

**ReOrg\_English**

**COLLABORATORS**

	<i>TITLE :</i> ReOrg_English		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 16, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ReOrg_English</b>	<b>1</b>
1.1	ReOrg_English.guide	1
1.2	ReOrg_English.guide/NODE_DISCLAIMER	2
1.3	ReOrg_English.guide/NODE_CONDITIONS	3
1.4	ReOrg_English.guide/NODE_INTRODUCTION	5
1.5	ReOrg_English.guide/NODE_REQUIREMENTS	7
1.6	ReOrg_English.guide/NODE_INSTALLATION	9
1.7	ReOrg_English.guide/NODE_PROGSTART	10
1.8	ReOrg_English.guide/NODE_MENU	12
1.9	ReOrg_English.guide/NODE_M_ABOUT	12
1.10	ReOrg_English.guide/NODE_M_HELP	12
1.11	ReOrg_English.guide/NODE_M_QUIT	13
1.12	ReOrg_English.guide/NODE_M_ICONS	13
1.13	ReOrg_English.guide/NODE_M_LOAD	13
1.14	ReOrg_English.guide/NODE_M_SAVE	14
1.15	ReOrg_English.guide/NODE_M_SAVEA	14
1.16	ReOrg_English.guide/NODE_M_PREDEF	14
1.17	ReOrg_English.guide/NODE_GADGETS	15
1.18	ReOrg_English.guide/NODE_G_FORMAT	16
1.19	ReOrg_English.guide/NODE_G_VERIFY	17
1.20	ReOrg_English.guide/NODE_G_WBENCH	17
1.21	ReOrg_English.guide/NODE_G_GRAPH	18
1.22	ReOrg_English.guide/NODE_G_OPT	18
1.23	ReOrg_English.guide/NODE_G_SOPT	18
1.24	ReOrg_English.guide/NODE_G_CLOPT	19
1.25	ReOrg_English.guide/NODE_G_DRIVE	19
1.26	ReOrg_English.guide/NODE_G_FROM	19
1.27	ReOrg_English.guide/NODE_G_TO	19
1.28	ReOrg_English.guide/NODE_G_MODE	20
1.29	ReOrg_English.guide/NODE_G_START	20

---

---

1.30	ReOrg_English.guide/NODE_G_CANCEL	20
1.31	ReOrg_English.guide/NODE_G_DATE	20
1.32	ReOrg_English.guide/NODE_G_LOCKS	21
1.33	ReOrg_English.guide/NODE_G_SIMUL	21
1.34	ReOrg_English.guide/NODE_G_GEO	22
1.35	ReOrg_English.guide/NODE_G_CLR	22
1.36	ReOrg_English.guide/NODE_G_FEXT	23
1.37	ReOrg_English.guide/NODE_G_FREEB	23
1.38	ReOrg_English.guide/NODE_G_1320	24
1.39	ReOrg_English.guide/NODE_G_CHFS	24
1.40	ReOrg_English.guide/NODE_G_CHOFS	25
1.41	ReOrg_English.guide/NODE_G_MEM	25
1.42	ReOrg_English.guide/NODE_G_LIMIT	26
1.43	ReOrg_English.guide/NODE_ONLINEHELP	26
1.44	ReOrg_English.guide/NODE_OPTIMIZATION	27
1.45	ReOrg_English.guide/NODE_FSCONVERSION	29
1.46	ReOrg_English.guide/NODE_OPTIONFILES	31
1.47	ReOrg_English.guide/NODE_COMPAT	32
1.48	ReOrg_English.guide/NODE_TECHDETAILS	33
1.49	ReOrg_English.guide/NODE_ERRORMESSAGES	38
1.50	ReOrg_English.guide/NODE_UPDATES	41
1.51	ReOrg_English.guide/NODE_HISTORY	42
1.52	ReOrg_English.guide/NODE_ACKNOWLEDGEMENTS	43

---

# Chapter 1

## ReOrg\_English

### 1.1 ReOrg\_English.guide

ReOrg

\*\*\*\*\*

This is the documentation for ReOrg V3.1, a fast shareware disk optimizer for AmigaDOS. Copyright (C) 1993 Holger Kruse. All rights reserved.

Disclaimer

Legal information

Usage / Copying

Usage and copying conditions

Introduction

Introduction to ReOrg

Requirements

Required hardware and software

Installation

How to install ReOrg

Program start

How to start ReOrg. List of command line options and Tooltypes

Menus

Description of all Menus

Gadgets

Description of all Gadgets

Online help

ReOrg's online help feature

Optimization

	How to optimize a disk
FileSystem Conversion	How to convert a disk to a different FileSystem
Option Files	Format of option files
Compatibility / Limitations	Compatibility risks
Technical Details	Description of the inner workings of ReOrg
Error Messages	List of all ReOrg error messages
Update information	How to get the latest ReOrg version
History	History of ReOrg
Acknowledgements	Acknowledgements

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

AmigaGuide, AmigaGuide.info, amigaguide.library, WDisplay,  
WDisplay.info (c) Copyright 1992 Commodore-Amiga, Inc. All Rights  
Reserved. Reproduced and distributed under license from Commodore.

## 1.2 ReOrg\_English.guide/NODE\_DISCLAIMER

Disclaimer

\*\*\*\*\*

ReOrg IS SUPPOSED TO BE A DISK-OPTIMIZER AND FILESYSTEM-CONVERTER.  
OPERATIONS OF THAT TYPE ARE ALWAYS RISKY, BECAUSE THEY BYPASS MOST  
SECURITY CHECKS IN THE OPERATING SYSTEM. IT IS THEREFORE VERY WELL  
POSSIBLE THAT ReOrg BEHAVES INCORRECTLY AND LEAVES YOUR DISK AND/OR  
YOUR DATA IN AN INCONSISTENT AND/OR DEFECTIVE STATE.

I HEREBY REJECT ANY LIABILITY OR RESPONSIBILITY FOR THESE OR ANY OTHER  
CONSEQUENCES FROM THE USE OF ReOrg WHATSOEVER. THIS INCLUDES, BUT IS  
NOT LIMITED TO, DAMAGE TO YOUR EQUIPMENT, TO YOUR DATA, PERSONAL  
INJURIES, FINANCIAL LOSS OR ANY OTHER KINDS OF SIDE EFFECTS.

ReOrg is PROVIDED AS-IS. THIS MEANS I DO NOT GUARANTEE ANY BUG FIXES,

UPDATES OR HELP IN ERROR RECOVERY. ALTHOUGH ReOrg HAS BEEN TESTED THOROUGHLY ON SEVERAL DIFFERENT MACHINES, I CANNOT RULE OUT THE POSSIBILITY THAT ReOrg

- \* IS SOMEHOW INCOMPATIBLE TO YOUR EQUIPMENT.
- \* HAS BUGS THAT SHOW UP ON YOUR EQUIPMENT.
- \* DOES NOT DO WHAT IT IS SUPPOSED TO DO ON YOUR EQUIPMENT.

IT IS SOLELY YOUR RESPONSIBILITY TO

- \* REGULARLY MAKE BACKUP COPIES OF YOUR DATA.
- \* VERIFY BEFORE AND AFTER AN OPTIMIZATION, THAT YOUR DATA IS CONSISTENT AND HAS NOT BEEN CHANGED BY ReOrg.
- \* ENSURE THAT ReOrg WORKS ON YOUR EQUIPMENT BY FIRST RUNNING IT ON UNCRITICAL DATA, LIKE SMALL TEST PARTITIONS.

WARNING: EVEN IF ReOrg SEEMS TO WORK WELL ON SOME OR ALL OF YOUR TEST EQUIPMENT, IT MAY VERY WELL FAIL ON DIFFERENT SETUPS OR ON THE SAME SETUP AT DIFFERENT TIMES.

I THEREFORE STRONGLY RECOMMEND TO BACKUP EACH AND EVERY OF YOUR DISKS AND VOLUMES BEFORE USING ReOrg. IT IS EVEN POSSIBLE, ALTHOUGH VERY UNLIKELY, THAT ReOrg DAMAGES DATA ON DIFFERENT VOLUMES THAN THOSE IT IS CURRENTLY WORKING ON.

IT IS YOUR RESPONSIBILITY TO MAKE ANY PRECAUTIONS NECESSARY TO PROTECT YOURSELF FROM THESE OR ANY OTHER EFFECTS. I EXPLICITLY REJECT ANY LIABILITY OR RESPONSIBILITY FROM THE CONSEQUENCES OF YOU USING ReOrg.

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.

AMIGAGUIDE 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.

## 1.3 ReOrg\_English.guide/NODE\_CONDITIONS

Usage / Copying

\*\*\*\*\*

ReOrg is shareware. The program may be freely distributed and copied, as long as the following conditions are fulfilled:

- \* The sales price must not be higher than the cost of an (empty) disk plus a nominal copying fee plus costs for shipping. The total price must not be higher than 6 US\$ or 10 DM or the equivalent in any other currency.
  - \* All parts of the program and the documentation must be complete. The distribution of single parts or incomplete subsets of the original distribution is not allowed.
-

- \* ReOrg or parts of it may not be sold in combination with or as part of commercial software. However, I may allow exceptions to that restriction in individual cases. If you want to spread ReOrg as a supplement to a commercial program, please contact me. Until you receive a written permission from me, do not assume you may distribute ReOrg with or as part of your commercial program.
- \* Program and documentation may not be changed in any way. Exception (this means: acceptable) is the use of archivers such as 'LHA' and packers like 'Imploder' or 'Powerpacker', as long as it remains possible to retrieve the original program/data.

