

RO Documentation

Oliver Rummeyer

Copyright © CopyrightÂ©1994,1995 Oliver Rummeyer

COLLABORATORS

	<i>TITLE :</i> RO Documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Oliver Rummeyer	February 6, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RO Documentation	1
1.1	RO User Documentation	1
1.2	Introduction	2
1.3	System Requirements	4
1.4	Unregistered Version	4
1.5	Installation	5
1.6	Main Window	5
1.7	Register Group	5
1.8	Information Textfield	6
1.9	Listview	6
1.10	Path Gadget	7
1.11	Drive Buttons	7
1.12	Command Buttons	7
1.13	Status Bar	11
1.14	Menus	11
1.15	Configuration	13
1.16	Filetype Configuration	14
1.17	Custom Filetype Configuration	15
1.18	Archive Configuration	17
1.19	General Configuration	18
1.20	Menu Configuration	25
1.21	Buffer Configuration	26
1.22	HotKey Konfiguration	26
1.23	HotDir Configuration	27
1.24	Drive Configuration	28
1.25	Button Configuration	29
1.26	about	34
1.27	action	34
1.28	all	34
1.29	assign	34

1.30	byname	35
1.31	bytes	35
1.32	change	35
1.33	copy	35
1.34	copydev	36
1.35	current	36
1.36	delete	36
1.37	disk	36
1.38	dup	37
1.39	expand	37
1.40	fit	37
1.41	help	37
1.42	icon	38
1.43	info	38
1.44	active	38
1.45	lcopy	38
1.46	lswap	38
1.47	listarc	38
1.48	parent	39
1.49	reload	39
1.50	root	39
1.51	makedir	39
1.52	move	39
1.53	none	40
1.54	note	40
1.55	protect	40
1.56	quit	40
1.57	reboot	40
1.58	relabel	41
1.59	rename	41
1.60	setdate	41
1.61	shrink	41
1.62	toggle	41
1.63	touch	42
1.64	unarc	42
1.65	update	42
1.66	Arexx Interface	42
1.67	Discussion	46
1.68	Registration	46

1.69	Registration Form	47
1.70	Updates	48
1.71	Support	48
1.72	History	49
1.73	Future Enhancements	51
1.74	Acknowledgments	52
1.75	Disclaimer	52
1.76	License	53
1.77	Installer	54
1.78	Magic User Interface	54
1.79	index	54

Chapter 1

RO Documentation

1.1 RO User Documentation

RO --- A MUI-Based FileManager, Shareware

Version 1.04

User Documentation

Copyright © 1994, 1995 by Oliver Rummeyer

1. Introduction

1.1

Short description
.....What is 'RO'?

1.2

System Requirements
.....What you need to use RO

1.3

Unregistered Version
.....Is RO crippled shareware?

2. Installation

2.1

How to install RO
.....How is RO installed?

3. How to use RO

3.1

Main Window
.....How is RO controlled

3.2

Menus
.....RO's Pull-Down Menus

3.3

Configuration

-Setting up RO
- 3.4 ARexx Interface
 -How to control RO via ARexx
- 3.5 Discussion
 -Questions and Answers. (Q&A)
- 4. Registration
 - 4.1 How to register for RO
 -How to become a registered user
 - 4.2 Registration Form
 -I need your name, address, ...
- 5. Other Topics
 - 5.1 Updates
 -Where to get updates
 - 5.2 Support
 -How to reach the author
 - 5.3 Acknowledgments
 -The author wishes to thank...
 - 5.4 History
 -The RO history
 - 5.5 Future
 -If you register...
- 6. Distribution Politics
 - 6.1 Disclaimer
 -Warranty? No warranty!
 - 6.2 License
 -License agreement
 - 6.3 Installer
 -About the installer program.
 - 6.4 Magic User Interface
 -About Magic User Interface

1.2 Introduction

Short description

RO is a new intuition-controlled, fully user-configurable file manager for OS 2.04 and newer. RO's mighty features will assist and help you on your daily work with your Amiga. With RO there is no need to learn those cryptic shell commands, because you have the complete power of your Amiga in one hand.

Due to the usage of MUI you are able to configure the look and feel of this tool as you like it. MUI is the

Magic User Interface

developed by

Stefan Stuntz which should be installed on every Amiga. It is an object oriented system to create and maintain graphical user interfaces. Users of MUI based applications have the ability to customize nearly every pixel of a programs interface according to their personal taste.

The RO archive should contain the following files:

```
RO/ARexx/Action.RO
RO/ARexx/AddArc.RO
RO/ARexx/ArcHandler.RO
RO/ARexx/Iconify.RO
RO/ARexx/LoadDir.RO
RO/ARexx/LoadOther.RO
RO/ARexx/PlayModule.RO
RO/Binary/RO, binary
RO/Config/RO.prefs, default
RO/Config/RO.prefs, example1
RO/Config/RO.prefs, example2
RO/Config/RO.prefs, example3
RO/Config/RO.prefs, example4
RO/Config/RO.prefs, example5
RO/Docs/RegForm, deutsch
RO/Docs/RegForm, english
RO/Docs/RegForm, francais
RO/Docs/RO.guide, deutsch
RO/Docs/RO.guide, english
RO/Docs/RO.guide, francais
RO/Icons/4col/Drawer.info
RO/Icons/4col/RegForm.info
RO/Icons/4col/RO.guide.info
RO/Icons/4col/RO.info
RO/Icons/8col/Drawer.info
RO/Icons/8col/RegForm.info
RO/Icons/8col/RO.guide.info
RO/Icons/8col/RO.info
RO/Install.info
RO/Install/Installer
RO/Install/RO-Install
RO/Install/RO-Install.info
RO/LiesMichZuerst
RO/LiesMichZuerst.info
RO/LisezMoiDabord
RO/LisezMoiDabord.info
RO/Locale/Catalogs/Deutsch/ro.catalog
RO/Locale/Catalogs/Italiano/ro.catalog
RO/Locale/Catalogs/Français/ro.catalog
```

```
RO/Locale/RO.cd
RO/Locale/RO.ct
RO/Previews/Preview1
RO/Previews/Preview1.info
RO/Previews/Preview2
RO/Previews/Preview2.info
RO/Previews/Preview3
RO/Previews/Preview3.info
RO/ReadMeFirst
RO/ReadMeFirst.info
RO.info
```

MUI is not included in the RO archive, but you should get it in every mailbox or at your local aminet-mirror.

1.3 System Requirements

System Requirements

RO requires version 2.04 of the Amiga operating system or newer. RO also requires version 2.3 or newer of the "Magic User Interface" by Stefan Stuntz. Kickstart 1.3 is not supported; Neither by the "Magic User Interface" nor by RO; this operating system is obsolete. Upgrade to 2.04 or, even better, to 3.1 of the Amiga OS. RO does not require a special processor, but of course the faster CPUs make life easier. Many complex calculations are needed for the management and layout of the objects, so a "base" 68000 based machine could be slow.

If you want RO to automatically decrunch XPK & PowerPacker packed files, you have to install `xpkmaster.library`, `powerpacker.library` and at least one XPK compressor library. If you want RO to treat archives just like directories you also need `ARCHandler`.

XPK is Copyright © Urban Mueller, Bryan Ford & Christian Schneider.
PowerPacker is Copyright © Nico François.
ARCHandler is Copyright © Rafael D'Halleweyn.

1.4 Unregistered Version

Unregistered Version

I decided to release RO as crippled shareware. RO is fully functional but in the unregistered version the about requester appears regularly to remind you to pay the shareware fee. RO has been and will be a time-consuming work for me. If you like this product and you want me to continue its development you should support me with the registration fee. Thank You.

How to register for RO

1.5 Installation

Installation

RO comes with an installer script which helps you to install all required files. If you are an 'expert user' you might want to do it manually.

Just take the RO icon and drop it into the directory where you want to install it. You might want to copy the file 'RO.prefs' into the same directory or to 'S:'. If you have a keyfile, copy it to 'S:', 'L:' or in the same directory where you copied the executable. You may also set the environment variable 'KEYPATH' to a path where you store your Keyfile. If you use Workbench 2.1 or better then you may copy the file ro.catalog to your 'LOCALE:Catalogs/<language>' directory.

1.6 Main Window

Main Window

After the installation procedure you may start RO, and the main window should open. It contains the following control objects...

Register	Changes the listview type
Information Textfield	Displays information about loaded dir
Listview	Displays the directory
Path Gadget	For manual directory browsing
Drive Buttons	Fast access of often used devices
Command Buttons	Performing actions on selected entries
Status Bar	Displays general information

1.7 Register Group

Register

The register at the top of the window lets you change the type of the listview below. You may choose between Directory, Buffers and Volumes. According to your MUI settings the register group might look like a

register or like a normal cycle gadget. The three different listview types are described in the

Listview
section.

If you own a mouse with three buttons it's also possible to change the listview type by clicking the middle mouse button above a listview.

1.8 Information Textfield

Information Textfield

There is a textfield above each directory listview where several informations are displayed: the free disk space in percent and KBytes, and the used disk space in KBytes. It looks similar to the information displayed in the title bar of a standard Workbench window so you should be familiar with it. If there is currently no directory loaded you will see 'No Directory'. The information text changes if you select some entries. You can see how many files, dirs and bytes currently selected, if you unselect all the text changes back to the disk space display.

