

KingFisher

COLLABORATORS

	<i>TITLE :</i> KingFisher		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 KingFisher	1
1.1 KingFisher 1.40	1
1.2 1 INTRODUCTION	2
1.3 1.1 Legalities	3
1.4 1.2 Requirements	3
1.5 The ARP Library	4
1.6 1.3 Features	4
1.7 AmigaGuide®	6
1.8 Regular Expressions	6
1.9 Creating a Directory	7
1.10 Aquarium	7
1.11 The Workbench Tooltypes	7
1.12 Version Links	8
1.13 1.4 Save the Fish	8
1.14 1.5 Philosophy	9
1.15 definitions	10
1.16 2 INSTALLATION	10
1.17 2.1 Floppy Disk Installation	11
1.18 2.2 Hard Disk Installation	11
1.19 2.3 Upgrading from v1.0	12
1.20 2.4 Operation under a BBS	12
1.21 3 CONCEPTS AND MENUS	13
1.22 3.1 Concepts	14
1.23 Adding Fish	15
1.24 Boolean Expressions	15
1.25 Colors	17
1.26 Copying between Tanks	17
1.27 Cut & Paste	18
1.28 Deleting Fish	18
1.29 The Display Database	18

1.30	Exporting the Database	19
1.31	The Fish.index File	19
1.32	Fishtanks	19
1.33	Fonts	20
1.34	Iconifying KingFisher	20
1.35	Insert Volume Requesters	21
1.36	The KingFisher.config File	21
1.37	The KingFisher.prefs File	21
1.38	Modifying the database	22
1.39	The Name.index File	22
1.40	Output Commands	22
1.41	Printing and Exporting	23
1.42	Reindexing	24
1.43	Searching	24
1.44	Verify Output Cmd	25
1.45	Version Select Buttons	26
1.46	Function keys	26
1.47	3.2 Menus	27
1.48	3.2.1 The Fish Menu	27
1.49	3.2.2 The Options Menu	28
1.50	Add Fish	29
1.51	Delete Fish	29
1.52	Reindex	29
1.53	Output	30
1.54	Output, Fish	30
1.55	Output, Fish w/Disk#	31
1.56	Output, Disk	32
1.57	Output, Range	32
1.58	Output, Whole Database	32
1.59	Output, Filter (search)	33
1.60	Copy from	33
1.61	Use Settings	34
1.62	Save Settings	34
1.63	About	34
1.64	Help	34
1.65	Quit	35
1.66	User Interface	35
1.67	AutoActive Gadgets	36
1.68	Keyboard Control	36

1.69	Wide Search String	38
1.70	Keep Settings	38
1.71	Description Font	38
1.72	Titlebar	39
1.73	Confirm Quit	39
1.74	Output Format Options	39
1.75	Output Format, Width	39
1.76	Output Format, Output to	39
1.77	Output Format, Format	40
1.78	Alert Options	40
1.79	Alerts, Reindex done	40
1.80	Alerts, Search ok	40
1.81	Alerts, Search fails	40
1.82	Background Reindex	41
1.83	Verify Output Cmd	41
1.84	Safe Iconify	41
1.85	Search Setup	41
1.86	Search Setup, Case Sensitive	42
1.87	Search Setup, Simple Mode	42
1.88	Search Setup, Auto Mode Detection	42
1.89	menusearchallowdospatterns	42
1.90	Search Setup, Trim Blanks	44
1.91	Clear output file	44
1.92	4 THE DATABASE	44
1.93	4.1 Recovering the Database	44
1.94	4.2 Volume References	46
1.95	4.3 Technical Background	46
1.96	4.4 Updating your database	49
1.97	5 QUESTIONS & ANSWERS	49
1.98	6 KingFisher and ARexx	59
1.99	ARexx Example Program	61
1.100	ARexx: ADDFISH	62
1.101	ARexx: GET	63
1.102	ARexx: HELP	64
1.103	ARexx: INHIBIT	65
1.104	ARexx: MOVETO	65
1.105	ARexx: QUIT	67
1.106	ARexx: RESET	67
1.107	ARexx: SEARCH	68