ReOrg is shareware, this means I request those users, who regularly use ReOrg, to send me the shareware fee of 10 US\$.

Even if you have already paid money to get your copy of ReOrg (for example from an electronic bulletin board or from a public domain dealer), this does not include the shareware fee, because I, the author of ReOrg, do not receive a share of those fees.

I ask everybody, who uses ReOrg (except for one or two test runs), to send the amount of

10 US\$ or 15 DM

to the following address:

Holger Kruse  
12006 Coed Drive  
Orlando, FL 32826  
USA

Phone: +1-407-381-3233  
Internet: kruse@cs.ucf.edu  
BIX: hkruse

Please send only cash, American checks, US postal money orders or international money orders (US\$ only !). Do not send stamps, disks, Eurocheques etc.

Please include the following information in your registration:

- \* your name, address (including any electronic mail addresses you might have) and telephone number.
- \* your Amiga model(s).
- \* the type of harddrive and controller you use.
- \* Kickstart and Workbench versions.
- \* the ReOrg version number (currently 3.1).
- \* If you experience problems using ReOrg, please send me a detailed list of ALL hardware and software (including PD utilities) !

I assure you that the information you send me will be treated

---



confidentially.

Thank you very much in advance !

If you send me your registration during the US summer term (May - July), please expect a LONG delay until you receive answer. I might not be at home for one or two months - sorry !

I will try and inform everybody, who has registered with me as described above, when a new version of ReOrg becomes available.

Former users of earlier ReOrg versions (V1.x or V2.x) do not need to register again for V3.1, of course. The upgrade from V1.x and V2.x to V3.1 is free.

## 1.4 ReOrg\_English.guide/NODE\_INTRODUCTION

### Introduction

\*\*\*\*\*

This chapter summarizes the features of ReOrg. Please be sure you have read

Disclaimer  
and  
Usage / Copying  
before you use ReOrg.

ReOrg is a disk optimizer, i.e. a program that improves the physical data layout on a floppy disk or harddisk, in order to speed up file and directory accesses.

ReOrg solves some problems that are caused by AmigaDOS:

- \* file fragmentation: When new files are created or existing files are extended, sometimes AmigaDOS may not be able to store all file contents in one single consecutive block, because there is no such block available on the disk. In this case AmigaDOS sometimes scatters the file contents over the whole disk. ReOrg removes this problem by placing the contents of each file in a single consecutive block.
  - \* directory fragmentation: In contrast to other operating systems (e.g. MS-DOS), AmigaDOS does not store directory data in a single consecutive block. This leads to weak performance for directory listings, unless the new Directory-Cache-FileSystem (DOS\4,DOS\5) is used. ReOrg tries to store directory data as closely together as possible. This considerably speeds up commands like 'dir' and 'list' and the directory display within file requesters.
  - \* differences in disk format: When you use disks under Kick1.2/1.3, which have been formatted under Kick2.04, directory accesses are often even much slower than usual. The reason for this is that the placement of "FileList"-blocks has been changed from Kick1.2/1.3 to Kick2.04. ReOrg has an option where to place "FileList"-blocks. This means you can optimize Kick2.04-disks for use with
-

Kick1.2/1.3.

- \* smarter directory handling under Kickstart 2.0x: Kickstart 2.0x and higher have a more efficient way to retrieve directory contents. This makes it possible for a disk optimizer to arrange the files within a directory in any arbitrary order without reducing performance. Files could e.g. be arranged in a way that Icons appear on Workbench very fast.
- \* inconsistencies in disk layout: Due to the very complex disk structure of the AmigaDOS filing system, disks can easily get corrupted, if you use wrong, obsolete disk tools or switch off your computer at the "wrong" time. ReOrg does a thorough (but not complete) check of your disk before optimizing it, and issues an error messages if it finds any defects. ReOrg can even automatically repair some kinds of defects, like
  - files that show up in a directory but cannot be accessed.
  - most checksum errors.

ReOrg can be used with the following disk formats:

- \* all currently available AmigaDOS filesystems, including
    - OldFileSystem (OFS: "DOS\0").
    - FastFileSystem (FFS: "DOS\1").
    - Kick2.0-FileSystem (includes OFS and FFS).
    - International FileSystem (has been in ROM at least since Kick2.04 and is supported from the Shell since WB2.1: "DOS\2", "DOS\3").
    - Directory-Cache-FileSystem (has been in ROM since Kick 3.0: "DOS\4", "DOS\5").
  - \* All floppy disk and harddisk sizes are supported. However, to optimize large hard disk partitions, you need a lot of memory.
  - \* ReOrg recognizes drives that support different disk capacities (e.g. High-Density-Drives for 1.76 MB capacity that are built in some new Amiga models).
  - \* ReOrg supports partitions with block sizes of 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768 bytes. There are two ways to specify the block size: by using the BlockSize keyword in your mountlist or by changing the number of SectorsPerBlock (that is what new versions of HDToolbox do). Both methods are supported by ReOrg.
  - \* ReOrg supports "hard links" and "soft links"; however, "soft links" are not completely supported by Commodore yet. Therefore, ReOrg only supports the (possibly preliminary) "soft link" implementation of FileSystem version V36 to V40. I can not guarantee that this version of ReOrg correctly handles "soft link" implementations of future FileSystem versions.
-

- \* CAUTION: DYNAMIC Ram disks, like "VD0:" ((C) ASDG) must NOT be optimized by ReOrg. Optimizing such disks may cause files on those disks to be destroyed !
- \* Of course, ReOrg can not optimize "RAM:". Anyway, this would not make much sense.
- \* You CAN use ReOrg to optimize your resident, non-dynamical RAM-Disk "RAD:" ((C) Commodore).
- \* ReOrg has a one-disk-mode (optimize a single disk) and a two-disk-mode (optimize one disk while copying it to another disk).

Compared to most other disk optimizers available for the Amiga, ReOrg is quite fast. Optimizing a 880 kB disk on a standard 68000 Amiga with 1 MB of free memory usually takes about 1:30 to 1:40 minutes. On a 68030 Amiga it takes only 1:20 to 1:30 minutes. Optimizing a 40 MB harddisk on a 68000 Amiga usually takes around 10 to 15 minutes, but this time depends very much on the amount of free memory, type of controller, type of processor and the degree of disk fragmentation. If you use a 68030 or 68040 Amiga and have enough memory available, the optimization time can be reduced to less than 3 minutes.

Note: If ReOrg takes very long to optimize your harddrive (e.g. more than half an hour for a 40 MB drive) please check if the value for "Sectors per track" in your harddisk installation software is correct. "Sectors per track" should not be smaller than 10. Some harddisk installation software lets you specify very small values (like 1) and - to compensate for this - a very large number of cylinders and/or heads. Although your harddisk usually works correctly with settings like that, it is better to use the correct value. If you want to use ReOrg, please make at least sure that the value you use is "reasonable" (i.e. it should not be larger than 200 and not be smaller than 10). Otherwise ReOrg might need VERY MUCH memory or VERY MUCH time for the optimization.

In addition to defragmenting a disk, ReOrg 3.1 can also be used to convert a disk to a different filesystem, e.g. to convert an old DOS\0 disk to the new DOS\5 format, that offers a larger capacity and faster directories.

Any filesystem can be converted to any other filesystem. However, for some conversions, there are limitations to the size of the disk and the amount of data on a disk. For details, see

FileSystem Conversion

.

## 1.5 ReOrg\_English.guide/NODE\_REQUIREMENTS

Requirements

\*\*\*\*\*

To use ReOrg, you need an Amiga, Kickstart 2.04 or above and enough

memory.

If you want to use ReOrg with other languages than English, it is recommended you use Workbench 2.1 or 3.0 (which includes locale.library V38 or higher). However, ReOrg DOES support languages other than English even under Workbench 2.04.

There is also a different version of ReOrg (V1.13) available that can be used under Kickstart 1.2/1.3, but that version has much fewer features, has no graphical user interface and requires more memory than ReOrg V3.1.

The amount of memory you need is approximately:

```

400 kB (needed for the program itself)
+   7 kB for each MB of disk size
+   6 * tracksize of your disk
+ 150 kB of safety memory

```

The safety memory is usually not used by ReOrg, but must be left to avoid low-memory situations. (For large partition sizes and much memory available, this number may become higher.) The Safety Memory gadget (see

```

Gadget Safety Memory
) allows you to change this value.

```

Examples:

<pre> 880 kB floppy disk: (11 sectors/track)  400 kB +   6 kB (7 kB * 880/1024) +  33 kB (11/2 kB * 6) + 150 kB ----- 589 kB </pre>	<pre> 40 MB harddisk: (32 sectors/track)  400 kB + 280 kB (7 kB * 40) +  96 kB (32/2 kb * 6) + 150 kB ----- 926 kB </pre>
---	---

```

200 MB harddisk:
(32 sectors/track)

400 kB
+ 1400 kB (7 kB * 200)
+  96 kB (32/2 kB * 6)
+ 150 kB
-----
2046 kB

```

These numbers are only rough approximations to give you an idea how much memory you need.

Actually, for large partition sizes ReOrg allocates most of your memory as a cache. This cache is the main reason why ReOrg is so fast. For optimum speed, the cache should be no smaller than 2 percent of the size of your partition.

---

Note: If you get an error message "Not enough memory" although you think you have enough memory for the optimization, please check if the value for "Sectors per track" in your harddisk installation software is correct. "Sectors per track" should not be higher than 200. Some harddisk installation software lets you specify much larger values and - to compensate for this - only a small number of cylinders. Although your harddisk usually works correctly with settings like that, it is better to use the correct values. If you want to use ReOrg, please make at least sure that the value is "reasonable" (i.e. it should not be larger than 200 and not be smaller than 10). Otherwise ReOrg might need VERY MUCH memory or VERY MUCH time for the optimization.