The Textfield is also used to indicate which listview is currently active. It is very important to know which one of the listviews is active, because it is considered as the source listview and it is the one which is affected by all operations you start via command button or via user menu. If you select a file in a listview the active side changes but you may also change the active side via '' (The key above 'tab') which cycles through the listviews or via 'tab' which cycles through the string gadgets. The textfield above the activated listview is filled with a 'Fillback' raster which could be configured in the 'Pens' section in MUI prefs by changing the according pens.

Last but not least the Textfield is also used as a hidden 'Parent' button. If you click on it the parent directory of the current directory is loaded into the according listview. If there is no parent directory RO will just reload the current loaded directory

1.9 Listview

Listview

RO supplies three different listviews...

The 'Directory' listview is RO's most important listview. It is used to display the directory. You may select files and directories there and doubleclick on them. If you doubleclick on a directory it is loaded immediately into the listview, but if you doubleclick on a file RO tries to determine what kind of file it is and will start the according command. You may define filetypes in RO.prefs and in addition if you use OS 3.0 or better RO supports Datatypes. For a detailed description how to configure these fileactions consult the

Configuration

chapter.

The 'Buffers' listview is a kind of history where RO remembers a specific number of paths. You may also insert paths upon startup as described in the

Configuration

section. You may get to any of the displayed paths just by doubleclicking them. This is one way of fast directory access supported by RO.

The 'Volumes' listview displays a list of all available AmigaDOS volumes currently mounted. You can load a volume into the directory listview by doubleclicking on it.

1.10 Path Gadget

Path Gadget

The path gadgets are below the two directory listviews and contain the path string of the current loaded directory. You may enter any path here and the according directory will be loaded immediately. RO extends all path names e.g. if you enter 'Libs:' you get something like 'System:Libs/'. The path gadgets are just another fast way to access directories because of the bulid-in path name completer. If you enter 'SYS:Devs/DOS<RETURN>' you get an extended path which should look like 'System:Devs/DOSDrivers'. You may also enter more complex things like 'System:Devs/DataTypes///L/File<RETURN>' and what you get should look like that 'System:L/FileSystem_Trans/'. The path name completion works only for the last part of the path string.

1.11 Drive Buttons

Drive Buttons

Below the path gadgets RO normally has 16 drive buttons but their number is automatically increased to 32 if you specify more buttons. In general you click on a drive button and the specified path name for the button is loaded into a listview. Into which one depends on the current operation mode of the drive buttons. There are two modes supported: The so called 'Active Mode' and the 'Normal Mode'. In 'Normal Mode' the drive button only affects the listview which is directly above it. In 'Active Mode' the drive button loads a path into the activated listview.

1.12 Command Buttons

Command Buttons

Below the drive buttons are 3 x 8 command buttons. These gadgets are used to perform actions for the selected files in the active directory

listview. 24 buttons are not enough for over 50 internal functions so I implemented a button bank switch mechanism. You can switch between the banks by clicking the 'Status Bar' below the command buttons. If you own a three-button-mouse you may also flip the button bank by pressing the middle mouse button above the bank.

Currently RO supports 5 button banks which results in 120 buttons but this might change in a future version. Most internal commands open a progress window. If you close it the operation is aborted but if you hit its iconify gadget RO continues the function in background.

Currently supported are the following internal functions...

About	Show about information.
Action	Perform fileaction on selected files.
All	Select all entries in active listview.
Assign	Create an assign for active directory.
ByName	Select files by name.
Bytes	Count bytes of selected entries.
Change	Change sorttype of active listview.
Copy	Copy files and directories.
CopyAs	Copy files and directories and change name.
CopyDev	Copy files to a device.
Current	Load current directory.
Delete	Delete files and directories.
Disk	Show information about active disk.
Dup	Duplicate entries.

Expand	Expand a path to its ARCHHandler equivalent.
Fit	Check if entries fit on destination.
Help	Show online help.
Icon	Select corresponding .info files.
Info	Show information for selected files.
LActive	Activate left listview.
LCopy	Copy active listview to inactive.
LSwap	Swap active and inactive listview
ListArc	List contents of archives.
LParent	Load parent directory of left path.
LReload	Refresh left directory.
LRoot	Load root directory of left path.
MakeDir	Create directories.
MDLoad	Create a directory and load it.
Move	Move files and directories.
MoveAs	Move files and directories and change name.
None	Deselect all entries.
Note	Modify filenotes.
Parent	Load parent directory.
Protect	Modify protection flags.

Quit	Quit 'RO'
RActive	Activate right listview.
Reboot	Reboot your system.
Relabel	Relabel volumes.
Reload	Refresh active directory.
Rename	Rename entries.
Root	Load root directory.
RParent	Load parent directory of right path.
RReload	Refresh left directory.
RRoot	Load root directory of right path.
SetDate	Modify date of a file or directory.
Shrink	Shrink an ARCHandler path to normal.
Toggle	Toggleselect all entries.
Touch	Set date or create file.
UnArc	Extract files from archives.
Update	Select files for update.

...and you may specify any AmigaDOS executable here but consult the

Configuration
chapter for further information.

1.13 Status Bar

Status Bar

RO's status bar is the textfield at the bottom of the main window, it is used to display all general informations about what is currently going on or why a certain operation failed. As described in the Command Buttons section the status bar is also needed to switch between the different command button banks. You can also show a clock and/or a memory display there but this is total under user control so I advice you to check out the

Configuration
chapter.

1.14 Menus

Menus

RO has the following menus:

- o Project

- Update All

- Reloads left and right directory

- Last Message

- Redisplays the last message in the status bar.

- About...

- Opens the about requester. This is the same requester which pops up regularly if you are an unregistered user.

- Spawn New

- You can start new RO sessions here to do more things at the same time.

- Iconify

- Iconifies RO and adds an app icon to your Workbench. You can use it as a multifunctional drop icon because every file dropped on it will invoke a action according to its filetype. If you start RO you may force an iconified startup by using the command line argument or the tooltype 'ICONIFY'.

- Quit

- Quits the program and removes RO from memory.

- o Options

- Copy

o Flags

Toggles whether you want to clone the protection bits while copying a file or not. It has the same effect as the `COPYFLAGS` option in the preference file.

o Date

Toggles whether you want to clone the date & time of a file while copying or not. It has the same effect as the `COPYDATE` option in the preference file.

o Comment

Toggles whether you want to clone the file note while copying a file or not. It has the same effect as the `COPYNOTE` option in the preference file.

o Check Files

Using this option you may specify if RO should verify files after copy. You can also set this option in the preference file with the `COPYVERIFY` keyword.

o Check Diskspace

Switch this option on if you want to automatically check if the selected files will fit on the destination device before copying them. Due to the fact that 'RAM:' is always 100% full it is excluded from checking. You can also set this option in the preference file with the `CHECKFIT` keyword.

- Delete

o Verify

Toggles the `DELVERIFY` option. This means that if you turn it on a warning requester pops up before deleting files.

o Force

If you switch this option on RO unprotects protected files before deleting them. This has the same effect as the

DELFORCE
keyword
in RO.prefs file.

- Overwrite

Here you may choose one of the four values 'Always', 'Never', 'Older' or 'Verify'. They have the same effect as the 'ALWAYS', 'NEVER', 'OLDER' and 'VERIFY' value for the
OVERWRITE
keyword. As you might
have guessed these entries are mutual exclusive.

- Drop icons

With this menu entry you can change the behaviour of RO if you drop icons on one of the directory listviews. You may set it to 'Load' or 'Copy'. If you use 'Load' the directory of the dropped icon is loaded in the listview and if you select Copy the file is directly copied into the current loaded directory, so be careful what you choose here. This option can also be set via the
APPWINDOW
keyword in the
configuration file.

1.15 Configuration

Configuration

RO is configured via a text file which should be copied to 'S:'. It may also be located in the same directory as the executable but it must be called 'RO.prefs'. If you used the install script all files are installed correct. At the current state of development you have to use a standard text editor to modify the config file but I included some examples. Have a look in the 'Config' directory. RO supports the following configuration keywords...

FILETYPE

CUSTOMTYPE

ARCHIVETYPE

VARIABLE

MENU

BUFFER

HOTKEY

HOTDIR

DRIVE

BUTTON

Note: Window size and other things like layout or keyboard settings are configured with the MUI prefs program and I really advice you to read MUI's documentation. Thanks. ←

1.16 Filetype Configuration

FILETYPE

SYNOPSIS

```
FILETYPE "TypeID" "Command"
```

DESCRIPTION

With this keyword you can set commands for the three basic filetypes. TypeID must be one of the Ids listed below and command is a CLI command which has to contain a '%f' where the file name should be inserted. (The file name is already in quotes so don't specify them again!)

ASCII	ASCII standard text file
DEFAULT	Files not regcognized by CUSTOMTYPE's
BINARY	A not recognized binary file

ARGUMENTS

"TypeID"	file type identifier.
"Command"	CLI command to be executed.