1.108ARexx: SET	70
1.109ARexx: SHOW	71
1.110ARexx: TELL	72
1.111ARexx: VERSION	73
1.112ARexx: SCREEN	73
1.1137 MISCELLANEOUS	74
1.114The Author	74
1.1157.2 Development Environment	74
1.1167.3 Problem Report Form	75
1.1177.4 History	75
1.1187.5 The Future	78
1.1197.6 Credits	79
1.120Edit Search String	81
1.121Select Fishtank	81
1.122Select the Use of the Search Range	81
1.123Search activation and string selection	81
1.124Select a Specific Disk	82
1.125Select Previous or Next Disk	82
1.126Select earlier or later Version	82
1.127Select Previous/Next; Search backward/Forward	83
1.128Toggle Search Mode	83
1.129Modify Search Range of Disks	83
1.130Reveal additional text	84
1.131Reveal text line-by-line	84
1.132Iconify KingFisher	84
1.133Jump to first/last Fish	85

Chapter 1

KingFisher

1.1 KingFisher 1.40

KingFisher v1.40
20, 1993
Copyright © 1992-1993 Udo Schuermann
All rights reserved
Freely Distributable

December ↔
6000 42nd Avenue, Apt. 405
Hyattsville, Maryland 20781 (USA)
email: walrus@wam.umd.edu

"New way? What's this I hear about a new way?"
-- Alex, "A Clockwork Orange"

While the most important chapters are marked with an asterisk (*) you should read as much of KingFisher's documentation as you can. It's been written with a purpose, after all:

1	INTRODUCTION
	*
1.1	Legalities
1.2	Requirements
1.3	Features
	*
1.4	Save the Fish!
1.5	Philosophy
1.6	Definitions
2	INSTALLATION
2.1	Floppy Disk
2.2	Hard Disk
2.3	Upgrading from v1.0

2.4	Operation under a BBS
3	CONCEPTS AND MENUS
	*
3.1	Concepts
3.2	Menus
3.2.1	The Fish Menu
3.2.2	The Options Menu
4	THE DATABASE
	4.1 Recovering the database
4.2	Volume References
4.3	Technical Background
4.4	Updating your database
	*
5	QUESTIONS & ANSWERS
6	ARexx
7	MISCELLANEOUS
7.1	The Author
7.2	Development Environment
7.3	Problem Report Form
	*
7.4	History
7.5	The Future
7.6	Credits

1.2 1 INTRODUCTION

KingFisher is a database tool designed for the maintenance, ←
searching and
perusal of the contents descriptions of Fred Fish's AmigaLibDisks.
KingFisher requires Kickstart 1.3 or later, at least one floppy disk drive
(two drives or a harddisk are recommended) and has been designed for
compatibility with all Amiga models. Some features are available only when
operating with Kickstart 2.04 (V37) or later.

*

1.1	Legalities
1.2	Requirements
1.3	Features

- *
- 1.4 Save the Fish!
- 1.5 Philosophy
- 1.6 Definitions

1.3 1.1 Legalities

KingFisher may not be distributed for profit or be included with ↔ software or hardware that is sold for profit except with express written and signed permission from the author. The intent of this program is to provide a free service to the Amiga community and not offer a chance for anyone to enrich themselves.

Express permission is given to Fred Fish to distribute KingFisher as part of his library of freely distributable software. KingFisher may also be distributed via public service electronic systems which do not charge explicit fees to the end user for downloading (example: Usenet and public ftp sites.)

KingFisher must be distributed in its entirety. That includes especially this documentation, but all datafiles as well --- an exact listing of the files that comprise KingFisher is given below. Distributing KingFisher in any diminished, reduced, or incomplete format is against the express wishes and commands of the author. Any and all exceptions shall be the author's prerogative only.

Files that must be part of the distribution:

KingFisher The program.
 KingFisher.info The icon for the program.
 KingFisher.guide The documentation and help file.
 KingFisher.guide.info The icon for the documentation and help file.
 Fish-0001-0500 Descriptions for Fish Disks 1-500.
 Fish-0501-1100 Descriptions for Fish Disks 501-1100.
 Fish.index Main index for database.
 Name.index Name Index for database.
 KingFisher.floppy Floppy disk customized version of KingFisher.prefs.
 KingFisher.hd Hard disk customized version of KingFisher.prefs.