## 1.6 ReOrg\_English.guide/NODE\_INSTALLATION

### Installation

\*\*\*\*\*

The 'Installer' utility ((C) by Commodore) is included in this distribution and will install ReOrg on your harddisk automatically. There are installation scripts for different languages in the 'Install' directory of the original disk.

Unfortunately, due to licensing restrictions I cannot include a localized version of the 'Installer' utility on this disk. This means all Installer-specific texts during the installation process are always in English, no matter which language you started the installation program in.

If you have a localized version of the 'Installer' utility, you can use that to install ReOrg:

- \* Install the localized 'Installer' program somewhere in your path on your harddisk.
  
- \* Select the ReOrg installation icon (e.g. 'Install' in the 'English' drawer) and use the Workbench menu item 'Icons/Information' to change the "Default Tool" to 'Installer' (remove the '/' at the beginning of '/Installer').

If you do not have a harddisk, you can make a copy of the original disk and run ReOrg directly from that disk.

The online help feature requires 'amigaguide.library' to be installed in your 'libs:' directory. Usually the installation program will take care of this. However if you run ReOrg directly from the original disk without installing it first, make sure you have 'amigaguide.library' installed, or you will not be able to use the online help feature.

## 1.7 ReOrg\_English.guide/NODE\_PROGSTART

Program start

\*\*\*\*\*

You can start ReOrg from Workbench or Shell. ReOrg recognizes several Tooltypes (if started from Workbench) or arguments (if started from Shell):

``FROM=sourcedisk'`

Specifies the source disk for optimization.

``TO=destinationdisk'`

Specifies the destination disk for optimization. If the ``TO'` keyword is not specified, ReOrg will switch to One-Disk-Mode. Even if you specify ``FROM'` and/or ``TO'`, ReOrg will not automatically start the optimization, unless you use the ``BATCH'` keyword. ``TO'` and ``FROM'` only cause ReOrg to initially select a disk for optimization.

``PUBSCREEN=pubscreenname'`

Specifies the name of the public screen that ReOrg will try to open on (default = Workbench screen).

``OPTFILE=optionfilename'`

Specifies the name of the option file used for optimizations. See

Option Files

for details about the format and use of option files.

``SETTINGS=settingsfilename'`

Specifies the name of the settings (preferences) file. If you do not specify a file name, ReOrg uses ``ReOrg.prefs'` as default.

``SCREENMODE=screenmode'`

If you specify the ``SCREENMODE'` keyword, ReOrg will open its window on a separate public screen. ``screenmode'` can be

\* a mode name like ``NTSC:High Res'`

\* a mode ID like `0x8000`

However, the use of a mode name is discouraged, because mode names change with different operating system versions and different languages. Therefore please use mode IDs whenever possible. Here are some examples:

\* `0x8000` = High-Res

\* `0x19000` = NTSC:High Res

\* `0x29000` = PAL:High Res

\* `0x8004` = High Res Interlace

\* `0x41000` = A2024: 10 Hz

\* `0x49000` = A2024: 15 Hz

A complete list of mode IDs can be found in the file  
'graphics/displayinfo.h' (part of commercial "C" development  
packages).

#### 'LANGUAGE=language'

Specifies the language ReOrg will use for its user interface. If  
this keyword is NOT specified, ReOrg will behave differently  
depending on the operating system you use.

- \* Workbench 2.04: ReOrg will display all texts in English.

- \* Workbench 2.1, 3.0 or later versions: ReOrg will use the  
topmost of your "preferred languages" you have selected in  
Locale Preferences that ReOrg supports. If none of your  
"preferred languages" is supported by ReOrg, ReOrg will fall  
back to "English".

Note: ReOrg supports the 'LANGUAGE' keyword even under Workbench  
2.04 ! This means that you do not necessarily need  
'locale.library' to run ReOrg in a language other than English.  
However, Workbench versions earlier than 2.1 do not support the  
concept of a system-wide "default language". Thus if you want  
ReOrg to use a language other than English under Workbench 2.04,  
you must use the 'LANGUAGE' keyword to specify the language you  
want !

#### 'HIDDEN'

The 'HIDDEN' keyword causes ReOrg to display its public screen  
behind all other screens and/or not to auto-activate its window to  
front when it is opened. This keyword can be useful in combination  
with the 'BATCH' keyword.

#### 'BATCH'

The 'BATCH' keyword causes ReOrg to auto-start the optimization  
and answer all standard requesters with "Yes". Use this option  
with caution, because ReOrg will automatically start the  
optimization after scanning your disk (without asking you first) !  
If ReOrg encounters an error during the optimization, it will  
leave batch mode, pop its screen and window to front and show an  
error requester. If you use 'BATCH' in combination with 'HIDDEN',  
ReOrg will try to perform the optimization silently without  
interfering with any other work on your Amiga. Note: Even if you  
use 'BATCH' and 'HIDDEN', ReOrg will still open its graphical user  
interface in the background - it only helps you to reduce the  
number of "mouse-clicks" needed to optimize a disk.

#### 'PENS=pen0,pen1,pen2,pen3'

This option lets you specify the pen numbers ReOrg uses for its  
graphical sector display. The default is

- \* pen0 = BACKGROUNDPEN (always color 0)

- \* pen1 = TEXTPEN (usually color 1)

- \* pen2 = HIGHLIGHTTEXTPEN (usually color 2)

- \* pen3 = FILLPEN (usually color 3)

---

If ReOrg has been started from Workbench, it creates an AppIcon on Workbench, and the drive selection gadgets act as AppWindows.

This means if you drag a disk icon over ReOrg's AppIcon or over one of the drive selection gadgets, ReOrg sets this gadget to the corresponding device name.

## 1.8 ReOrg\_English.guide/NODE\_MENU

Menus

\*\*\*\*\*

ReOrg / About ReOrg

ReOrg / Help

ReOrg / Quit ReOrg

Settings / Create Icons

Settings / Load Settings

Settings / Save Settings

Settings / Save Settings as

Predefined settings

## 1.9 ReOrg\_English.guide/NODE\_M\_ABOUT

Menu item "ReOrg / About ReOrg"

=====

"About ReOrg" shows you the internal ReOrg version number and the address of the programmer. Please send US\$ 10 to this address if you decide to register - thank you !

## 1.10 ReOrg\_English.guide/NODE\_M\_HELP

Menu item "ReOrg / Help"

=====

"ReOrg / Help" tries to show the top level help page of ReOrg. This is only possible if you have installed 'amigaguide.library' in your

---



system. Usually, the installation process does that for you. However, if you have run ReOrg directly from the distribution disk, you might not have 'amigaguide.library' installed, so the help function might not work.

Selecting this menu item is slightly different from pressing the Help key: The Help key shows you context-sensitive help, i.e. it shows you help about the gadget under the mouse pointer, whereas the menu item "ReOrg / Help" always shows you the top-level help page.

ReOrg tries to use the same language for the help pages, that is used for all other texts in ReOrg. However, if you have not installed the help texts for that language, ReOrg will show the help pages in a different language.

## 1.11 ReOrg\_English.guide/NODE\_M\_QUIT

Menu item "ReOrg / Quit ReOrg"

=====

Selecting this menu item leaves the program.

## 1.12 ReOrg\_English.guide/NODE\_M\_ICONS

Menu item "Settings / Create Icons"

=====

This menu item is a toggle switch. If it is turned on, ReOrg creates an icon ('.info' file) when you save the settings, so you can later double-click on the settings file from Workbench to start ReOrg using those settings.

The default setting for "Settings / Create Icons" is "On" if you started ReOrg from Workbench and "Off" if you started ReOrg from Shell.

## 1.13 ReOrg\_English.guide/NODE\_M\_LOAD

Menu item "Settings / Load Settings"

=====

This menu item lets you load a new settings file. ReOrg will display a file requester asking you for a filename and then load the file and redisplay all gadgets to reflect the new settings.

Although ReOrg 3.1 can read settings files from all previous ReOrg versions, this is not recommended, because ReOrg 3.1 has some new options that earlier ReOrg versions did not have. If you load an old

settings file version, ReOrg will switch off all new options that are not specified in the settings file.

Older ReOrg versions cannot read new ReOrg 3.1 settings files.

## 1.14 ReOrg\_English.guide/NODE\_M\_SAVE

Menu item "Settings / Save Settings"

=====  
This menu item saves the current settings in the most recently used ReOrg settings file. Usually this is 'PROGDIR:ReOrg.prefs', meaning the file 'ReOrg.prefs' in the directory where the program file 'ReOrg' is located.

## 1.15 ReOrg\_English.guide/NODE\_M\_SAVEA

Menu item "Settings / Save Settings as"

=====  
This menu item saves the current settings. ReOrg will display a file requester to ask you for the file name, and then save its settings to that file. An '.info'-file will be created if you have switched "Settings / Create Icons" on.

## 1.16 ReOrg\_English.guide/NODE\_M\_PREDEF

Menu "Predefined Settings"

=====  
The menu "Predefined Settings" can be used to set most gadgets to suitable values for typical types of optimization. These are:

- \* Optimizing a harddisk
  
- \* Optimizing a floppy disk while copying it (e.g. DF0 to DF1)
  
- \* Optimizing a floppy disk with only one drive
  
- \* Optimizing a floppy disk for Kickstart 1.3 while copying it
  
- \* Optimizing a floppy disk for Kickstart 1.3 with only one drive

All of these predefined settings set the following gadgets to appropriate values: "Format", "Verify", "Mode", "Update disk date", "Keep Locks", "Simulate optimization", "Read drive geometry", "FileExt blocks", "Optimize for", "Change FileSystem" and "Change OFS/FFS". Other settings are not affected.