You may specify these '%' and '&' parameters for 'Command':

%a	arexx portname
%f	source filename with path in quotes
%F	source filename with path without quotes
%n	only the source filename in quotes
%N	only the source filename without quotes
%p	source path in quotes
%P	source path without quotes
%t	target path in quotes
%T	target path without quotes
%s	source device in quotes
%S	source device without quotes
%d	destination device in quotes
%D	destination device without quotes
%r	request a string from user for every file
%R	request a string from user only once
&c	change current directory to source path
&C	change current directory to target path
&r	reloads the source directory
&R	reloads the destination directory

```
&w          output window must be closed by user

%%          percent sign
&&          and sign
```

DEFAULT

Command is set to "MultiView >NIL: %f" for all supported file types.

EXAMPLE

```
FILETYPE BINARY "FileX >NIL: %f"
```

1.17 Custom Filetype Configuration

CUSTOMTYPE

SYNOPSIS

```
CUSTOMTYPE "Pattern" "Hex" "Desc" "Command"
```

DESCRIPTION

With this keyword you can set commands which should be executed according to the file type of the selected file. Since version 1.00 of RO it is possible to add your own file types. If the recognition fails the

DEFAULT

command will be executed and if it fails as well the file is considered as binary file and the

BINARY

command ist used to display it. XPK & PP files are decrunched before checking the file type. 'Command' has to contain a '%f' where the file name should be inserted. (The file name is already in quotes so don't specify them again!) 'Pattern' is an AmigaDOS filename pattern which is used to find the matching file. 'Hex' is a string of ASCII characters also used to determine the filetype. You may specify a normal string like '??-lh?' or a hex string like "\$????2D6C68??2D". So if the first character of the string is a '\$' sign RO translates it into a normal string. This might get useful if you need non-printable chars. As you might have guessed '?' is a wildcard but don't forget to specify '??' if you are using a "\$"-Hex string.

ARGUMENTS

```
"Pattern"  AmigaDOS filename pattern to match
"Hex"      Hex string to match
"Desc"     Filetype description string
"Command"  CLI command to be executed.
```

You may specify these '%' and '&' parameters for 'Command':

```
%a          arexx portname
```

```

%f          source filename with path in quotes
%F          source filename with path without quotes
%n          only the source filename in quotes
%N          only the source filename without quotes
%p          source path in quotes
%P          source path without quotes
%t          target path in quotes
%T          target path without quotes
%s          source device in quotes
%S          source device without quotes
%d          destination device in quotes
%D          destination device without quotes
%r          request a string from user for every file
%R          request a string from user only once

&c          change current directory to source path
&C          change current directory to target path
&r          reloads the source directory
&R          reloads the destination directory
&w          output window must be closed by user

%%          percent sign
&&          and sign

```

DEFAULT

```

CUSTOMTYPE "#?" "FORM????8SVX" "Iff 8SVX File" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "FORM????ANIM" "Iff ANIM File" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "FORM????ILBM" "Iff ILBM File" "MultiView >NIL: %f"

CUSTOMTYPE "#?" "GIF8?a" "GIF Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "DMS!" "DMS Archive" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "??????JFIF" "JPeg Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "$0A050108" "PCX Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "$424D" "BMP Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "$59A66A95" "SunRaster Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "$000002" "Tga Picture" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "$00000A" "Targa Picture" "MultiView >NIL: %f"

CUSTOMTYPE "#?" "II*?" "Intel Tiff" "MultiView >NIL: %f"
CUSTOMTYPE "#?" "MM?*?" "Motorola Tiff" "MultiView >NIL: %f"

CUSTOMTYPE "#?.dvi" "" "DVI File" "MultiView >NIL: %f"
CUSTOMTYPE "#?.info" "" "Icon" "MultiView >NIL: %f"
CUSTOMTYPE "#?.mpg" "" "MPeg Movie" "MultiView >NIL: %f"
CUSTOMTYPE "#?.fli" "" "FLI Anim" "MultiView >NIL: %f"
CUSTOMTYPE "#?.guide" "" "AmigaGuide File" "MultiView >NIL: %f"
CUSTOMTYPE "#?.html" "" "HTML File" "MultiView >NIL: %f"
CUSTOMTYPE "#?.ps" "" "PostScript File" "MultiView >NIL: %f"

```

EXAMPLE

```

CUSTOMTYPE "#?" "@database" "AmigaGuide File" "MultiView >NIL: %f"

```

1.18 Archive Configuration

ARCHIVETYPE

SYNOPSIS

```
ARCHIVETYPE "Pattern" "Hex" "Desc" "ListArc" "UnArc"
```

DESCRIPTION

With this keyword you can set commands which should be executed according to the archive type of the selected files. There are two commands for each archive type: a listarc command and an unarc command. 'Pattern' and 'Hex' are needed to recognize archives. 'Pattern' is an AmigaDOS filename pattern which is used to find the matching file. 'Hex' is a string of ASCII characters also used to determine the filetype. You may specify a normal string like '??-lh?-' or a hex string like "\$????2D6C68??2D". So if the first character of the string is a '\$' sign RO translates it into a normal string. This might get useful if you need non-printable chars. As you might have guessed '?' is a wildcard but don't forget to specify '??' if you are using a "\$"-Hex string. 'ListArc' and 'UnArc' should contain a '%f' for the filename. Since version 1.00 of RO adding files to archives is handled by normal command buttons for example using "LhA >NIL: -x -r a %R %n &c" as command string.

ARGUMENTS

"Pattern"	AmigaDOS filename pattern to match
"Hex"	Hex string to match
"Desc"	Filetype description string
"ListArc"	Command for listing archive
"UnArc"	Command for extracting archive

You may specify these '%' and '&' parameters for 'Command':

%a	arexx portname
%f	source filename with path in quotes
%F	source filename with path without quotes
%n	only the source filename in quotes
%N	only the source filename without quotes
%p	source path in quotes
%P	source path without quotes
%t	target path in quotes
%T	target path without quotes
%s	source device in quotes
%S	source device without quotes
%d	destination device in quotes
%D	destination device without quotes
%r	request a string from user for every file
%R	request a string from user only once
&c	change current directory to source path
&C	change current directory to target path
&r	reloads the source directory
&R	reloads the destination directory

```
&w          output window must be closed by user

%%          percent sign
&&          and sign
```

DEFAULT

```
ARCHIVETYPE "#?.lha" "" "LHA/LZH Archive" "LhA v %f" "LhA x %f &C &R"
ARCHIVETYPE "#?.lzx" "" "LZX Archive"      "LzX v %f" "LzX x %f &C &R"
ARCHIVETYPE "#?.arc" "" "ARC Archive"      "Arc v %f" "Arc x %f &C &R"
ARCHIVETYPE "#?.arj" "" "ARJ Archive"      "Arj v %f" "Arj x %f &C &R"
ARCHIVETYPE "#?.zip" "" "ZIP Archive"      "Zip v %f" "Zip x %f &C &R"
ARCHIVETYPE "#?.zoo" "" "ZOO Archive"      "Zoo v %f" "Zoo x %f &C &R"
```

EXAMPLE

```
ARCHIVETYPE "#?" "$4C5A5800" "LZX Archive" "LzX v %f" "LzX x %f &C &R"
```

1.19 General Configuration

VARIABLE

SYNOPSIS

```
VARIABLE "VarID" "Value"
```

DESCRIPTION

This keyword is needed to set some important options. 'VarID' is one of the variable names listed below and value is a parameter string for the variable. Currently the following 'VarID's are supported:

Global:

- APPWINDOW

Here you can specify what happens when you drop an icon on a listview. If you set it to 'LOAD' the directory of the dropped icon will be loaded into the listview, if you use 'COPY' the dropped file or directory will be copied into it the current loaded directory.

Example: VARIABLE APPWINDOW "LOAD"

- COMPLETED

Here you can specify what should happen if RO finished an operation while it is iconified. You may set it to 'NOTHING', 'FLASH' or 'POPUP'.

Example: VARIABLE COMPLETED "NOTHING"

- CHECKFIT

Here you can switch ON or OFF whether you like to check the filesize if it fits on the destination disk or not. RO counts the bytes and

checks how many blocks are needed for organisation etc. on the destination device. So don't get confused by values which seem to be wrong.

Example: VARIABLE CHECKFIT "ON"

- CREATEICONS

If you want RO to create a directory icon if you use mkdir or to create a project icon if you create a file set this option ON, if not switch it OFF.

Example: VARIABLE CREATEICONS "OFF"

- DRIVEBUTTONS

It is a bit difficult to explain... there are two different types of Drivebutton handling. The first one is called 'Normal' and the second one is called 'Active'. In normal mode you have eight Drivebuttons for every side, so if you click on a Drivebutton below the left side the directory is loaded in the left side. In active mode you have sixteen Drivebuttons for both sides. If you click on a Drivebutton it affects the active Listview and not the one according to the side where you clicked.

Example: VARIABLE DRIVEBUTTONS "Active"

- FLUSHMEMORY

If you want RO to flush memory on exit, switch this option ON. This will remove MUI from memory if no other application using it is currently running.

Example: VARIABLE FLUSHMEMORY "ON"

- FOLLOWSCROLL

This keyword is needed to switch ON or OFF whether you like to scroll the directory window to follow operations or not.

Example: VARIABLE FOLLOWSCROLL "OFF"

- HISTORY

Here you can set the number of buffers RO should remember.

Example: VARIABLE HISTORY "42"

- HOTDIRS

Exactly the same as DRIVEBUTTONS but for HotDirs. Function keys F1 to F5 affect the left side, F6 to F10 affect the right side.