1.4 1.2 Requirements

KingFisher requires a little under 380K of free memory, which ↔ accounts for the program itself, graphics images, fonts, fish and name index (which allocate up to 25% extra space) and internal structures.

Chip RAM: 80K (varies depending on screen mode)
Other RAM: 300K

Also needed is at least one floppy drive, although two are highly recommended, and a harddisk would make your life much easier. Your Amiga should run Kickstart 1.3 or later, although it is possible that KingFisher runs under earlier versions of Kickstart, too. When operating under Kickstart 1.3 the presence of the
arp.library
version 39 is required.

KingFisher can operate with less memory if invoked with the CLI or TOOLTYPE option 'TINY' or the 'Tiny-RAM=yes' option in the .prefs file. This is useful especially for
Bulletin Board Systems
or installations that do not
ordinarily add many fish to the database at once.

1.5 The ARP Library

The ARP Library is one of the most popular and most widely distributed external code libraries in the Amiga's history. It provided the first widely available standard file requester, for example, and numerous other functionalities.

KingFisher uses the ARP Library (stored as the file LIBS:arp.library) only for its file requester, and only when operating under AmigaDOS before version 2.0

Under AmigaDOS 2.0, KingFisher uses the operating system's standard ASL Library for both file and font requesters.

1.6 1.3 Features

KingFisher offers the following features:

- o Maintains a
split database
across multiple disk volumes,
 - o Searches this database more than 3 times faster than
Aquarium
 - o A program name index offers extremely fast location of programs
by name,
 - o
Boolean expression
allow searches involving strings combined with
AND, OR, and NOT operators,
 - o KickStart 2+
regular expressions
available as part of boolean
-

- expressions,
- o Case sensitivity available in all search modes,
- o Highlights keywords upon successful search,
- o Supports Workbench tooltypes,
- o Keyboard shortcuts to GUI functions for moving around in the database,
- o AmigaGuide® Documentation and Help,
- o ARexx port with 13 commands (numerous options),
- o The 10 function keys can now each invoke shell scripts or CLI programs for a great variety of added flexibility and power.
- o Adds new fish directly from Contents files or Usenet postings, even from multiple concatenated postings or email contents files,
- o Fish can be deleted from the end of the database to undo errors,
- o Database is a line-oriented text file accessed through index, that can be modified by hand and then reindexed,
- o Flexible print and export functions with search filter,
- o Select from multiple different search strings,
- o Limit searches to a portion of the database,
- o Follow version links to older or newer versions of software,
- o Maintain multiple different ``bookmarks`` into the database,
- o Supports proportional and scalable fonts for description text,
- o Uses the display database under 2.0+ and has limited support for overscan, PAL, and interlace resolutions under 1.3
- o String gadgets optionally kept active for keyboard comfort,
- o Full configuration saved between executions,
- o Iconifies to save Chip RAM (46K).

I would like to stress that I have made every effort to assure that KingFisher is bug free and compatible with many configurations, both software and hardware. I provide this program as a service to the Amiga community without any expectations of compensation for my efforts (although I won't reject any offers :)

If you have suggestions, questions, or you find a problem with KingFisher, I welcome your feedback either via email or standard (snail) mail:

Udo Schuermann
6000 42nd Avenue, Apt. 405
Hyattsville, Maryland 20781 (USA)

email: walrus@wam.umd.edu

1.7 AmigaGuide®

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

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.

AmigaGuide and related software is the Amiga's new hypertext documentation and standard help system made available by Commodore-Amiga for users of the system software starting with V34 (KS1.3)

KingFisher will always attempt to obtain access to the KingFisher.guide file unless the option 'Help-file' is set to a null value (i.e. "Help-file=")

1.8 Regular Expressions

KingFisher supports the use of Regular Expressions only under ↔
AmigaDOS 2.0

or later, and only if these regular expressions are part of a quoted(!) string. The following example contains a regular expression to search for any software of which 'Fred Fish' was part of the author(s), as opposed to being mentioned by name in the description (this is a capability that goes one step beyond the ordinary

boolean expressions
that KingFisher now

also supports):

```
"#?author:#?fred fish#?"
```

Notice that the expression is enclosed in quotes. Notice also that the expression begins and ends with a #? wildcard. This is because the expression may occur anywhere within the description. If you had left off the first #? the expression would have been anchored to the beginning of the description (actually to the beginning of the program title, which is then separated from the description by a single character, the newline symbol.