---

These menu items make it easier to select "typical" settings without actually changing the setting of each gadget one at a time.

Example: To optimize ANY harddisk partition:

\* First select the menu item "Predefined settings / Harddisk DH0".

\* Then, if you want to optimize a different partition than DH0, click on the correct partition name in the drive selection gadget.

By following this procedure, you can configure ReOrg properly for most types of optimizations with only two mouse clicks.

## 1.17 ReOrg\_English.guide/NODE\_GADGETS

Gadgets

\*\*\*\*\*

Format

Write Verify

Workbench Mode

Graphical sector display

Option file

Select option file

Clear option file

Drive

From drive

To drive

Mode

Start

Cancel

Update disk date

Keep Locks

Simulate optimization only

Read drive geometry

---

Clear unused blocks

FileExt blocks

Free blocks

Optimize for (Kick1.3/2.0)

Change FileSystem

Change OFS/FFS

Safety memory

Small file limit

## 1.18 ReOrg\_English.guide/NODE\_G\_FORMAT

Gadget "Format"  
=====

This gadget lets you specify whether ReOrg should format the output disk. There are three possible settings:

'Off'

ReOrg will only use standard 'CMD\_WRITE' commands. This is the default for One-Disk-Mode. You can also use that option in Two-Disk-Mode, but only if the destination disk has already been formatted. If you use ReOrg with floppy disks, you should use the setting 'On' instead of 'Off', because 'On' is faster for all current versions of 'trackdisk.device'.

'On'

This causes ReOrg to format each track that is used by ReOrg when it is first written to. This option can be used in One-Disk-Mode and Two-Disk-Mode. It is the preferred setting in One-Disk-Mode for floppy disks, but do not use this option if your destination disk is unformatted ! For all other devices (custom devices, ram drives etc.), 'Off' is usually the better choice.

'All tracks'

Causes ReOrg to completely format the destination disk. If you are using a new, unformatted disk as the destination disk (in Two-Disk-Mode), you MUST use this option or parts of the destination disk will be unusable. This option cannot be used in One-Disk-Mode.

Note: Except for floppy disks, you should usually leave this gadget at its defaults ('Off' in One-Disk-Mode, 'All Tracks' in Two-Disk-Mode).

There is currently one known limitation caused by the way ReOrg works internally: If your device uses blocksizes larger than 512 bytes AND the number of sectors per track is not an integer multiple of the number of sectors per block, you can only use 'Format=Off' for that

---

device. However, this should not be a problem, because block sizes larger than 512 bytes are currently only supported for harddisks, and harddisks should have 'Format' set to 'Off' anyway.

## 1.19 ReOrg\_English.guide/NODE\_G\_VERIFY

Gadget "Write Verify"

=====

If this gadget is selected, ReOrg will verify the output data after it has been written.

Notes:

- \* For floppy disks or similar devices, it is recommended to switch "Verify" on. Otherwise you might not detect physical damages on your destination disk. ReOrg uses an asynchronous-style verify that works rather fast.
- \* DO NOT USE VERIFY FOR HARDDISKS !!!

Always make sure that "Verify" is switched off for harddisks - for two reasons:

- \* Some harddisk controllers report incorrect error codes if "Verify" is switched on. This may confuse ReOrg and even destroy data on your harddisk.
- \* Verify really does not make any sense for harddisks. Harddisks usually have an on-disk buffer, so verify requests will simply read the data from that buffer and therefore can not check if the data has been written properly anyway.

## 1.20 ReOrg\_English.guide/NODE\_G\_WBENCH

Gadget "Workbench Mode"

=====

If this switch is activated, ReOrg will store icons (`.info'-files) in such a way that Workbench can access them very quickly, so icon images appear on Workbench very fast. This option has been improved for ReOrg 3.1. If the gadget "Optimize for" is set to "Kick2.0/3.0", ReOrg 3.1 causes Workbench to display icons even faster than previous ReOrg versions did.

Other hints if you want to make icons appear on Workbench even faster:

- \* "Snapshot" all icons ! This will avoid the delay that Workbench needs to calculate a suitable icon position.

- \* Use small images with a small number of colors for your own icons !  
They will be displayed faster than large, complicated icons.

## 1.21 ReOrg\_English.guide/NODE\_G\_GRAPH

Gadget "Graphical sector display"

=====

If this switch is activated, ReOrg will create a graphical display of the physical disk layout during optimization, but this may slow down the optimization process. However the speed of the optimization will benefit from a fast processor (68030/040) and from the improved graphics routines in Kickstart 3.0 (compared to Kickstart 2.0).

## 1.22 ReOrg\_English.guide/NODE\_G\_OPT

Gadget "Option file"

=====

This gadget works in combination with the "Select option file" and "Clear option file" gadgets. It lets you specify the name of an option file. Option files can be used to selectively change some of ReOrg's settings for individual files on your disk. Thus option files give you more control over the exact disk layout that ReOrg creates during the optimization. See

Option Files

for details about the usage and format of option files.

## 1.23 ReOrg\_English.guide/NODE\_G\_SOPT

Gadget "Select option file"

=====

If you click on this gadget, ReOrg displays a file requester that lets you specify the name of an option file. See

Option Files

for

details about the usage and format of option files.

## 1.24 ReOrg\_English.guide/NODE\_G\_CLOPT

Gadget "Clear option file"

=====

This option clears the contents of the "Option File" gadget. If no option file has been specified when you start the optimization, ReOrg will use default values. See  
Option Files  
for details about the usage  
and format of option files.

## 1.25 ReOrg\_English.guide/NODE\_G\_DRIVE

Gadget "Drive"

=====

In this gadget you select the drive you want ReOrg to optimize in One-Disk-Mode. There does not have to be a disk in the drive when you start the optimization. ReOrg will ask you to insert a disk. If you want to use different source and destination drives, click on the "Mode" gadget to toggle between One-Disk-Mode and Two-Disk-Mode.

## 1.26 ReOrg\_English.guide/NODE\_G\_FROM

Gadget "From drive"

=====

In this gadget you select the source drive you want ReOrg to use for optimization in Two-Disk-Mode. There does not have to be a disk in the drive when you start the optimization. ReOrg will ask you to insert a disk. Also see

To drive

.

## 1.27 ReOrg\_English.guide/NODE\_G\_TO

Gadget "To drive"

=====

In this gadget you select the destination drive you want ReOrg to use for optimization in Two-Disk-Mode. There does not have to be a disk in the drive when you start the optimization. ReOrg will ask you to insert a disk. The destination drive must be different from the source

---

drive in Two-Disk-Mode. If you want to use the same drive for source and destination, click on the "Mode" gadget to toggle between One-Disk-Mode and Two-Disk-Mode.

## 1.28 ReOrg\_English.guide/NODE\_G\_MODE

Gadget "Mode"  
=====

This gadget toggles between One-Disk-Mode ("One Drive") and Two-Disk-Mode ("Two Drives").

In One-Disk-Mode, the optimization is done "in place", i.e. there is only one volume involved in the optimization. Only use One-Disk-Mode if you have made a backup copy of your data before the optimization !

In Two-Disk-Mode, ReOrg copies your source disk to the destination disk and at the same time optimizes your data. Therefore, two compatible drives are needed. Your source disk is not altered by ReOrg in any way. Whenever possible you should use Two-Disk-Mode, because it is safer than One-Disk-Mode.

## 1.29 ReOrg\_English.guide/NODE\_G\_START

Gadget "Start"  
=====

This gadget actually starts the optimization. ReOrg checks if you selected valid drive(s) and prompts you to insert the disk(s). After that the source disk is scanned and - if no errors occur - another requester appears asking your permission to start the actual optimization.

## 1.30 ReOrg\_English.guide/NODE\_G\_CANCEL

Gadget "Cancel"  
=====

Selecting this gadget leaves the program.

## 1.31 ReOrg\_English.guide/NODE\_G\_DATE

---



Gadget "Update disk date"

=====

If this gadget is activated, ReOrg will change the disk creation date after the optimization to inform AmigaDOS that the disk layout has been changed.

THIS GADGET SHOULD USUALLY BE SWITCHED ON !

Note: Usually you should not switch this gadget off, because there is hardly any need for the user to control the disk creation date. But even in the unlikely case that you do not want the disk creation date to change, ReOrg 3.1 is more robust than previous ReOrg versions. If ReOrg detects that AmigaDOS might crash if you leave this gadget switched off, ReOrg will not allow you to start the optimization. This usually happens if there are still active "locks" on your disk. In this case, either switch this gadget back on or leave ReOrg and reboot your computer BEFORE starting the optimization.

## 1.32 ReOrg\_English.guide/NODE\_G\_LOCKS

Gadget "Keep Locks"

=====

If this gadget is switched on, ReOrg moves all open references (like "assigns", "file locks" etc.) from the unoptimized disk to the optimized disk after the optimization has finished. This means that you usually do not have to reboot your computer after the optimization, even if you optimized your boot partition. However, this does not include open files. NEVER try to optimize a partition that has open files !

The implementation of this function accesses some "private data structures" of AmigaDOS and might therefore not be compatible to all future AmigaDOS versions. Thus, for compatibility reasons, this gadget is switched off by default. However for all current AmigaDOS, FileSystem, FastFileSystem and DirectoryCacheFileSystem versions (V36 to V40), it is safe to switch this gadget on.