Example: VARIABLE HOTDIRS "Active"

- OUTPUT

Specify the default output window for all commands launched by RO.

Example: VARIABLE OUTPUT "CON:0/0/640/400/Output/CLOSE/AUTO"

- OVERWRITE

You can specify what should happen if you write to a file which already exists perhaps if you want to copy another one over it. You may choose between 'ALWAYS', 'NEVER', 'OLDER' and 'VERIFY'. ALWAYS will quietly overwrite the file without further information, NEVER will break the operation if a file already exists, OLDER will only replace older files and VERIFY will pop up a 'Do you really want...' requester.

Example: VARIABLE OVERWRITE "ALWAYS"

- PATHEXPAND

This switch controls the path extension. If you switch it ON all paths are 'expanded'. E.g. if you enter LIBS: it is expanded to something like "System:Libs/". Some older filesystems like ParNet, SerNet or MessyDOS may cause problems with it, so if you use one of them turn this switch OFF.

Example: VARIABLE PATHEXPAND "ON"

- QUITVERIFY

Switches the 'Do you really want to quit RO?' requester ON or OFF, where you can select Quit, Iconify or Cancel. Hitting the Close Gadget equals clicking the Cancel Gadget.

Example: VARIABLE QUITVERIFY "ON"

- TEMPDIR

Here you can specify a temporary directory which is needed for some special actions like decrunching files or listing archives.

Example: VARIABLE TEMPDIR "TEMP:"

Copy:

- COPYBUF

Here you can set the maximum buffer RO will use for copying files.

Example: VARIABLE COPYBUF "50000"

- COPYDATE

Here you can switch ON or OFF whether you like to copy the date of a file or not. Otherwise the date will be set to the actual date.

Example: VARIABLE COPYDATE "OFF"

- COPYFLAGS

Here you can switch ON or OFF whether you like to copy the protection bits or not. If not, RO sets the default protection (RWED).

Example: VARIABLE COPYFLAGS "OFF"

- COPYNOTE

Here you can switch ON or OFF whether you like to copy the note of a file or not. If not, RO clears the note field.

Example: VARIABLE COPYNOTE "OFF"

- COPYSTAT

Here you can switch ON or OFF whether you like to see the currently copied file or not. This could be useful if you copy many small files because displaying every copied file in the status bar could mean a heavy cpu load.

Example: VARIABLE COPYSTAT "OFF"

- COPYVERIFY

Here you can switch ON or OFF whether you like to verify files after copy or not. The copy procedure might take a bit longer if you switch it ON, but you know that the files are error free.

Example: VARIABLE COPYVERIFY "OFF"

Delete:

- DELDELAY

Here you can specify a delay in ticks (50 ticks = 1 second) to wait before deleting a temporary generated file (e.g. while listing an archive or decrunching a XPK packed file). This could be useful if you have a text viewer which detaches from CLI and is not fast enough to load the temp file before it is deleted. Warning: There might be problems deleting the file if the command locks the temp file. So its better to use a command that does not detach from CLI.

Example: VARIABLE DELDELAY "50"

- DELFORCE

Here you can switch ON or OFF whether you want to force deleting of protected files. This setting also affects overwriting protected files. If you switch it ON the files are quietly deleted/overwritten.

Example: VARIABLE DELFORCE "ON"

- DELSTAT

Here you can switch ON or OFF whether you like to see the currently deleted file in the status bar or not. This might be useful for the

same reasons as COPYSTAT.

Example: VARIABLE DELSTAT "OFF"

- DELVERIFY

Switches the 'Are you sure you want to delete these entries?' requester ON or OFF.

Example: VARIABLE DELVERIFY "OFF"

Listview:

- FORMATLEFT/FORMATRIGHT

With this keyword you can specify which information is displayed in the listview entries. You can choose a combination of the letters N (Name), S (Size), F (Flags), D (Date), T (Time) and C (Comment).

Example: VARIABLE FORMATLEFT "NTSC"

- LOADLEFT/LOADRIGHT

You can specify a path which is automatically loaded in the according Listview on startup.

Example: VARIABLE LOADRIGHT "RAM:"

- PAGELEFT/PAGERIGHT

This variable sets the page which is in front if you start RO. There are three pages: 'Directory', 'Buffers' and 'Volumes'.

Example: VARIABLE PAGERIGHT "Volumes"

- FIRSTLEFT/FIRSTRIGHT

Adjust which type of directory entries is displayed first. Possible settings are "Dirs", "Files" and "Mixed".

Example: VARIABLE FIRSTLEFT "Mixed"

- SORTLEFT/SORTRIGHT

Indicate which field should be used as sort criteria. Possible settings are "Name", "Date" and "Size".

Example: VARIABLE SORTRIGHT "Size"

- SYMMETRICAL

This switch controls the look of the listviews. If you switch it OFF the listviews of the right side will look as the listviews of the left side. If you use ON the position of right listview's scroll bars is set to left in order to provide a symmetrical look.

Example: VARIABLE SYMMETRICAL "ON"

- CHECKDUPES

Here you can specify if you want to check for dupes on the 'Buffers' Page or not. If you set this Option ON you will see no dupes otherwise dupes are displayed.

Example: VARIABLE CHECKDUPES "OFF"

Clock:

- CLOCKPRI

You can specify the priority of the 'RO Clock' Task here. I suggest to use the default value. The value must be between -128 and 127

Example: VARIABLE CLOCKPRI "0"

- CLOCKREFRESH

Here you should specify the ticks (50 ticks = 1 second) between two clock updates. This value should be chosen carefully, because if you specify a very small value the clock starts flashing.

Example: VARIABLE CLOCKREFRESH "50"

- SHOWDATE

Here You may specify if you want to display a date string in the status bar not.

Example: VARIABLE SHOWDATE "OFF"

- SHOWDAY

Here you may specify if a day string like 'Sunday' or 'Monday' should be displayed.

Example: VARIABLE SHOWDAY "ON"

- SHOWTIME

Switch this variable 'ON' if you want RO's Clock to display the time.

Example: VARIABLE SHOWTIME "OFF"

- SHOWSECONDS

Specifies if RO's Clock should display seconds. This variable takes only affect if SHOWTIME is set to 'ON'. I suggest to turn this option off to avoid heavy cpu loads.

Example: VARIABLE SHOWSECONDS "ON"

- SHOWMEMORY

Here you can switch the Workbench-Style Memory display 'ON' and 'OFF'

Example: VARIABLE SHOWMEMORY "ON"

DEFAULT

VARIABLE	APPWINDOW	"LOAD"
VARIABLE	FLUSHMEMORY	"OFF"
VARIABLE	CREATEICONS	"ON"
VARIABLE	FOLLOWSCROLL	"ON"
VARIABLE	PATHEXPAND	"ON"
VARIABLE	QUITVERIFY	"ON"
VARIABLE	COMPLETED	"Flash"
VARIABLE	DRIVEBUTTONS	"Normal"
VARIABLE	HOTDIRS	"Normal"
VARIABLE	TEMPDIR	"T:"
VARIABLE	HISTORY	"50"
VARIABLE	OUTPUT	"CON:0/0/640/200/Output/CLOSE/AUTO"
VARIABLE	COPYBUF	"1048576"
VARIABLE	COPYDATE	"ON"
VARIABLE	COPYFLAGS	"ON"
VARIABLE	COPYNOTE	"ON"
VARIABLE	COPYSTAT	"ON"
VARIABLE	COPYVERIFY	"ON"
VARIABLE	CHECKFIT	"ON"
VARIABLE	OVERWRITE	"VERIFY"
VARIABLE	DELDELAY	"0"
VARIABLE	DELFORCE	"OFF"
VARIABLE	DELVERIFY	"ON"
VARIABLE	DELSTAT	"ON"
VARIABLE	FORMATLEFT	"NSFDTC"
VARIABLE	FORMATRIGHT	"NSFDTC"
VARIABLE	LOADLEFT	" "
VARIABLE	LOADRIGHT	" "
VARIABLE	PAGELEFT	"Directory"
VARIABLE	PAGERIGHT	"Directory"
VARIABLE	FIRSTLEFT	"Dirs"
VARIABLE	FIRSTRIGHT	"Dirs"
VARIABLE	SORTLEFT	"Name"
VARIABLE	SORTRIGHT	"Name"
VARIABLE	SYMMETRICAL	"OFF"
VARIABLE	CHECKDUPES	"ON"
VARIABLE	CLOCKREFRESH	"500"
VARIABLE	CLOCKPRI	"-10"
VARIABLE	SHOWDAY	"ON"
VARIABLE	SHOWDATE	"ON"
VARIABLE	SHOWTIME	"ON"
VARIABLE	SHOWSECONDS	"OFF"

VARIABLE SHOWMEMORY "ON"

1.20 Menu Configuration

MENU

SYNOPSIS

MENU "MenuText" "Command"

DESCRIPTION

With this keyword you can create a user menu with up to 20 entries, the command may contain a '%f' if you want to use your menu entry to perform an action on the selected files. If you just want to start a general utility like a disk copier or a calculator do not specify a '%f' within the command string otherwise a selected entry is needed to start the command.