1.9 Creating a Directory

If you are a Workbench user, creating a new directory is also called ``creating a new Drawer.``

Workbench: The SYSTEM drawer contains a drawer named "Empty."
 Select this drawer and while it is selected, select from the Workbench menu the command "Copy" (or "Duplicate".) This will create a drawer named "Copy of Empty." Rename this drawer to what you wish it to be called, and then drag and drop it where you want it to exist.

Example: rename it to KF and drag and drop it into the root directory of your harddisk.

CLI: Change your current directory to the root directory of your harddisk, and issue the CLI command "mkdir KF" -- this creates a subdirectory named KF. Be aware that no icon is created for this directory, just in case that you expect to find this directory from the Workbench, too.

1.10 Aquarium

B. Lennart Olsson created the first widely distributed tool for storing and searching the contents of Fred Fish's disk library. Compared to KingFisher, Aquarium was somewhat primitive, yet it served the Amiga community admirably for a number of years.

Two of Aquarium's short-comings were responsible for the creation of KingFisher: a somewhat arbitrary and inflexible set of flags were the only way to search for software other than by a single keyword (setting these flags was a painful process), and the database outgrew in size the capacity of a single floppy disk and could not span multiple disks.

1.11 The Workbench Tooltypes

Tooltypes are parameters that are part of TOOL icons. AmigaDOS ←
 defines a
 standard set of such parameters (tooltypes) to create a uniform interface for programs that use the Workbench interface. You can modify a TOOL's parameters (tooltypes) by selecting the icon and issuing the Workbench menu command INFO.

KingFisher offers the use of the standard SETTINGS tooltype, which specifies the exact name of KingFisher's configuration file. This file should be named

```
KingFisher.prefs
; older versions of KingFisher accepted
```

only the name KingFisher.config.

If the SETTINGS tooltype is given, but it specifies no file, i.e. you provide an entry such as

SETTINGS=

then KingFisher will make NO attempt to load a configuration from a file, but instead use a built-in default configuration. You can determine what this default configuration is by saving a configuration file that did not exist when you started KingFisher.

Any configuration can be augmented/modified by specifying the necessary settings as tooltypes, just as they appear in the preferences/config file. This functionality is available regardless if a configuration is loaded from disk or not, and will always override the settings in the file.

1.12 Version Links

During an
Add Fish
or
Reindex
operation, KingFisher builds a list of
previous and following versions of software to allow you to quickly and easily skip forward and backward from one version of a particular program to another.

This is especially useful to compare the significance of progress that an application has made over the years, or perhaps how frequently it has been revised over time.

You should be aware that KingFisher is far from perfect when it comes to building these links. It relies heavily on the fact that nearly every revised entry in the database states something to this effect:

"This is version 19.93, and upgrade to version 19.92 on disk 91."

It understands that some programs use the word "update" rather than "upgrade." It is true that computers are rather daft when it comes to understanding things that humans take almost for granted. I don't think it a good idea, however, to program exceptions to the majority into KingFisher just to catch the 10% whose wording does not follow the hoped-for pattern.

1.13 1.4 Save the Fish

Fred Fish has recently announced that further maintenance of his famous library is not worth his time because he lacks permanent subscribers. At the time of this writing, Fred had only 41 permanent subscribers left in the entire world, which barely paid for his time and efforts.

This is a rather blatant plea to all user groups, other organizations, and perhaps to somewhat more affluent individuals, too, to support Fred Fish by ordering subscriptions to his library! Fred is considering to lower the per-disk price of large-volume subscriptions to his library from \$3 US to \$2 US per disk to make it easier for small organizations or individuals to

buy the next 100 disks or so.