## 1.33 ReOrg\_English.guide/NODE\_G\_SIMUL

Gadget "Simulate optimization only"

=====

If this gadget is switched on, all optimizations are only simulated, i.e. ReOrg does not change your disk in any way. This can be useful if you try ReOrg on your system for the first time, or whenever you change your system configuration.

---

However, even if the simulation succeeds, this is no guarantee that the actual optimization will work correctly. There still might be some problems that only show up when ReOrg actually has to write data to disk. Therefore - even if the simulation succeeded - always backup your disk before starting an optimization in One-Disk-Mode !

Note: For disks that use the new directory-cache format, not all parts of the reorganization can be simulated. This is normal, so do not be concerned if ReOrg tells you after the simulation that it could not simulate some parts of the optimization process.

## 1.34 ReOrg\_English.guide/NODE\_G\_GEO

Gadget "Read drive geometry"

=====

For all currently available disk drives, this gadget should be switched OFF !

Caution: If you switch this gadget on for devices that do not support this function, your data may get damaged - even in simulation mode !

This function is used for devices that support more than one disk format. Currently the only such device is 'trackdisk.device'. However, you do not need to switch this gadget on for 'trackdisk.device', because ReOrg automatically recognizes 'trackdisk.device' and supports double density and high density disks even if this gadget is switched off.

The only drive type you might need this gadget for is some new SCSI drive for removable disks that supports more than one disk format.

If you think you might need this function on your system, please read

Technical Details

first ! That chapter describes what exactly "Read drive geometry" does and when it has to be used.

## 1.35 ReOrg\_English.guide/NODE\_G\_CLR

Gadget "Clear unused blocks"

=====

Usually ReOrg only reads and writes blocks that contain data, so the contents of unused blocks is undefined after the optimization. For normal use, this is no problem. However some disk recovery tools, like "UnDelete"-type or "DiskSalv"-type programs may get confused by the garbage data that is left in unused blocks. Therefore, if you often use such tools, it is advisable to switch "Clear unused blocks" on. This

slows down the optimization, but causes ReOrg to clear all unused blocks to prevent problems with disk recovery tools.

## 1.36 ReOrg\_English.guide/NODE\_G\_FEXT

Gadget "FileExt blocks"

=====

This gadget controls the placement of "File Extension Blocks", sometimes also called "File List Blocks". These blocks are used by AmigaDOS if the size of a file exceeds a certain limit (34.3 kB for floppy disks). The way AmigaDOS treats File Extension Blocks has changed between Kickstart 1.3 and Kickstart 2.0. For a complete description of this gadget, please see

Technical Details

. Usually the

best setting is:

- \* "Mid" if you use Kickstart 2.0/3.0 and use large buffers for your harddisk.
- \* "Front" if you use Kickstart 1.2/1.3.
- \* "Scatter" if you use Kickstart 2.0/3.0 and are optimizing a boot disk from which large program files are loaded.

For Kickstart 1.2/1.3, ALWAYS use "Front" or directories will be displayed VERY slowly.

For Kickstart 2.0/3.0, one of "Mid" or "Scatter" might give best results. Usually "Mid" is better for files that are accessed randomly, not sequentially (e.g. files that emulate an MS-DOS harddisk in an Amiga file). "Scatter" is better for files that are accessed sequentially with very small buffers. A lot of programs written in "C" only use buffers of 488 or 512 bytes size. For files that are accessed by those programs, "Scatter" gives best results.

You can change the setting of the "FileExt blocks" functions for individual files by using option files.

## 1.37 ReOrg\_English.guide/NODE\_G\_FREEB

Gadget "Free blocks"

=====

This gadget controls the position of the free blocks pool after the optimization. There are three possible settings:

'after files'

causes the directory area and the file area to be stored consecutively. The free blocks are then placed after the directory

area.

`after directories'

causes the free blocks to be placed between directories and files.

`after bootblocks'

causes the free blocks to be placed immediately after the bootblocks, possibly in the middle of the file area. This setting requires the directory area to be no larger than half the size of your disk.

The settings "after bootblocks" should only be used for floppy disks, and only if your disk contains a bootblock loader, bootblock picture etc. that uses more space than just the bootblocks.

In all other cases "after directories" or "after files" should be used. Which of those settings gives better results very much depends on your exact disk layout, the number of files and the percentage of used blocks. There is no easy answer which setting is better, but usually the difference is only minor. For a more detailed description of this gadget, please see

Technical Details

.

## 1.38 ReOrg\_English.guide/NODE\_G\_1320

Gadget "Optimize for (Kick1.3/2.0)"

=====

This gadget lets you select under which Kickstart version you want to use the optimized disk. "Kick1.3" optimizes the disk in Kickstart 1.3 fashion. This is also the technique ReOrg versions 1.x and 2.x used.

The setting "Kick2.0" is new. If you choose this setting, ReOrg takes advantage of the improved directory scan algorithm in the new Kickstart 2.0 filesystem and sorts the files in each directories in a way that - for example - icons are displayed on Workbench even faster.

Disks optimized with "Kick2.0" are still readable under Kickstart 1.3, but the directory display will be slower than with the old "Kick1.3" setting.

## 1.39 ReOrg\_English.guide/NODE\_G\_CHFS

Gadget "Change FileSystem"

=====

In addition to optimizing a disk, ReOrg can also be used to convert the filesystem of a disk. This gadget lets you convert the filesystem

to "Std Mode" (DOS\0 and DOS\1), "Intl Mode" (DOS\2 and DOS\3) or "DCFS Mode" (DOS\4 and DOS\5). "Intl Mode" disks require Kickstart 2.04 and Workbench 2.1 or higher. "DCFS Mode" disks require Kickstart 3.0 and Workbench 3.0.

This gadget only affects the filesystem 'generation'. The 'OldFileSystem vs. FastFileSystem' setting is not changed by the conversion. Use the gadget "Change OFS/FFS" to convert a disk between OldFileSystem and FastFileSystem.

Some restrictions apply. Please see  
FileSystem conversion  
for  
details.

## 1.40 ReOrg\_English.guide/NODE\_G\_CHOFS

Gadget "Change OFS/FFS"

=====

In addition to optimizing a disk, ReOrg can also be used to convert the filesystem of a disk. This gadget lets you convert the filesystem of a disk between OldFileSystem (DOS\0, DOS\2, DOS\4) and FastFileSystem (DOS\1, DOS\3, DOS\5).

This conversion only affects the "OldFileSystem vs. FastFileSystem" setting, not the filesystem 'generation', like Intl FS or Directory Caching. If you want to change those aspects of the filesystem, use the "Change FileSystem" gadget.

Some restrictions apply. Please see  
FileSystem conversion  
for  
details.

## 1.41 ReOrg\_English.guide/NODE\_G\_MEM

Gadget "Safety Memory"

=====

In this gadget you can specify the amount of memory ReOrg should NOT use as cache. Default is 150000 bytes, but you can reduce this value to 50000.

ReOrg will try to leave at least this much memory behind during the optimization. If you want to run other programs concurrently with ReOrg, you might want to increase the "Safety Memory" value.

Values smaller than 50000 bytes are not allowed to avoid low-memory situations while ReOrg is running.

## 1.42 ReOrg\_English.guide/NODE\_G\_LIMIT

Gadget "Small file limit"  
=====

ReOrg stores large files in a separate area on your disk to speed up the access to your directory data. However, small files are usually stored within the directory area to speed up access to those files. This does not affect the speed of directory accesses very much, but speeds up accesses to small files considerably.

In this gadget you can specify the maximum size for files that should be stored in the directory area. The default is 2 blocks. Large values slow down the directory access ('dir', 'list'); small values slow down the access to small files.

When you want to have the fastest possible directories and do not care about access time to slow files, you might want to set this value to 0. On the other hand, when you optimize boot disks (e.g. for games, where you usually do not list the directory), larger values can be useful.

## 1.43 ReOrg\_English.guide/NODE\_ONLINEHELP

Online-Help  
\*\*\*\*\*

ReOrg has a built-in online help feature. When you press the Help key, ReOrg shows you a context-sensitive help page using 'amigaguide.library'. You must have 'amigaguide.library' installed to use online help. If you install ReOrg using 'Installer', 'amigaguide.library' will automatically be installed on your system, unless you already have a more recent version.

When you press Help, ReOrg shows you information about the gadget under the mouse pointer. If your mouse pointer is not currently over a gadget, ReOrg shows you the main help page.

It is also possible to receive help about menu items: Pop up a menu by pressing the right mouse button, move the mouse pointer over a menu item and (with the right mouse button still held down) press Help.

Once online help is displayed, you can navigate through the whole help file as with any AmigaGuide document. Please see the AmigaGuide documentation for details.

Online help is available in several different languages. ReOrg

---

chooses the language for online help in the following fashion:

- \* First ReOrg tries to open the help file in the same language in which ReOrg is running.
- \* If the help file for this language is not available, and you are running under Workbench 2.1 or higher, ReOrg will try to use help files in your "preferred languages" as specified in 'Locale' preferences.
- \* If ReOrg still does not succeed (or if you are running under Workbench 2.04), ReOrg falls back to the default English help file.

## 1.44 ReOrg\_English.guide/NODE\_OPTIMIZATION

Optimization

\*\*\*\*\*

When you start the optimization, ReOrg first inhibits the drive(s) (i.e. it prevents AmigaDOS from accessing them) and prompts you to insert a disk. After that, a status window appears and ReOrg starts reading and processing the disk directory.

During this phase ("scanning disk"), ReOrg may be interrupted at any time even in One-Disk-Mode. At the end of this phase, ReOrg asks the user if this disk should really be optimized. In One-Disk-Mode, ReOrg MUST NOT BE INTERRUPTED OR CANCELLED after this message or your disk will be corrupt.