ARGUMENTS

"MenuText" 40 characters as menu text
 "Command" CLI command to execute

You may specify these '%' and '&' parameters for 'Command':

%a	arexx portname
%f	source filename with path in quotes
%F	source filename with path without quotes
%n	only the source filename in quotes
%N	only the source filename without quotes
%p	source path in quotes
%P	source path without quotes
%t	target path in quotes
%T	target path without quotes
%s	source device in quotes
%S	source device without quotes
%d	destination device in quotes
%D	destination device without quotes
%r	request a string from user for every file
%R	request a string from user only once
&c	change current directory to source path
&C	change current directory to target path
&r	reloads the source directory
&R	reloads the destination directory
&w	output window must be closed by user
%%	percent sign
&&	and sign

DEFAULT

MENU "" ""

EXAMPLE

```
MENU "Calculator" "SYS:Tools/Calculator"
```

1.21 Buffer Configuration

BUFFER

SYNOPSIS

```
BUFFER "PathName"
```

DESCRIPTION

With this keyword it is possible to fill the 'Buffer' page with directories you often need. The number of buffers is only limited by the amount of memory you have.

ARGUMENTS

```
"PathName"    directory name to insert
```

DEFAULT

```
BUFFER ""
```

EXAMPLE

```
BUFFER "DH1:Comm/Term/Uploads"
```

1.22 HotKey Konfiguration

HOTKEY

SYNOPSIS

```
HOTKEY "RawKey" "Commandtype" "Command"
```

DESCRIPTION

Here you may specify up to 50 hotkeys. With these hotkeys it is possible to execute every internal function or cli command. You should care that you use unique rawkeys for all Commodities and the hotkeys only work if the main window of RO is activated. All internal functions are described in the

BUTTON
configuration chapter.

ARGUMENTS

```
"RawKey"      standard AmigaDOS rawkey string  
"CommandType" COMMAND or FUNCTION for a cli command or a function
```


"Command" CLI command to execute

You may specify these '%' and '&' parameters for 'Command':

```
%a      arexx portname
%f      source filename with path in quotes
%F      source filename with path without quotes
%n      only the source filename in quotes
%N      only the source filename without quotes
%p      source path in quotes
%P      source path without quotes
%t      target path in quotes
%T      target path without quotes
%s      source device in quotes
%S      source device without quotes
%d      destination device in quotes
%D      destination device without quotes
%r      request a string from user for every file
%R      request a string from user only once

&c      change current directory to source path
&C      change current directory to target path
&r      reloads the source directory
&R      reloads the destination directory
&w      output window must be closed by user

%%      percent sign
&&      and sign
```

DEFAULT

```
HOTKEY "" FUNCTION ""
HOTKEY "" FUNCTION ""
HOTKEY "" FUNCTION ""
HOTKEY "" FUNCTION ""
HOTKEY "" FUNCTION ""

HOTKEY "" COMMAND ""
HOTKEY "" COMMAND ""
HOTKEY "" COMMAND ""
HOTKEY "" COMMAND ""
HOTKEY "" COMMAND ""
```

EXAMPLE

```
HOTKEY "ctrl alt del" FUNCTION "REBOOT"
HOTKEY "shift c"      FUNCTION "COPYAS"
HOTKEY "left"        FUNCTION "PARENT"
```

1.23 HotDir Configuration

HOTDIR

SYNOPSIS

```
HOTDIR "Number" "PathName"
```

DESCRIPTION

Here you can specify up to ten Hotdirs. These Hotdirs can be accessed via the function keys F1-F10. 'Number' must be between 1 and 10.

ARGUMENTS

```
"Number"    function key number between 1 and 10
"PathName"  directory name to load
```

DEFAULT

```
HOTDIR 1 ""
HOTDIR 2 ""
HOTDIR 3 ""
HOTDIR 4 ""
HOTDIR 5 ""
HOTDIR 6 ""
HOTDIR 7 ""
HOTDIR 8 ""
HOTDIR 9 ""
HOTDIR 10 ""
```

EXAMPLE

```
HOTDIR 5 "DH1:Comm/Term/Uploads"
```

1.24 Drive Configuration

DRIVE

SYNOPSIS

```
DRIVE "Number" "GadgetText" "PathName"
```

DESCRIPTION

Here you can specify up to 32 drive buttons, if you specify 16 buttons you will get only one row of gadgets if you specify more, a second row will be added. The GadgetText is limited to 10 characters this should be enough and more would need a really high resolution. It might happen that RO does not fit on your screen even with 5 characters per button. It would be best if you play around with it but I suggest to use 3 characters.

ARGUMENTS

```
"Number"    number of drive button between 1 and 32
"GadgetText" 10 characters as gadget text
"PathName"  directory name to load
```

DEFAULT

```
DRIVE 1 "DF0" "DF0:"
DRIVE 2 "DF1" "DF1:"
DRIVE 3 "DH0" "DH0:"
DRIVE 4 "DH1" "DH1:"
DRIVE 5 "DH2" "DH2:"
DRIVE 6 "DH3" "DH3:"
DRIVE 7 "CD0" "CD0:"
DRIVE 8 "RAM" "RAM:"
DRIVE 9 "DF0" "DF0:"
DRIVE 10 "DF1" "DF1:"
DRIVE 11 "DH0" "DH0:"
DRIVE 12 "DH1" "DH1:"
DRIVE 13 "DH2" "DH2:"
DRIVE 14 "DH3" "DH3:"
DRIVE 15 "CD0" "CD0:"
DRIVE 16 "RAM" "RAM:"
```

EXAMPLE

```
DRIVE 2 "INBOUND" "Work:News/Inbound/"
```

1.25 Button Configuration

BUTTON

SYNOPSIS

```
BUTTON "Number" "ButtonType" "GadgetText" "Command"
```

DESCRIPTION

This is the most important keyword in the RO's configuration, here you can specify which text appears on which button and which command is executed if you click on a button. After the Keyword you have to supply a number of the button you want to set. It must be between 1 and 120. Currently RO supports two types of Buttons: A command button and a function button. A command button is a button where you can implement a standard CLI command. All you have to do is to supply a command string with a '%f' at the position where the file name should be inserted. RO supports several other % instructions which are described below. Of course RO also cares about the standard system path which means that you don't have to specify the complete path for a command if it is in the AmigaDOS search path. The function button is a button where you call internal functions. You must also supply a gadget string for the button which may not be longer than 10 characters. If you specify a gadget string like '_Copy' or 'R_rename' then you are able to access the gadget via keys, but beware of dupes like '_Delete' & '_Disk'. Currently supported are the following internal functions:

About

Show about information.

Action	Perform fileaction on selected files.
All	Select all entries in active listview.
Assign	Create an assign for active directory.
ByName	Select files by name.
Bytes	Count bytes of selected entries.
Change	Change sorttype of active listview.
Copy	Copy files and directories.
CopyAs	Copy files and directories and change name.
CopyDev	Copy files to a device.
Current	Load current directory.
Delete	Delete files and directories.
Disk	Show information about active disk.
Dup	Duplicate entries.
Expand	Expand a path to its ARCHandler equivalent.
Fit	Check if entries fit on destination.
Help	Show online help.
Icon	Select corresponding .info files.
Info	Show quick information for all devices.
LActive	Activate left listview.

LCopy	Copy active listview to inactive.
LSwap	Swap active and inactive listview
ListArc	List contents of archives.
LParent	Load parent directory of left path.
LReload	Refresh left directory.
LRoot	Load root directory of left path.
MakeDir	Create directories.
MDLoad	Create a directory and load it.
Move	Move files and directories.
MoveAs	Move files and directories and change name.
None	Deselect all entries.
Note	Modify filenotes.
Parent	Load parent directory.
Protect	Modify protection flags.
Quit	Quit 'RO'
RActive	Activate right listview.
Reboot	Reboot your system.
Relabel	Relabel volumes.
Reload	Refresh active directory.
Rename	Rename entries.

Root	Load root directory.
RParent	Load parent directory of right path.
RReload	Refresh left directory.
RRoot	Load root directory of right path.
SetDate	Modify date of a file or directory.
Shrink	Shrink an ARCHandler path to normal.
Toggle	Toggleselect all entries.
Touch	Set date or create file.
UnArc	Extract files from archives.
Update	Select files for update.