While there is a grass-roots movement underway to encourage Fred to maintain his library by providing donations, Fred needs permanent subscribers to assure that the Fish Library has a future. Donations are generally a one-time injection of life-support; I'm sure Fred appreciates the support he is getting in so many ways, but it is actually becoming a trade-off between what his professional time is worth and the dwindling income from subscribers that pays for his time.

Please save the Fish! Support Fred with long-term subscriptions! Do it not for Fred or for me. Do it for yourself most of all: count the programs you have retrieved from ftp sites, bulletin boards, and other sources, and ask yourself this: ``Who else but Fred Fish can I depend on to produce a collection of software so rich and reliable as Fred's Fish Disks?''

What if the Fish Library ends where it is right now? Don't you look forward to the next disks that Fred releases? Don't you read the listings with relish, learning what cool new software is now available from such a reliable source? What if that came to an end? Would you not miss it?

Save the Fish!

Tomorrow may be too late. Do it now!

(This has been a public service announcement for your own benefit!)

1.14 1.5 Philosophy

In writing KingFisher, I examined my own needs rather than ←
 duplicating the
 functionality of

Aquarium

. I literally never used the key flags, finding them too limited to adequately describe the variety of programs in the library: Nearly always did I search for a substring in the descriptions. KingFisher does that job over three times faster than Aquarium. On my (harddisk) system KingFisher can search (unsuccessfully) the entire database of 800 disks in under 35 seconds.

Adding new fish to the Aquarium database usually meant setting the key flags. Those of you who have done this in the past know how dreadfully time consuming a task this is. Largely due to this fact is why my updates to the database were so often late. I'm glad that I let Kyle Saunders talk me into not implementing the key flags in KingFisher. KingFisher will never have the sort of keyflags that Aquarium had.

Many people add their own entries to the database. To make this process as painless as possible, KingFisher includes a fairly clever parser that can skip over embedded message headers such as are commonly generated by news and email software. Should a contents file fool the parser, you can simply remove the problematic entries or as many disks as necessary from the database, modify the contents file to be ``clean'' and then re-add it.

If your index file should get corrupted or lost, or you decide to make

spelling corrections or other such changes to the database files themselves, then KingFisher can easily rebuild the index. This means that KingFisher's database is far more robust and manageable than Aquarium's.

1.15 definitions

This section defines some terms that are used somewhat loosely in this document:

KS 2.0 Kickstart 2.0 features are available starting with KS 2.04, which is more accurately described as V37 of the operating system. 2.0 through 2.03 is labeled V36.

Config file Refers either to the KingFisher.prefs or KingFisher.config files. Ideally these should be stored in the current directory, or ENV:KingFisher/ and ENVARC:KingFisher/

Freely Distributable Software of this nature may be distributed free of charge but the author often retains a variety of rights on the software. As such, the software may continue to belong to the author and may not be used by others against the wishes of the author.

Public Domain Software of this nature defines a superset of freely distributable software; the author has, however, given up all rights to the software, making it public property. Such software is subject to no restrictions whatsoever.

1.16 2 INSTALLATION

Installation requires two floppy disks, or approximately 1.3M of space on your harddisk. KingFisher stores its configuration in the file

```
KingFisher.prefs
--- this file can be modified by hand, although it is
recommended that you do so only when absolutely necessary.
```

NOTE: In earlier versions, KingFisher required that this file be named KingFisher.config rather than KingFisher.prefs. While KingFisher continues to prefer the .config extension it will now also recognize the .prefs extension in accordance with Kickstart 2.0+ guidelines.

KingFisher will work equally well from the CLI and the Workbench.

2.1 Floppy Disk

- 2.2 Hard Disk
- 2.3 Upgrading from v1.0
- 2.4 Operation under a BBS

1.17 2.1 Floppy Disk Installation

Copy the file ``KingFisher.floppy`` to
 KingFisher.prefs
 --- This file
 references the files that KingFisher needs on the floppy disk volumes
 ``KingFisher-1``, ``KingFisher-2`` etc. At present, ``KingFisher-3`` is
 not needed. When you add fish disk 1101, however, you will need that disk.