During the actual optimization, ReOrg shows how much time the optimization will approximately need.

After this phase, other (usually short) phases are used to copy bootblocks, write the bitmap etc.

When the optimization is completely finished, ReOrg asks you to remove the disk(s) and ReOrg shows you some statistical information about your disk and the optimization process.

During the optimization, the status window shows the following information:

- \* The number of directories and files which ReOrg has already processed ("done") and which ReOrg has queued for later processing ("queued").
- \* The current cache usage. If this value ever reaches 100% during the optimization, this indicates that the optimization would be faster if you had more memory in your system.
- \* The cache size (in absolute numbers, and relative to the disk size)
- \* The current status. One of:  
'checking disk'  
Before the actual optimization starts, ReOrg checks your disk

if it is a valid AmigaDOS disk and if the disk format is understood by ReOrg.

``scanning disk'`

ReOrg is reading the directory and is calculating the new disk layout for the destination disk.

``awaiting user response'`

ReOrg is waiting for you to confirm the start of the optimization.

``preparing optimization'`

ReOrg is translating cached sectors to the output disk format and creating several internal tables needed for the optimization.

``moving blocks'`

ReOrg is optimizing your disk. This phase usually takes most of the time and is the most "critical", i.e. it must not be interrupted !

``writing bitmap'`

ReOrg is writing the new bitmap back to your disk.

``formatting empty tracks'`

This phase only occurs if you set "Format" to "All tracks". ReOrg is formatting all unused tracks of your destination disk.

``clearing unused blocks'`

This phase only occurs if you set "Format" to "On" or "Off" and selected "Clear unused blocks". ReOrg is clearing all unused blocks of your destination disk.

``copying reserved blocks'`

This phase only occurs in Two-Disk-Mode. ReOrg is copying bootblocks to the destination disk.

``adjusting directory cache'`

This phase only occurs if the destination disk uses directory caching (DCFS). ReOrg is updating the directory cache information.

``converting filesystem'`

This phase only occurs if ReOrg has to convert the disk format between OFS and FFS. ReOrg is converting data blocks between OFS and FFS format.

``finished'`

ReOrg has finished the optimization.

\* The remaining time needed for the optimization. This value is only an approximation of the remaining time needed. If source and destination devices have a very different speed (e.g. "RAD" and "DF0"), the estimation is very inaccurate.

\* A gauge that shows how many percent of disk blocks have already

---



been moved. This percentage also corresponds to the percentage of time needed for the optimization.

- \* If the user selected the gadget "Graphical sector display", ReOrg shows a graphical display that tells which sectors are currently processed.

## 1.45 ReOrg\_English.guide/NODE\_FSCONVERSION

FileSystem conversion

\*\*\*\*\*

As of Kickstart 3.0, there are six different filesystems used on the Amiga:

'DOS\0'

The old filesystem that is compatible to all Kickstart versions.

'DOS\1'

The original FastFileSystem. This used to be a disk-based FileSystem under Kickstart 1.3 and has been in ROM since Kickstart 2.04.

'DOS\2'

The international version of the old filesystem. This filesystem fixes a bug in DOS\0 that caused filenames with international characters in them (like 'Ärger') to be case-sensitive, i.e. 'Ärger' and 'ärger' used to be two different file names. In this new filesystem version they refer to the same file.

'DOS\3'

The international version of the FastFileSystem. Both DOS\2 and DOS\3 have been in ROM since Kickstart 2.04 (NOT Kickstart 2.0 !).

'DOS\4'

The Directory-Cache version of the international old filesystem. This filesystem (as well as DOS\5) uses a new block type (directory cache block) to speed up directory accesses.

'DOS\5'

The Directory-Cache version of the international FastFileSystem. Both DOS\4 and DOS\5 have been in ROM since Kickstart 3.0. Caution: Some early beta-versions of Kickstart 3.0 had incompatible and/or buggy versions of this filesystems, so be sure to use only Kickstart 3.0 versions that have been officially released by Commodore !

As can be seen from this list, each filesystem (DOS\0 - DOS\5) has two properties:

- \* OldFileSystem or FastFileSystem.
- \* The "generation", i.e. Standard FS, International-FS or Directory-Cache-(International)-FS.

ReOrg lets you convert any filesystem to any other filesystem during

the optimization. There is one gadget ("Convert OFS/FFS") to toggle between OldFileSystem and FastFileSystem, and one gadget ("Convert FileSystem") to change the "generation" of the filesystem.

There are some restrictions on filesystem conversions caused by the filesystem itself:

- \* Conversions from "Std Mode" to "Intl Mode" or "DCFS Mode" will fail if there are two files in the same directory with international characters in the filenames and if those filenames only differ by case, e.g. files 'Ärger' and 'ärger'. Those two filenames are considered different under "Std Mode", but identical under "Intl Mode" and "DCFS Mode". If two such files exist on your harddisk, ReOrg will not let you convert the filesystem.
- \* Conversions from "Std Mode" or "Intl Mode" to "DCFS Mode" require some additional space on your disk for directory cache blocks. If your disk does not have enough space available, ReOrg will not let you convert the filesystem and issue a "Disk full" error message.
- \* The OldFileSystem only supports 26 bitmap blocks, which usually means that an OldFileSystem disk cannot be larger than 49.18 MB. However for block sizes other than 512 bytes, this number may vary. ReOrg does not let you convert a FastFileSystem partition to OldFileSystem if it is too large.
- \* FastFileSystem disks can store more file data than OldFileSystem disks. If you want to convert a FastFileSystem disk that is nearly full to OldFileSystem, the data might not fit on your disk. In this case, ReOrg issues an error message "Disk full".

One restriction is caused by ReOrg: If you want to convert a large OFS partition to FFS, you need a very large cache, because ReOrg has to store a lot of sectors in temporary memory during the conversion. If you do not have enough memory, ReOrg will not let you convert the partition from OFS to FFS.

Please make sure that you only convert to filesystems that your Kickstart version supports. In particular, you can usually not use the Directory-Cache FileSystem under Kickstart 2.x, although ReOrg lets you convert partitions to and from DCFS, even when running under Kickstart 2.x.

Here is a brief description how ReOrg converts the filesystem:

- \* Conversions that affect the "generation" only are rather easy and can be done "on the fly", i.e. they do not take considerably longer than normal optimizations. ReOrg only has to re-hash filenames and check for name clashes (as described above).
- \* Conversions between OFS and FFS are much more tricky, because ReOrg actually has to redistribute all file data among data blocks, which also means that the number of data blocks and/or FileList blocks for each file may change. To do this, ReOrg completely recalculates all control blocks used by the filesystem. Therefore this type of optimization is not done "on the fly". Instead, ReOrg uses another pass ("converting filesystem") after the actual optimization to redistribute the data. ReOrg tries to cache the whole disk in memory (if there is enough memory

available) between the optimization pass and the conversion pass. Thus - even when converting the filesystem - ReOrg only requires one read access and one write access to floppy disks, so filesystem conversions for floppy disks should not take much longer than usual optimizations, if you have enough cache memory. The only drawback for floppy disks is that read and write accesses cannot overlap (like during standard optimizations), because two different passes are involved. This is different for harddisks, however. Harddisks can usually not be cached in RAM, so ReOrg has to use twice as many read and write accesses, which slows down the optimization considerably.

## 1.46 ReOrg\_English.guide/NODE\_OPTIONFILES

Option files

\*\*\*\*\*

Most users will probably not need option files. However, option files are useful to make some "final" optimizations, e.g. before disks are distributed. Especially when you optimize game boot disks, a "perfect" optimization using option files may save you some seconds when you boot from your disk.

Option files allow you to change the settings of the "FileExt blocks" option, to change the order of directory entries and to specify the placement of data blocks for individual files. These settings override the global settings ("FileExt blocks" gadget and "Small file limit" limit gadget) for those files.

Example: You want to optimize a game boot disk that contains a lot of files. You therefore used the setting "FileExt blocks=Scatter". The disadvantage of this setting is that relative accesses to large files become slower. If you know that your game often accesses a single large file in relative mode (let's say the name of this file is 'graphics/dungeon.gfx'), you should change the setting of "FileExt blocks" to "Front" or "Mid" for this single file only. These kinds of settings can be specified in option files.

Option files are ordinary ASCII text files. You can use any ASCII text editor to create and edit them (e.g. MEmacs or ED).

File Format:

1st line:

\$1

(A dollar sign ("\$\$") and the digit One ("1") in the first two columns of the line).

all other lines:

options:filename:comment

valid options are:

'I0'

Data blocks of this file are not placed near the header block (default).

``I1'`

Data blocks of this file are placed near the header block (file accesses get faster, directory accesses get slower).

``E0'`

The same as "FileExt blocks=Front" for a single file.

``E1'`

The same as "FileExt blocks=Scatter" for a single file.

``E2'`

The same as "FileExt blocks=Mid" for a single file.

``P#'`

Specify a relative priority for this file's directory entry. Priorities can be between -32768 and 32767. The higher the priority, the earlier this file appears in the directory. (This option is only respected if the "Optimize for" gadget has been set to "Kick2.0/3.0". In "Kick1.2/1.3" mode the directory order is controlled by the hash code of the file name only, which cannot be modified.) The default priority for all files is 0. If Workbench mode is used, all `'#?.info'` files have priority 20000. Example: By using option "P10000" a file will appear after all `'#?.info'` files, but before all other files in the directory.

You can combine several options on a single line (e.g.:  
E0I1P500:filename:comment).