ARGUMENTS

"Number" number of command button between 1 and 120
 "ButtonType" COMMAND or FUNCTION button
 "GadgetText" 15 chars as gadget text
 "Command" CLI command to execute

You may specify these '%' and '&' parameters for 'Command':

%a arexx portname
 %f source filename with path in quotes
 %F source filename with path without quotes
 %n only the source filename in quotes
 %N only the source filename without quotes
 %p source path in quotes
 %P source path without quotes
 %t target path in quotes
 %T target path without quotes
 %s source device in quotes
 %S source device without quotes
 %d destination device in quotes
 %D destination device without quotes
 %r request a string from user for every file
 %R request a string from user only once

&c change current directory to source path
 &C change current directory to target path
 &r reloads the source directory
 &R reloads the destination directory
 &w output window must be closed by user

%% percent sign
 && and sign

DEFAULT

BUTTON	1	COMMAND	"Read"	"Most >NIL: %f"
BUTTON	2	COMMAND	"Run"	"Run >NIL: %f"
BUTTON	3	FUNCTION	"_Delete"	DELETE
BUTTON	4	FUNCTION	"_Move"	MOVE
BUTTON	5	FUNCTION	"_Copy"	COPY
BUTTON	6	FUNCTION	"_Bytes"	BYTES
BUTTON	7	FUNCTION	"All"	ALL
BUTTON	8	COMMAND	"Arc"	"LhA >NIL: -x -r a %R %n &c"
BUTTON	9	COMMAND	"View"	"MultiView >NIL: %f"
BUTTON	10	COMMAND	"_Edit"	"Ed >NIL: %f"
BUTTON	11	FUNCTION	"Touch"	TOUCH
BUTTON	12	FUNCTION	"Move As"	MOVEAS
BUTTON	13	FUNCTION	"Copy As"	COPYAS
BUTTON	14	FUNCTION	"Fit"	FIT
BUTTON	15	FUNCTION	"None"	NONE
BUTTON	16	FUNCTION	"ListArc"	LISTARC
BUTTON	17	FUNCTION	"_Action"	ACTION
BUTTON	18	COMMAND	"XEdit"	"FileX >NIL: %f"
BUTTON	19	FUNCTION	"MakeDir"	MAKEDIR
BUTTON	20	FUNCTION	"_Rename"	RENAME
BUTTON	21	FUNCTION	"Dup"	DUP
BUTTON	22	FUNCTION	"Refresh"	RELOAD
BUTTON	23	FUNCTION	"_Pattern"	BYNAME
BUTTON	24	FUNCTION	"UnArc"	UNARC
BUTTON	25	COMMAND	"Music"	"DeliTracker >NIL: MODULE %f"
BUTTON	26	FUNCTION	"Assign"	ASSIGN
BUTTON	27	COMMAND	"Command"	"%r"
BUTTON	28	FUNCTION	"_Comment"	NOTE
BUTTON	29	FUNCTION	"Change"	CHANGE
BUTTON	30	FUNCTION	"Parent"	PARENT
BUTTON	31	FUNCTION	"Reverse"	TOGGLE
BUTTON	32	FUNCTION	"_Disk"	DISK
BUTTON	33	FUNCTION	"LSwap"	LSWAP
BUTTON	34	FUNCTION	"_Relabel"	RELABEL
BUTTON	35	COMMAND	"Execute"	"Execute %r"
BUTTON	36	FUNCTION	"_Protect"	PROTECT
BUTTON	37	FUNCTION	"_Expand"	EXPAND
BUTTON	38	FUNCTION	"Root"	ROOT
BUTTON	39	FUNCTION	"_Update"	UPDATE
BUTTON	40	FUNCTION	"_Info"	INFO
BUTTON	41	FUNCTION	"LCopy"	LCOPY

BUTTON	42	FUNCTION	"CopyDev"	COPYDEV
BUTTON	43	COMMAND	"Rexx"	"Rx %r"
BUTTON	44	FUNCTION	"SetDate"	SETDATE
BUTTON	45	FUNCTION	"_Shrink"	SHRINK
BUTTON	46	FUNCTION	"Current"	CURRENT
BUTTON	47	FUNCTION	"Icon"	ICON
BUTTON	48	FUNCTION	"Help"	HELP

EXAMPLE

```
BUTTON 98 COMMAND "DirTree" "DirTree %f"
```

1.26 about**About**

This function opens the requester which contains my address and other information about the registration.

1.27 action**Action**

This function performs a command on every selected file according to its file type. You have to specify a command for each file type if you want to use this function otherwise MultiView will be started for all file types. Consult the

FILETYPE

section in the configuration chapter for further information.

You can abort this function by closing the progress indication window.

1.28 all**All**

This function selects all entries in the active directory listview.

1.29 assign**Assign**

This function creates an assign for the path currently loaded in the active listview. A requester will pop up to ask you for a device name.

You should enter something like 'foobar:' here. The colon is optional and may be left out.

1.30 byname

ByName

This function selects all entries matching a specified pattern in the active directory listview. A requester pops up and asks you for a pattern. You may specify all standard AmigaDOS wildcards such as '#?' or '*' here.

1.31 bytes

Bytes

This function counts all bytes of the selected entries and their sub directories in the active listview. After that an information is displayed in the status bar which should look like '64728 Bytes in 109 Files and 15 Directories'.

You can abort this function by closing the progress indication window.

1.32 change

Change

With this function you can change the current sort type of the active listview. This works analog to the

SORTLEFT

,

SORTRIGHT

and

FIRSTLEFT

,

FIRSTRIGHT

keywords in the configuration file.

A requester opens where you can select the sort criterias.

1.33 copy

Copy

This function copies all selected files and/or directories from the source listview to the destination path. If you use CopyAs a requester pops up

which asks you for a new name of the currently copied file. You can abort the operation by hitting the 'Cancel' gadget or you may skip the current file by clicking on 'Skip'.

If you set the according options in the preference file or you selected them from menu, the flags, comment and date of the copied files are cloned. Refer to the

VARIABLE
configuration chapter.

You can abort this function by closing the progress indication window.

1.34 copydev

CopyDev

This function copies all selected files of the active listview to a specified device. A requester pops up which asks you where the entries should be copied to. This is useful if you have a couple of ascii text files and want to print them. All you have to do is 'CopyDev' them to 'PRT:'.

1.35 current

Current

This function loads the current directory of the active listview.

1.36 delete

Delete

This function deletes all entries currently selected in the active directory. A requester might pop up which asks you if you are really sure that you want to delete the selected entries but this depends on your configuration or on the

DELVERIFY
settings in the menu.

You can abort this function by closing the progress indication window.

1.37 disk

Disk

This function opens a requester which shows all information about the device currently loaded in the active listview. You can obtain things like free bytes or block size of the device here. To close the requester just click on okay or use the close gadget.

1.38 dup

Dup

This function duplicates all selected entries in the active listview. You will be asked at each entry how to call the copy. You may abort this function by hitting the 'Cancel' gadget or you may skip a single file by clicking on the 'Skip' gadget.

1.39 expand

Expand

If you use ARCHandler this might be useful for you. This function expands the path name of the actual loaded directory of the active listview. e.g. System:Devs/ to Archives:System/Devs/

1.40 fit

Fit

This function calculates the number of bytes the selected entries will need on the destination device. RO also cares about the blocks needed for directory organisation. After that an information is displayed in the status bar which tells you if the selected entries fit on the destination device or not.

You can abort this function by closing the progress indication window.

1.41 help

Help

This function opens the online help main page. You can also get this by pressing the HELP key. When the HELP key is pressed RO tries to obtain the help node from the current object (the one under the mouse pointer) and shows the according help page. Make sure that you set the environment 'ENV:AmigaGuide/Path' correct so RO can find the helpfile if it's not in the same directory as the executable.

1.42 icon

Icon

This function selects all corresponding '.info' files for the current selected entries in the active listview.

1.43 info

Info

This function opens a requester which shows all information about the currently selected entries of the active listview. You can obtain things like file type and protection bits here. To get to the next file close the requester just by clicking on okay. The function may be aborted by hitting the 'Cancel' gadget or by closing the window via close gadget.

1.44 active

Active

This function lets you activate a certain Listview. LActive activates the left one and RActive activates the right one.

1.45 lcopy

LCopy

This function copies the active listview path to the other listview.

1.46 lswap

LSwap

This function swaps the active and inactive listview paths.

1.47 listarc

ListArc

This function lists all selected archive files using the matching archive lister and the ascii file type command.

You can abort this function by closing the progress indication window.

1.48 parent

Parent

This function loads the parent directory of the active listview. If you use 'LParent' it performs a Parent function on the left listview if you use 'RParent' it uses the right listview. If RO can't find a parent directory for the path it just updates the directory.

1.49 reload

Reload

The functions Reload, LReload and RReload update the active, left or right directory currently loaded in the directory listview.

1.50 root

Root

This function loads the root directory of the active listview. If you use LRoot or RRoot this function applies to the left or right side.

1.51 makedir

MakeDir

If you use this function a requester appears which asks you for a name. The name you enter will be used as name for a directory which is created in the active listview. If you use 'MDLoad' instead the directory is also loaded into the source listview.

1.52 move

Move

This function moves all selected files and/or directories from the source listview to the destination path. If you use MoveAs a requester pops up which asks you for a new name of the currently moved file. You can abort the operation by hitting the 'Cancel' gadget or you may skip the current file by clicking on 'Skip'. RO notices if you move files from a device to the same device. It just renames the files which is probably much faster than copying and deleting them.

If you set the according options in the preference file or you selected them from menu the flags, comment and date of the moved files are cloned. Refer to the

VARIABLE
configuration chapter.

You can abort this function by closing the progress indication window.

1.53 none

None

This function deselects all entries in the active directory listview.

1.54 note

Note

This function lets you change the file comment of all selected entries in the active listview. A requester pops up which contains the old comment and you may change it there. The operation can be aborted by hitting the 'Cancel' gadget or you may skip the current file by clicking the 'Skip' gadget. To proceed just press return or use the 'Okay' button.

1.55 protect

Protect

This function opens a requester where you may change the protection bits for the selected entries. For each protection flag you may choose 'Ignore', 'Set' or 'UnSet'. 'Ignore' means that this bit is not changed, 'Set' means that this bit is set and 'UnSet' means that this bit is cleared. You may change these bits step by step for each entry by using the 'Okay' gadget or you may change them all at once by clicking the 'All' button. The operation may be aborted by hitting the 'Cancel' gadget or by closing the progress indication window.

