more than one sample file.

If DzDir finds *no* correlation in the first 480 bytes it will tell you that there was no new type. There is a fairly scarce chance of this, but it may happen. If it does then you will have to make the type manually.

Due to the way the WhatIs.Library is setup, you will have to quit DzDir and flush the library in order to see this new type from DzDir. You can use 'avail FLUSH' to do this providing that no other program is using the Whatis.library. We Hope to change this inferior loop in the future. Just tell us you want is to change so we give it a higher priority.

1.15 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 we choose to use it. You may patch it to ReqTools.library or MagicFileRequester if you wish, but we use ASL for the good of future compatibility.

Unlike some word processors...

1.16 DzDir.guide/Object Edit Window

This is where you setup your appicons, hotkeys and what actually gets executed when you double click filelists. You may give this appicon or hotkey an alias which will be used in the list object, and to link these to filetypes in the filetypes editor.

The reason for having objects abstracted from filetypes is so you can alter many filetypes actions in one go. You save time and RAM. Its a little hard to explain but it has to do with normalization of databases and relational databases. For the context of DzDir, you setup an object to be "TTX" and call it "Text Edit" then go into the filetypes editor and happily match "Text Edit" to "Txt,Doc," etc. Later when you get a different editor you can alter the object "Text Edit" and any filetype references you made to "TTX" through the object "Text Edit" will now run your new editor.

To generate a Hotkey just don't specify an icon. 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 acceptable hotkeys, you may end this by pressing "esc" or by closing the Hotkey window.

Click here
to see how to edit this section of DzDir.

1.17 DzDir.guide/Object Edit Window

Commands

All the names and commands of the objects you have.

Entry

New Add a new object to the list

Copy Copy the active object in the commands list.

Delete Delete the active object in the commands list.

List

Save/Load/Sort: Mail me if you don't understand.

See

Input methods

For the specifics of the "attributes" section.

1.18 DzDir.guide/Main Configuration window

Dir Windows

- Open Directory windows at start-up, which if unchecked will make DzDir act

like a dock. If selected then the directory windows are opened at startup.

- 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 qualifier down and slap a "s" key. Automatically the first file beginning with "s" is highlighted and brought into view. This option is currently not in use.

History

- This list contains paths that are to be displayed in the pull down list from ↔ near the filestring that is just above a filelist. These are generic paths that you ↔ want access to alot.
- The add and delete is self evident.
- Max history, when a directory is read into a filelist the old directory name is put into a buffer, this slider is how manu entries this buffer can contain.

File I/O

- Copy Buffer Size, how much file to Read/Write in one go when copying. A larger buffer makes coping faster particularly to/from a floppy. If This amount of RAM is not available then DzDir will try progressively smaller blocks before telling you of its delemor.
- Refresh Workbench's Icons, If you select this option DzDir will use DiskObject() calls on icons, so workbench's windows are updated in real time. This is an option because it slows things down when operating on icons.
- Free Popup Nodes: if selected, when a popup dock is closed the RAM is freed so you always have your RAM, the only down side is if you want the popup dock it has to be loaded from disk again.

SendTo

- You can add and delete generic nodes that will appear in DzDirs sendto lists. These nodes are referances to Objects.

TextView

- Auto Hot View: A amigados pattern containg paths that ViewText will consider 'hot' see

View Text

for more information on heat.

- Text View Buffer Size: how much ram should be allocated in one go by ViewText, larger buffers cause less memcpy() calls

Rexx

Rx Path, Rx Jumpto, and Rx Loadfile, are used for loading files into external viewers and jumping to a line. The only implementation at the moment is with grep.

- RxPath is where 'rx' is to be found.
- RxJumpto is called with the filename and line num, see 'ViewTextLine.rx' for more information.
- RxLoadfile is called with the file to load.

Time

- What date style to use

NewShell

- Window definition of a shell opened by DzDir.