The above example:

```
$1
E0:graphics/dungeon.gfx:any comment
```

Some more notes about the file format:

- \* The file name must be specified relative to the root of the disk, i.e. please do not write `'/tst/name.doc'`, `'sys:tst/name.doc'` or `'disk:tst/name.doc'`, but only `'tst/name.doc'`.
- \* Spaces and other special characters may occur in the file name. You need not (and must not) use quotation marks to surround the file name !
- \* Patterns (like `'*.txt'` or `'#?.txt'`) are not allowed !
- \* Comments are optional. If you do not write a comment, you can omit the trailing colon. This means, lines like `'E0:filename'` are allowed.

## 1.47 ReOrg\_English.guide/NODE\_COMPAT

Compatibility / Limitations

\*\*\*\*\*

---

ReOrg should be compatible to all "trackdisk"-like devices including all harddisk devices. Exception: Do not use ReOrg with any devices that make assumptions about the filesystem format (like dynamic ram disks).

ReOrg supports all currently existing native Amiga FileSystems (DOS\0, DOS\1, DOS\2, DOS\3, DOS\4, DOS\5).

Previous ReOrg versions were not compatible to CycleToMenu ((C) Federico Giannici). This problem has been fixed in ReOrg V3.1.

ReOrg inhibits your drive when it starts the optimization. This may cause problems with programs that patch the operating system and try to load something from harddisk at certain times (like animations, sound, catalog files etc.).

Example: A program that modifies system requesters to play a sound sample or show an animation whenever a requester appears.

Some of those programs apparently get confused when the required volume (usually the boot partition) is not mounted.

Problems like that have been reported for

- \* Some versions of "Upd", a sound daemon used in combination with "Arq" ((C) Martin J. Laubach).

- \* "ReqTools" ((C) Nico François) versions earlier than 2.1. Nico fixed this problem in ReqTools 2.1, so ReqTools 2.1 and later are compatible to ReOrg - thanks, Nico !

There is one potential problem area if ReOrg is used in combination with third-party virtual-memory software (theoretically, because of ReOrg's asynchronous device accesses, ReOrg may deadlock if it tries to load disk data into memory that has been paged out to disk). To avoid those kinds of problems, all memory allocated by ReOrg uses the MEMF\_PUBLIC flag so it does not use virtual memory. If your virtual memory software lets you enable virtual memory individually for each program, I strongly suggest you DISABLE virtual memory for ReOrg (or even better, remove your virtual memory software completely while running ReOrg - this will also increase the amount of cache ReOrg can use).

If you have BOTH an 020/030/040 (NOT EC020) Amiga with memory outside of the 24-Bit address range AND a 24-Bit-DMA harddisk controller, it is important that you set the "Mask" keyword in your mountlist or in your harddisk installation software correctly. "Mask" must be set in such a way that it does NOT include memory outside of the 24-Bit address range. Usually, '0xffffffff' or '0xfffffe' are good values. There may be leading zeros in front of the first 'f' (like '0x00fffffff').

## 1.48 ReOrg\_English.guide/NODE\_TECHDETAILS

Technical details

\*\*\*\*\*

---

## Gadget "FileExt blocks"

=====

Files larger than 34.3 kB (OFS) or 36 kB (FFS) (these numbers are different for block sizes <> 512 bytes) require "FileExt" blocks (also called "FileList" blocks). There are different strategies where to place FileExt blocks, and they affect directory performance and file performance in different ways.

A rule of thumb is:

- \* If you want to use a disk under Kickstart 1.2/1.3, use "Front". Otherwise
- \* For floppy disks that contain program files, use "Scatter", unless the floppy disk contains files that are accessed randomly. In this case, use "Mid".
- \* For harddisks use "Scatter" or "Mid". Usually "Mid" is better.

Explanation:

## 'Front'

All FileExt blocks are placed immediately behind the file header block. This is the default under Kickstart 1.2/1.3, because for some obscure reason, Kickstart 1.2/1.3 requires FileExt blocks not only for file accesses, but also for directory displays ! This means: If you set the "FileExt blocks" gadget to something other than "Front" and want to use the optimized disk under Kickstart 1.2/1.3, directory displays will get MUCH slower than they already are. This is also the reason why directory displays of disks that have been written under Kickstart 2.0/3.0 are much slower under Kickstart 1.2/1.3 than usual. Conclusion: If you want to use the disk under Kickstart 1.2/1.3, use "Front". Otherwise use one of the other settings.

## 'Mid'

All FileExt blocks are placed behind the first sequence of data blocks. This places FileExt blocks "out of the way" for directory displays (thus Kickstart 2.0/3.0 directory displays get faster), but all FileExt blocks for one file are still in the same location, so random file accesses ("Seeks") are quite fast. This setting is usually best when you run under Kickstart 2.0/3.0, with one exception: Large files, that are accessed sequentially using a very small number of program buffers should be optimized at setting "Scatter". This is especially true for slow media, like floppy disks. You can use option files to change this setting for individual files.

## 'Scatter'

FileExt blocks are scattered among data blocks. This means, the filesystem finds them "just when it needs them" for sequential data access. This avoids one annoying effect when starting large program files from your floppy disk: Often, your disk drive head behaves like this: read three or four cylinders, step across the disk back and forth, read three or four cylinders, step across the disk back and forth and so on. This "stepping across the disk back

and forth" is caused by the filesystem when it needs the next FileExt block. By setting "FileExt Blocks" to "Scatter", you can avoid this effect, so loading program files from slow media gets faster. Major drawback: If you choose this option for large files that are accessed randomly (like files that simulate an MS-DOS harddisk), those accesses get EXTREMELY slow, because the drive head has to walk all over the disk to find the correct FileExt block !

#### Gadget "Free blocks after"

=====

This gadget controls where on your disk the area of free blocks should be located. There are three possible settings:

##### 'directories'

causes the area of free blocks to be placed between the directory and the file data area.

##### 'files'

causes the area of free blocks to be placed after both directory and file data area.

##### 'bootblocks'

causes the area of free blocks to be placed immediately after the bootblocks. This setting should only be used for disks that have a custom boot-loader that requires more blocks than the two bootblocks reserved by the filesystem. It causes ReOrg not to overwrite a custom boot-loader. Do not use this setting in any other situation, because it may make some file accesses slower.

Usually, "directories" or "files" is the correct setting. For the current filesystems, there is not much difference between both settings. In most cases, "directories" may make the creation of new files slightly faster, especially for files that are created at a deep directory level, but the "files" setting may be better if you often create files or directories at the root directory level and still need a fast display of the root directory.

The setting also affects the access time to existing files. Usually, "directories" gives you faster access to existing files at a deep directory level, whereas "files" gives you faster access to files near the root directory.

#### Gadget "Read Drive Geometry"

=====

Starting with Kickstart 2.0, AmigaDOS supports devices that handle different physical disk formats. One such example is the new 'trackdisk.device' that can handle old 880kB disks and new 1.76MB disks (if you have installed a high density drive). To have the filesystem automatically switch between 880kB and 1.76MB, the following mechanism is used:

- \* When a new disk is inserted, 'trackdisk.device' asks 'disk.resource' to inquire the drive type. This is done in hardware by sending some signals through the disk drive interface

and waiting for the disk drive to reply. Depending on the type of disk inserted, the drive answers with "DD" or "HD".

- \* Now 'trackdisk.device' knows the correct disk type, but the filing system does not yet. As soon as the filing system finds out that the disk has been changed, it sends a "TD\_GETGEOMETRY" to the 'trackdisk.device' to inquire the drive type and change its internal variables.

This mechanism works fine as long as only 'trackdisk.device' is used and not any other third-party device. Unfortunately, most third-party devices do not support "TD\_GETGEOMETRY" yet. Even worse, some devices do not respond with the correct error code ("IOERR\_NOCMD"), but they do something completely different and dangerous, like changing SCSI bus settings or even hard-formatting your drive !

If the gadget "Read Drive Geometry" in ReOrg is switched on, ReOrg tries to inquire the drive geometry by sending "TD\_GETGEOMETRY" to the device, just like the filing system does for 'trackdisk.device'. This function may be useful for some future devices that support multiple disk formats (like ultra-high density drives etc.). However in the meantime until third-party controllers have been fixed to support this command, I strongly recommend you leave this gadget "Off". The only situation where this gadget is useful is if you have a device on your SCSI bus that supports different physical formats (like DD, HD, XD) AND if your SCSI controller already supports "TD\_GETGEOMETRY".

The internal function "Read Drive Geometry" is always set to "On" for 'trackdisk.device' even if the corresponding gadget is switched off. Thus there is no need for you to manually switch this gadget on for floppy disks.

ReOrg V3.1 uses a binary "brain file" that contains a list of devices which support (or do not support) commands like "TD\_GETGEOMETRY". Therefore if at some time in the future controllers and devices get available that support multiple disk formats and "TD\_GETGEOMETRY", you just need a new brain file and can continue to use ReOrg.

I would be thankful for any reliable information which devices support the "critical" commands "TD\_GETGEOMETRY", "CMD\_UPDATE" and "TD\_PROTSTATUS" correctly, so I can update the brain file.

#### Misc. Information

=====

- \* The "bitmap" of the source disk is not required by ReOrg, i.e. it is not necessary to "validate" your disk before optimizing it.
- \* Before ReOrg changes anything on your disk, it first checks the structure of your disk. This check has been improved since earlier ReOrg versions, so ReOrg now recognizes all possible errors except for physical or logical errors in file data blocks.

Checksum errors are not recognized and reported by ReOrg, but ReOrg automatically recalculates all checksums during optimization (and thereby corrects checksum errors).