1.56 quit

Quit

Quits the program and removes RO from memory.

1.57 reboot

Reboot

Reboots your computer without any warning requester. :)

1.58 relabel

Relabel

With this function you can change the device name of the directory loaded in the active listview. A requester pops up and you may modify the name there. If you want to abort the operation use the 'Cancel' button.

1.59 rename

Rename

This function is needed to rename the selected entries. A requester pops up for every entry and you may change its name. If you want to abort the operation click on 'Cancel' but if you just want to skip the current entry select 'Skip'. After you have changed the name press return to proceed or hit the 'Okay' Button.

1.60 setdate

SetDate

This function lets you set the date and time of all entries currently selected in the active listview. A requester pops up with the old date and time of the entry as string gadget contents and you may modify it. The input must be in standard AmigaDOS format. You may abort the function by using the 'Cancel' gadget or you may skip the current file by hitting the 'Skip' button.

1.61 shrink

Shrink

If you use ARCHandler this might be useful for you. This function shrinks the path name of the actual loaded directory of the active listview. e.g. Archives:System/Devs/ to System:Devs/ but it works only if the currently loaded directory is no archive file.

1.62 toggle

Toggle

This function toggle selects all entries in the active directory listview.

1.63 touch

Touch

If you have one or more entries selected this function updates the date and time, otherwise a requester asks you for a file name and this file will be created in the active path. If you are updating dates you may abort the operation by closing the progress indication window otherwise just press the 'Cancel' button.

1.64 unarc

UnArc

This function extracts all selected archives to the destination listview using an archive extractor according to the file type.

You can abort this function by closing the progress indication window.

1.65 update

Update

This function selects entries in the active listview which do already exist in the destination listview. This is quiet useful if you update a software package. If you want to select the entries which do not exist in the destination listview you should use 'Update' and 'Reverse'.

1.66 Arexx Interface

ARexx Interface

RO has a built-in ARexx port, which understands the following commands:

- QUIT
Ends the application.

 - HIDE
Hides (iconifies) the application

 - SHOW
Shows (pops up) an iconified application.
-

- INFO ITEM/A

According to the given parameter the result string is filled with the following contents:

- "title" Title of the application
- "author" Author of the application
- "copyright" Copyright message
- "description" Short description
- "version" Version string
- "base" Name of the ARexx port
- "screen" Name of the public screen

Example: 'Info "title"'

- HELP FILE/A

A list of all ARexx commands available for the application is written into the given file.

Example: 'Help "RAM:RexxCommands"'

- ACTIVE SIDE/A

With this function you can make a certain listview the active one. 'Active Left' activates the left one and 'Active Right' the right one. If you use 'Active Active' the active one gets activated which seems a bit useless but as the 'Active' command returns the last active as result, you can obtain the currently active listview without changing it. 'Active Inactive' activates the listview which is currently unselected.

Example: 'Active Inactive'

- CHOICEREQ STRING/A

'ChoiceReq String' opens a requester which may be answered clicking 'Okay' or 'Cancel'. If you use 'Okay' it returns '1' otherwise it returns '0'.

Example: 'ChoiceReq "Are you sure?"'

- COMMAND STRING/A,FILE/A

This function starts a command. All you have to do is to supply a

parse string with the supported % parameters. FILE/A is the name of the file which replaces the %f or %F in the command string. If an error occurred while executing the command RC is set to a value unequal zero.

Example: 'Command "Delete %f" "S:User-Startup"'

- FILEACTION FILE/A

With this function you could activate the file recognition of RO. It happens exactly the same as if you double clicked the file specified by FILE/A.

Example: 'FileAction "S:Startup-Sequence"'

- FUNCTION NAME/A

With this command you may call any internal function RO supports.

Example: 'Function "COPYAS"'

- GETENTRY SIDE/A,TYPE/A

'GetEntry' returns the first selected entry of the specified side and deselects it. Side may be 'Left', 'Right', 'Active' or 'Inactive'. If you want that 'GetEntry' returns only the selected files, use 'Files' for the TYPE/A argument. For drawers only use 'Dirs' and if you want all entries use 'Any'. If there are no entries selected of the specified type 'GetEntry' sets 'Result=RESULT'.

Example: 'GetEntry Active Files'

- LOAD SIDE/A,DIR/A

Loads a directory into the specified side. As you have guessed SIDE/A may be 'Left', 'Right', 'Active' or 'Inactive'. DIR/A is the directory to load.

Example: 'Load Active "RAM:"'

- MESSAGE STRING/A

'Message "This is a test"' writes the message 'This is a test' to the status bar. To prevent it from being overwritten by the Clock you should use 'Sleep' before and 'Wakeup' after it.

Example: 'Message "Das ist ein Test"'

- NUMENTRIES SIDE/A

Returns the number of selected entries of the requested side. SIDE/A

may be 'Left', 'Right', 'Active' or 'Inactive'. If you want to process only the selected files use 'Files' for TYPE/A. For drawers only use 'Dirs' and otherwise use 'Any'.

Example: 'NumEntries Active Dirs'

- PATH SIDE/A

With this function you will get the path currently loaded in the specified side. SIDE/A may be 'Left', 'Right', 'Active' or 'Inactive'. If there is no directory loaded 'Path' will set 'Result=RESULT'.

Example: 'Path Right'

- REFRESH SIDE/A

You may refresh the specified side with this function. SIDE/A may be 'Left', 'Right', 'Active' or 'Inactive'. If there is no directory loaded 'Path' will set 'Result=RESULT'.

Example: 'Refresh Inactive'

- SLEEP

With this function you may put RO asleep, which means that it rejects all inputs and the Clock is frozen. This function should be the first one you call in every arexx program to prevent the Clock from erasing messages you wrote into the status bar using the 'Message' command.

- STRINGREQ STRING/A

Using 'StringReq "PRT:' you will get a input requester where you may edit the string 'PRT:'. If you press 'Cancel' Result is set to 'RESULT'.

Example: 'StringReq "T:___TEMPFILE___"'

- WAKEUP

With 'Wakeup' you are able to wake RO up. Please make sure that every 'Sleep' is followed by an 'Wakeup' command otherwise RO is unusable until you call the 'Wakeup' function.

If you call an arexx program within RO for example using a button to start a script make sure that the command string contains 'Rx '. RO needs this to recognize that you are calling an arexx program. This is necessary to prevent a deadlock.

If you need the name of the selected file in your arexx script then do not use %f but %F otherwise you might get some problems with the quotes.

1.67 Discussion

Q: Sometimes RO does not recognize that I have a file selected. Why?

A: No there is another problem. It is sometimes really difficult to see if a file is selected or not because the cursor which is normally needed for keyboard control has nearly the same color as a selected file with the cursor on it. You can 'fix' this problem if you set the 'BG Listview Cursor' image to another color.

Q: I start RO and nothing happens. What's the problem?

A: Your screen resolution might be too low especially the screen width so RO doesn't fit on your screen. Try using a smaller font or use the Overscan Prefs to gain some pixels. The minimum RO needs to work are 640 x 200 and topaz 8 font.

Q: I created a new button to start a cli command but it doesn't work. Why?

A: Make sure that you created a COMMAND button and no FUNCTION button.

Q: I want to have the same register look as shown on the preview pictures. What should I do?

A: All you have to do is setting the 'Register Level' to One. This is done via the according slider on the 'Lists' Page of the MUI-Prefs program.

1.68 Registration

Registration

If you register you will receive a disk with the newest public release of RO and a personalized, so-called "keyfile" that turns the crippled version into a full functional and powerful Directory Tool! With the keyfile you can unlock all future versions of RO.

The minimum price for a RO registration is

30.- DM (D-Mark),
20.- US\$ (US Dollar)

or an equivalent amount of US\$ 25.- (twentyfive!) in any other currency. "Twentyfive" because I have to visit my bank and sell your foreign currency which is some kind of expensive here in Germany. If you think your bank does it cheaper, feel free to get some 20.- US\$ at home and send them.

The fastest, cheapest and easiest way to register is put the money together with the filled registration form into a letter and send it to:

Oliver Rummeyer

Hochbergstraße 49/1
D-88213 Ravensburg
Germany/Europe

Euro cheques (in DM) are also welcome, but please do not send any kind of foreign cheques since the bank charges outrageous fees from both the sender and the receiver. If you really feel you must use such cheques, please include extra \$10.

You can also transfer the money directly to my bank account, but beware: international transfers via banks are very expensive.

Kreissparkasse Ravensburg, BLZ 650 501 10, Konto 48-276999

In any case, be sure to provide me with your name, address, phone number and e-mail address for filing purposes. You can send this information via electronic mail, if you wish. If you don't mind, this data will be stored and processed in electronic form. You may use the sample

Registration Form
included in the documentation.

Thank you for supporting Shareware!

1.69 Registration Form

Oliver Rummeyer
Hochbergstraße 49/1
D-88213 Ravensburg

Germany/Europe

Dear Oliver,

I would like to become a registered user of "RO". Please send the disk containing the latest release and my keyfile to

Name: _____

Address: _____

Country: _____

Phone: _____

E-Mail: _____

I am sending the shareware fee via (please mark as appropriate)

- o DM 30.- in bills, coming with this letter.
- o SFr 30.- in bills, coming with this letter.
- o US\$ 20.- in bills, coming with this letter.
- o eurocheque in DM, coming with this letter.
- o foreign cheque in US\$, coming with this letter
(including extra \$10 for cashing).
- o Bank transfer.