1.19 DzDir.guide/Arexx

At this stage DzDir only offers a very limited arexx interface. This is because of our limited knowledge of the language itself. If you see a way to improve this area, please write to us. ↔

Current commands are

- Dirwinsopen
Open the directory windows.
 - Dirwinclose
Close the directory windows.
 - Fileypeswinopen
Open the Instant Filetype Configuration window.
 - Iconinfo
Bring up the icon view window.
 - Edit
Toggle edit mode on/off.
 - 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.
 - Dirsfirsr
Sort the directory window's contents with directories always first.
 - Dirslastr
Sort the directory window's contents with directories always last.
 - Dirsmixed
Sort the directory window's contents with directories intermixed with the files.
 - GrepSetup
Open the grep window.
 - GrepName
-

What to search for.

- Grep
Start search on selected files in last active filelist.
- SelectFile
Select a file in the active file list.
- Read
Read a dir into the last active filelist.
- waitlist
Control will be passed back when the active filelist has finished reading.
- Opendock
you give the dock(s) name as the only parameter. This window(s) are opened.

1.20 DzDir.guide/Requirements

All of MUI's list and;

V3+ of AmigaDos
MUI 3.7+
1+ megs of ram

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

V3+ of AmigaDos
Latest MUI
2+ megs of ram
Hard Drive
68020+ CPU
Registered version...

1.21 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 don't have a .IFF tail. Just pick IFF and that's all the list will accept. You can of course filter on more than one type.

These ways of filtering are logically ANDed together, so you can filter on name a #? between 0-10K big, made 3 days ago that is a (JPEG, GIF, IFF or EPS).

1.22 DzDir.guide/Registration

Simply send Aust\$55 to:

Ben Martin
Po box 2131
Southport Queensland
Australia 4215

When your registration is received, you will be sent a disk with the latest DzDir full version on it. There may also be included versions (possibly beta) of some other things that Digital Zenith is working on, these may be for example; a Pim (Personal Information Manager, like a FiloFax), a isometric game, or a screenblinker. The blanker works fine, but it needs a rather rare and expensive dongle to run (The Retina 24 Bit board).

Please include the following;

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

and any criticism/praise or new ideas you have, perhaps the most important part.

1.23 DzDir.guide/Future

The unsure future of DzDir lays...In your hands. The author can't devote all his time to its generation anymore. And we at Digital Zenith are quite happy with it, so we shan't be adding much to it. We do have a few ideas but were almost out of 'em, for DzDir atleast. This might well be the last release if no support is given to DzDir.

Current Ideas;

More detailed arexx interface.

Better Drag And Drop for button configuration.

I-Net support

Perhaps a picture viewer, but to compete with the ample Pd (public domain) ↔ software

would be hard and fruitless. However we have just written a viewer that uses MUI 2.1's Bitmap class. If you complain enough (we don't think you will) then we'll add it. We also have a version for the retina, but it is very much the same as "RetinaDisplay" in terms of speed.

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

REGISTER!

; -)

1.24 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 icon that is setup for the object that a left click points at to the selected files. Also works for files in a Lha.

·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 follows the rename function's parameters. You can also replace all.

·CopyAs

Copy, but ask for a new name for each object.

·Delete

Can convert protected files to non-protected ones on the fly, and convert all. Skip all files that cause errors. Delete all directories selected.

·Directories first

Sort the filelists with directories first.

·Directories last

Sort the filelists with directories last.

·Directories mixed

Sort the 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.
 - File information
This is a internal which accepts flags. if you don't tell it to Ignore arguments then for each argument the filetype is shown. Also the src/dst directory selected file/dir info sizes of selected and free disk, together with the % Fit. The ← filewindows must be open for this command to work.
 - Grep
•Left Action
The same as double clicking on a file.
 - Mkdir
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.
 - Make filetype
Make a new filetype from the selected file(s), DzDir will ask for the types name, and display correlation.