If ReOrg reports an error in the "disk scan" phase, it has not destroyed or changed your disk yet, so there is no need for you to restore it from your backup. But when the "moving blocks" phase starts, ReOrg must not be interrupted any more, because during this phase the disk layout is inconsistent.

- \* Although ReOrg checks your disk for most errors, I recommend you thoroughly check your disk for ALL kinds of errors (particularly for unreadable sectors on floppy disks) before optimizing it. Otherwise ReOrg might report a physical error in the "moving blocks" phase and destroy some of your files.

If an error occurs...

=====

If ReOrg stops, aborts or is interrupted in the "moving blocks" or "converting filesystem" phase, the following things happen:

- \* Your disk is displayed on Workbench as 'DF0:REOR' ('DF0' being the drive that contains your disk).
- \* When you try to access your disk, AmigaDOS reports "Not a DOS disk".
- \* The contents of your disk are TOTALLY UNUSABLE and COMPLETELY DESTROYED.

In this situation there is only one thing you can do:

REFORMAT YOUR DISK ("FORMAT ... QUICK") AND RESTORE YOUR BACKUP !

Do not trusts "hints" from computer magazines, mailboxes etc. telling you you could still repair parts of your disk. I explicitly warn you:

Do not try to repair your disk using tools like "DiskSalv", "FixDisk", "DiskDoctor" or any commercial disk tools.

Those tools have not been made for this purpose. During the optimization ReOrg seemingly randomly swaps data blocks, and there is no way for such tools to figure out which blocks belong to which file, because the disk structure is not even remotely consistent.

Although you might by coincidence really restore some of your files, you will most certainly "restore" a lot of corrupt files, but you will not get an error message from your disk recovery tool. This is especially true for all FFS disk formats.

"Repairing" corrupt program files is one of the "easiest" ways to provoke software failures in the future, so please do not try it !

If ReOrg failed during the "moving blocks" or "converting filesystem" phase, please restore your backup !

## 1.49 ReOrg\_English.guide/NODE\_ERRORMESSAGES

Error messages

\*\*\*\*\*

`Error 1'

AmigaDOS error: The file size does not correspond to the number of data blocks of FileExtension blocks.

`Error 2'

Device error: I/O error during read

`Error 3'

Device error: I/O error during write

`Error 5'

Device error: I/O error trying to switch off the motor of the source drive

`Error 6'

Device error: I/O error trying to switch off the motor of the destination drive

`Error 7'

AmigaDOS error: Invalid block number in directory tree

`Error 8'

AmigaDOS error: A block is used twice (corresponds to Disk-Validator error message "Key ### is already set"). Meaning: One sector occurs in the block list for more than one file/directory.

`Error 9'

AmigaDOS error: A control block has an invalid primary ID code in the first longword (must be =2).

`Error 10'

AmigaDOS error: A control block has an invalid secondary ID code in the last longword (currently -4,-3,1,2,3,4 are recognized)

`Error 11'

File error: Cannot open option file

`Error 12'

File error: Cannot read option file

`Error 13'

File error: Option file has invalid format

`Error 14'

File error: Option file has an invalid version

`Error 15'

File error: Option file is too large (more than 65534 lines)

`Error 16'

---

---

File error: Unexpected end of line in option file

`Error 17'

File error: Invalid option in option file

`Error 18'

File error: Invalid combination of options in option file

`Error 19'

Run time error: Not enough memory

`Error 20'

No FROM-device selected

`Error 21'

No device selected

`Error 22'

No TO-device selected

`Error 23'

Same device selected as FROM and TO device

`Error 24'

FROM and TO device are incompatible

`Error 25'

Device error: Cannot open device

`Error 26'

Format error: Disk format unknown

`Error 27'

Format error: Disk does not have a valid AmigaDOS format

`Error 28'

File error: Error writing the settings file

`Error 29'

File error: Error reading the settings file

`Error 31'

Device error: I/O error during verify. If this error occurs, you can cancel the optimization, try again to write and read the blocks or ignore the error

`Error 32'

No disk inserted

`Error 34'

Disk is write protected

`Error 35'

Device error: Difference between read and written data during verify. If this error occurs, you can cancel the optimization, try again to write and read the blocks or ignore the error

---

- `Error 36'  
Runtime error: Not enough temporary memory. If this error occurs, please increase the amount of "Safety memory" and start the optimization again
- `Error 37'  
AmigaDOS error: The "parent" pointer in a control block is invalid
- `Error 38'  
AmigaDOS error: The "self" pointer in a control block is invalid
- `Error 39'  
Partition is too large for FFS to OFS conversion
- `Error 40'  
Runtime error: You do not have enough memory to convert the partition from OFS to FFS
- `Error 41'  
Runtime error: The directory area on your disk is too large for the setting "Free blocks after bootblocks". Either decrease the size of the directory area (by decreasing the "Small file limit") or set the "Free blocks" gadget to some other value.
- `Error 42'  
Runtime error: Could not open online help file
- `Error 43'  
Runtime error: 'amigaguide.library' not available.
- `Error 44'  
Device error: If you want to Format tracks on the output device, BlocksPerTrack must be divisible by SectorsPerTrack. ReOrg cannot format devices for which logical blocks span more than one track.
- `Error 45'  
You must switch "Update disk date" on if there are open locks on your disk !
- `Error 50'  
Disk is full !
- `Error 401'  
Device error: Device not found (this error should not occur)
- `Error 402'  
Device error: Device does not have one of the valid AmigaDOS formats (DOS\0, DOS\1, DOS\2, DOS\3, DOS\4, DOS\5)
- `Error 403'  
Device error: Block length invalid (must be 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768 bytes)
- `Error 404'  
Device error: SecOrg is not equal to zero
- `Error 407'
-

Device error: The number of reserved sectors is zero

`Error 408'  
Device error: The number of sectors per track is zero

`Error 409'  
Device error: ReOrg could not inhibit the drive

`Warning 502'  
Warning: Invalid control block found while moving blocks. A file or directory on the destination disk may be damaged

`Warning 503'  
Warning: Invalid data block found while moving blocks. A file on the destination disk may be damaged

`Warning 504'  
Warning: The Mountlist entry "MaxTrans" is too small to read a whole track. This warning usually only indicates that your controller has been configured improperly

`Warning 505'  
Device error: Error returned by write protection check

`Warning 506'  
Device error: CMD\_UPDATE after last write access returned an error

`Internal error ####'  
If an internal error occurs, please write down all numbers displayed in the error requester and contact the program author. Internal errors indicate that ReOrg's internal data structures somehow got destroyed, probably due to a bug in the program.

## 1.50 ReOrg\_English.guide/NODE\_UPDATES

Update information  
\*\*\*\*\*

I will try and further improve ReOrg if I have time for it. Registered users will at least get a notification when new versions of ReOrg get available that contain significant changes or enhancements.

New versions of ReOrg will also be available on the Internet per "anonymous FTP" (look on Aminet), on some other computer networks like "BIX" and - probably with some delay - on "Fish-Disks".

In order to be able to improve and/or correct ReOrg, I would like to ask every user to do the following:

- \* send me the registration fee of US\$ 10 or DM 15.
  - \* send me bug reports, if you find any bugs.
  - \* give me hints how to improve ReOrg.
-

Thank you very much in advance !

## 1.51 ReOrg\_English.guide/NODE\_HISTORY

### History

\*\*\*\*\*

`V3.1 (Rev 206, 30.08.1993, 6th release)'

- \* support for directory-caching (DCFS: DOS\4, DOS\5).
- \* support for blocksizes <> 512 bytes.
- \* implemented filesystem conversion during optimization.
- \* all windows and requesters are font-sensitive.
- \* added option "Keep Locks" that allows you to continue your work after optimizing the boot partition without having to reboot the system.
- \* improved directory optimization algorithm used under Kick2.0. Side effect: icons appear faster on Workbench.
- \* localization (even when running under Workbench 2.04).
- \* online help function using AmigaGuide.
- \* numerous bug fixes and enhancements.

`V2.33 (Rev 99, 30.09.1992, 5th release)'

- \* fixed a bug that could cause deadlocks during the "moving blocks" phase at "99% finished".

`V2.32 (Rev 94, 16.07.1992, 4th release)'

- \* ReOrg can now handle Kickstart-1.1 compatible Mountlist/RigidDiskBlock entries.

`V2.31 (Rev 88, 04.07.1992, 3rd release)'

- \* Fixed several minor bugs.

`V2.3 (Rev 75, 20.06.1992, 2nd release)'

- \* ReOrg now works correctly with cache sizes larger than 8 MB.
- \* implemented workarounds for controllers that do not implement asynchronous verify or get\_geometry correctly.
- \* several minor bug fixes and enhancements.

`V2.1 (Rev 49, 28.05.1992, 1st release)'

initial release.

---

## 1.52 ReOrg\_English.guide/NODE\_ACKNOWLEDGEMENTS

### Acknowledgements

\*\*\*\*\*

Thank you very much to

- \* all beta testers of ReOrg V3.1 and all previous versions. I am afraid there are too many beta testers to list them all here.
  - \* everybody who translated documentation and program texts to other languages (Danish: Robert Justesen, Dutch: Louis van Dompelaar).
  - \* James Cooper from SAS Institute for beta-testing and for his help with the excellent SAS/C compiler.
  - \* Randell Jesup from Commodore for answering lots of questions about new Amiga filesystem features.
  - \* all users who have already sent (or are planning to send) me their registration fee.
  - \* everybody who sent me bug reports, suggestions or criticism and thus helped to improve ReOrg.
  - \* all programmers who write public domain and shareware software for the Amiga.
-