I have read the license and distribution details and agree.

(Date)

(Signature)

1.70 Updates

Updates

Whenever a new version of RO gets released, I will post some information in the appropriate newsgroups of some electronic networks. The new archive will soon be available on many bulletin boards and on all 'aminet' FTP servers.

The RO Support Mailbox 'Uplink BBS' Sysop Alexander Abt always offers the most recent release for Fido file request at these nodes:

```
Node 1: +49-0751-96217 2:246/2320 USR 28.8 DS
Node 2: +49-0751-97522 2:246/2321 USR 16.8 DS
Node 3: +49-0751-97251 2:246/2322 USR 14.4 DS
Node 4: +49-0751-97261 2:246/2323 USR 14.4
Node 5: +49-0751-97376 2:246/2324 Zyx E+
```

Use the magic name 'RO'. Alternatively you can login with a user name of 'RO' and a password of 'RO'.

As mentioned above, registered users will neither need a new keyfile nor a special personalized program version. They can use all new features immediately.

1.71 Support

Support

If you have suggestions, questions, critics or bugreports or you simply like my program, the write to:

SMail: Oliver Rummeyer
 Hochbergstraße 49/1
 D-88213 Ravensburg
 Germany/Europe

Email: Internet oliver.rummeyer@student.uni-ulm.de
 oliver@rumor.rv.sub.org
 Fidonet 2:246/2320.1@fidonet

-----BEGIN PGP PUBLIC KEY BLOCK-----
 Version: 2.6ui (Amiga)

```
mQBNAi9OUNkAAAEcAMMunTdwFA+DX4SksnWNl0QqP4uAkrStXnHoVTb1Yaz59TMC
Nw/2lMP0QO1G+4qIpIhZ6ayQFrkAeSWtkRG+r4EABRG0KU9saXZlciBSdW1tZXll
ciA8b2xpdmVyQHJlbW9yLnJ2LnN1Yi5vcmc+
=iYz8
-----END PGP PUBLIC KEY BLOCK-----
```

1.72 History

History

- Version 0.80 (26.8.94)

Initial Release.

- Version 0.81 (7.9.94)

- o Fixed some minor bugs.
- o Added a configuration switch to toggle path extension.

- Version 0.82 (11.10.94)

- o Fixed some bugs.
- o Uses the CLI command path if started from Workbench.
- o Added some filetypes.
- o Added command line option
 ICONIFY
 .
- o Added config variable
 FLUSHMEMORY
 .
- o Removed config variable 'HELPPFILE' use 'ENV:AmigaGuide/Path' instead.
- o Fully localized (german catalog included).
- o Command buttons support keyboard shortcuts.
- o Many other minor changes.

- Version 0.83 (5.12.94)

- o Fixed some minor bugs.
 - o Added some filetypes.
 - o New internal Functions:
 - UPDATE
 - &
 - CURRENT
 - o German documentation included.
 - o Catalog files changed.
 - o Included italian catalog file.
 - o Major code cleanup.
 - o Other minor changes.
- Version 0.90 (9.1.95)
- o requires MUI 2.3 or better.
 - o Fixed all known bugs.
 - o Added some filetypes.
 - o Added config variables
 - FORMATLEFT
 - &
 - FORMATRIGHT
 - .
 - o Added new config keyword
 - HOTKEY
 - .
 - o Added config variable
 - COPYVERIFY
 - .
 - o Added config variable
 - SYMMETRICAL
 - .
 - o Improved listview activation.
 - o Changed command strings in configuration.
Added many new '%' parameters.
 - o Changed archive configuration.
 - o Removed Functions 'COMMAND', 'EXECUTE' and 'REXX'.
 - o Enhanced
 - arexx interface
 - supports all basic functions.
 - o Catalog files changed.
 - o Many other minor changes.
- Version 0.91 (23.1.95)
- o Fixed some minor bugs.
 - o Added new '&' parameters.
 - o Changed archive default commands.
 - o Changed overwrite & unprotect requester.
 - o Included some arexx examples.
 - o Catalog files changed.
 - o Improved installer script.
 - o Information window displays version string now.
- Version 1.00 (20.3.95)
- o Fixed some bugs.
 - o Catalog files changed.
 - o Minor code cleanup.
-

- o Added user configurable filetypes.
 - o Changed
 - filetype
 - configuration.
 - o Changed
 - archive
 - configuration.
 - o Removed config variable ARCHANDLER.
 - o Added new and improved old arexx scripts.
- Version 1.01 (22.3.95)
- o Fixed some major bugs.
- Version 1.02 (3.4.95)
- o Fixed some bugs.
 - o Catalog files changed.
 - o Improved german documentation.
 - o New menu entry
 - Last Message
 - .
 - o New menu entry
 - Update All
 - .
- Version 1.03 (4.6.95)
- o Fixed some bugs.
 - o Improved installation script.
 - o Improved default handling for unknown filetypes.
 - o Added config variable
 - CHECKDUPES
 - .
 - o Added '%a' parameter to get arexx port name.
- Version 1.04 (9.7.95)
- o Fixed some minor bugs.
 - o Catalog files changed.
 - o French documentation included.
 - o Included french catalog file.
 - o Other minor changes.
 - o Added list cycle with middle mouse button.
 - o Added button bank switching with middle mouse button.
 - o New menu entry
 - Spawn New
 - .

1.73 Future Enhancements

Future Enhancements

- o Configuration Editor
-

1.74 Acknowledgments

Acknowledgments

The author wishes to thank his beta testers...

- Alexander Abt

...for his 'Recog' filetype recognition.

- Frank Schlichting

...for being my first beta tester, for his great suggestions
and for reworking the documentation.

- Juergen Schubert

...for his ingenious suggestions and for writing the installer
script.

...and...

- Rafael D'Halleweyn

...for his "ARCHandler".

- Manuel Giorgini and Marco Musso

...for the italian localization.

- Jean-Francois Fabre

...for the french documentation and catalog file.

- Martin "XEN" Hutteloher

...for his MagicWB package and his permission to use some icons
within this package.

- Stefan Stuntz

...for his "Magic User Interface".

...and last but not least the registered users.

1.75 Disclaimer

Disclaimer

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.76 License

License

- You may use RO for a period of up to 14 days of evaluation. After that you have to register.
 - You may copy and distribute verbatim copies of the program's executable code and documentation as you receive it, in any medium, provided that you conspicuously and appropriately publish only the original, unmodified program, with all copyright notices and disclaimers of warranty intact and including all the accompanying documentation, example files and anything else that came with the original.
 - Except when otherwise stated in this documentation, you may not copy and/or distribute this program without the accompanying documentation and other additional files that came with the original. You may not copy and/or distribute modified versions of this program.
 - You may not copy, modify, sublicense, distribute or transfer the program except as expressly provided under this license. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the program is void, and will automatically terminate your rights to use the program under this license. However, parties who have received copies, or rights to use copies, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.
 - By copying, distributing and/or using the program you indicate your acceptance of this license to do so, and all its terms and conditions.
 - Each time you redistribute the program, the recipient automatically receives a license from the original licensor to copy, distribute and/or use the program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
 - You may not disassemble, decompile, re-source or otherwise reverse engineer the program.
 - You agree to cease distributing the program and data involved if
-

requested to do so by the author.

1.77 Installer

Installer

Along with RO comes the 'Installer' from Commodore:

Installer and Installer project icon
(c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.
Reproduced and distributed under license from Commodore.

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE;
NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY
OR RESPONSIBILITY IS ASSUMED.

1.78 Magic User Interface

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.79 index

A

About

Acknowledgments

Action

All

APPWINDOW

ARCHIVETYPE

ARexx Interface

Assign

B

BINARY

BUFFER

BUTTON

ByName

Bytes

C

Change

CHECKFIT

Command Buttons

Configuration

Copy

CopyAs

COPYDATE

CopyDev

COPYFLAGS

COPYNOTE

COPYVERIFY

CURRENT

Current

CUSTOMTYPE
D

DEFAULT

Delete

DELFORCE

DELVERIFY

Disclaimer

Discussion

Disk

Drive Buttons

DRIVE

Dup
E

Expand
F

FILETYPE

FIRSTLEFT

FIRSTRIGHT

Fit

FLUSHMEMORY

FORMATLEFT

FORMATRIGHT

Future
H

Help

History

HOTDIR

HOTKEY

How to install RO

How to register for RO
I

Icon

ICONIFY

Info

Information Textfield

Installer

L

LActive

LCopy

License

ListArc

Listview

LParent

LReload

LRoot

LSwap

M

Magic User Interface

Main Window

MakeDir

MDLoad

MENU

Menus

Move

MoveAs

N

None

Note

O

OVERWRITE

P

Parent

Path Gadget

Protect

Q

Quit

R

RActive

Reboot

Register

Registration Form

Relabel

Reload

Rename

Root

RParent

RReload

RRoot

S

SetDate

Short description

Shrink

SORTLEFT

SORTRIGHT

Status Bar

Support

SYMMETRICAL

System Requirements

T

Toggle

Touch

U

UnArc

Unregistered Version

UPDATE

Update

Updates

V

VARIABLE