see: Instant Filetype Creation
•Move
If a file exists with the same name then you may rename it on the fly, this follows the rename function's parameters. You may also replace/skip all files that ← al-ready exist.
 - MoveAs
Move, but ask for a new name for each object.
 - New Shell
Open a shell in the directory of the source directory window. You will need a file in S: called DzShell-Startup.
 - Open file 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 parameters. See filtering "Filtration"} for more about the filter class.
 - 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 replace the file with that name.
- ReRead Directory
Update the active filelist.
 - Reselect files
Reselects the files you had selected. eg. You have 100 files highlighted and are about to shift select another but, you slip on the shift key, and now have only one or two file(s) selected! So now you can "Reselect" what you had selected before the tragedy. If you reselect again you undo the previous reselect, working like a toggle.
 - Right Action
The same as double clicking the right mouse button on clicking a file.
 - Root
Display the root directory to that which is in the current directory window.
 - Search
Search the selected directories for certain files. Based on a filter class you tell DzDir what your looking for. See filtering for more about the filter class.
 - Select
Based on a filter class you can select/deselect/toggle files that match the specifications you set. See filtering for more about the filter class.
 - 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.
 - Touch Icon
If you extract a Lha, then the new icons will not have appeared in a Workbench window for that directory, if you touch them then they will appear in the workbench window in (near) real time.
 - View Text
View a selected file with the output capturing method of DzDir. If you make this viewer 'hot' then any files you request to load after now with a matching path will be loaded into this window, otherwise a new window is made.
-

1.25 DzDir.guide/Grep

You can grep files and directories for strings or regular expressions and DzDir will show a list of all the files matching the discription.

You can save and load the output of a grep action via buttons on the panel.

You can also select between fast and slow grep, the differance being that fast grep uses no pattern matching. If you hit return after entering a string the default is a fast grep, but if the string you entered has a '#?' in it the greping will switch to slow grep. You can also select which type you want from the GUI.

The slow grep regular expressions work like file pattern matching. For example e#? would find lines sarting wirth "e" or "E", and #?e would find lines ending in "e" or "E"

The patterns are fairly extensive, and approximate some of the ability of Unix/grep "regular expression" patterns. Here are the available tokens:

? Matches a single character.
 # Matches the following expression 0 or more times.
 (ab|cd) Matches any one of the items seperated by '|'.
 ~ Negates the following expression. It matches all strings that do not match the expression (aka ~(foo) matches all strings that are not exactly "foo").
 [abc] Character class: matches any of the characters in the class.
 [~bc] Character class: matches any of the characters not in the class.
 a-z Character range (only within character classes).
 % Matches 0 characters always (useful in "(foo|bar|%)").
 * Synonym for "#?", not available by default in 2.0. Available as an option that can be turned on.

"Expression" in the above table means either a single character (ex: "#?"), or an alternation (ex: "#(ab|cd|ef)"), or a character class (ex: "#[a-zA-Z]").

1.26 DzDir.guide/History

DzDir was born in the december of 1993. At the time the two directory utilities that I conidered the major players were Dw 1.62 and V4 of Dopus (I think it was V4) and these were (and are still) great directory management solutions. That said they were at either ends of a spectrum, Dopus being large and powerful, and Dw being very fast and small. But nothing was both powerful and resourse friendly. This is when I decided to write a program that used MUI and could supply very fast small

code, but also be very powerful and graphicly pritty.

So now in '97 I have this Object oriented program that although its large still runs fast and has very many tricks (eg. make a button a path list) because of its design. It also has a relational database inbuilt to make change to its configuration very easy and avoid the problems associated with a first normal format database.

Revision history was not kept over DzDir's youth, consequently it begins at version 9.10.

New in version 9.20; 2 Major bug(s) fixed.

- Fixed a buf in the Unique key generator of the object database
- If you try to open a dock that is to large for your screen then it becomes a virtual group in that dock so the window will open. This is very inefficient and you are better to then edit that dock and make it either smaller or virtual on a permanent basis.
- Qualified file showing is now ok. You can configure a qualifier (in menu select Edit/Main), then when you have a filelist active (and its window) if you hold that qualifier down and type an "S" (no quotes) then the first file beginning with and "S" will be shown in the filelist. If you keep the qualifier down and type a "Z" then the first file beginning with "SZ" is shown. You can keep concatenating letters for a more defined selection until you deselect the qualifier, then the process is repeated.
- Began to update the structure and contents of the documentation to be more reader friendly and imformative.

1.27 DzDir.guide/Concept

Because we spend so many hours getting data where we want it on disk and retrieving it form this known structure; doing this in a fast, elegant, powerful and graphically agreeable manner is what DzDir is made of.

The simple challenge DzDir tries to achive is giving the power of unix type systems to those who don't want to use a shell.

1.28 DzDir.guide/Installation

DzDir requires the Installer program from Commodore for its installation. Therefore installing DzDir is basic, just open the install script and away you go.