Copy all other files except the file ``Fish-0501-1100`` to the disk
 ``KingFisher-1``

Example Floppy Disk Setup:

```
-----
S:
    KingFisher.prefs
    * (originally KingFisher.floppy)
KingFisher-1: KingFisher
              KingFisher.info
              KingFisher.doc (May not fit on 837K disks)
              KingFisher.doc.info (May not fit on 837K disks)
              Fish.Index
              Name.Index
              Fish-0001-0500
KingFisher-2: Fish-0501-1100
KingFisher-3: Fish-1100-1700
KingFisher-4: Fish-1701-9999
-----
```

(*) Ideally this file is stored in the same directory where KingFisher itself is started. Another wonderful place to store this file is in the ENVARC:KingFisher/ directory: this causes this file to be copied to the ENV:KingFisher/ directory during startup, using up some RAM in the process, but allowing for somewhat quicker startup because ENV: is usually located on the RAM: disk!

1.18 2.2 Hard Disk Installation

To install KingFisher on a harddisk system,
 create a subdirectory
 for
 KingFisher's files and copy all files to this directory. Copy the file
 ``KingFisher.hd`` to
 KingFisher.prefs
 -- This file references the files

that KingFisher needs in the default directory. See section ? for information on how to tell KingFisher to use a specific volume or directory.

Example Harddisk Setup:

```
-----
S:
    KingFisher.prefs
    *      (originally KingFisher.hd)
DH0:KF/   KingFisher
          KingFisher.info
          KingFisher.doc
          KingFisher.doc.info
          Fish.Index
          Name.Index
          Fish-0001-0500
          Fish-0501-1100
-----
```

(*) Ideally this file is stored in the same directory where KingFisher itself is started. Another wonderful place to store this file is in the ENVARC:KingFisher/ directory: this causes this file to be copied to the ENV:KingFisher/ directory during startup, using up some RAM in the process, but allowing for somewhat quicker startup because ENV: is usually located on the RAM: disk!

1.19 2.3 Upgrading from v1.0

If you are upgrading from version 1.0 you may need to add (or ← modify) one of the following lines in the KingFisher.prefs file. Only one such line may exist in that file:

```
NameIndexFile=Name.Index          (hard disk)
NameIndexFile=KingFisher-1:Name.Index (floppy disk)
```

If this line is not present, KingFisher may not be able to load or save the name index and complain about this fact. If you have run KingFisher without this addition, it will have created such entries but they may lack the proper filenames. You must find these lines and supply the names as in the example above.

1.20 2.4 Operation under a BBS

With the help of two command line options and KingFisher's ARexx ← port, it is now possible to operate KingFisher as part of a bulletin board system in order to offer the system's users the chance to search the contents of the Fish Database.

KingFisher has never been designed for multi-user access. You must take care, therefore, to assure that only one user at a time is allowed to search the database, otherwise a great confusion will ensue, profiting nobody.

Even though multiple copies of KingFisher may be run at one time, this requires a fair bit of RAM overhead. This functionality has not been tested, either, so you are on your own with that, at least for now. I would like to hear from anybody who succeeds with either single or multi-user access to KingFisher.

On to the goodies: KingFisher offers two command line options, described by the following template:

```
SETTINGS\K,NOABOUT\S,TINY\S,STAYBEHIND\S
```

I would like to stress(!) that I am using an extremely simple parser to detect these on the command line. For those of you not entirely familiar with the template above, use the following examples to guide you:

```
SETTINGS=env:kingfisher/kingfisher.prefs
SETTINGS env:kingfisher/kingfisher.prefs
NOABOUT
TINY
STAYBEHIND
```

If the SETTINGS file is missing, KingFisher will attempt to load this file from its accustomed places. If the SETTINGS parameter specifies a null file, then NO .prefs file is loaded at all!

The NOABOUT flag is available only from the command line and serves to prevent the appearance of the "About Window" which can cause access to KingFisher to be delayed until a possibly manual operation rids you of this window and allows ARexx commands to be heard.

The TINY flag is the equivalent of the 'Tiny-RAM' option in KingFisher's preferences file. When this flag is missing or the option turned OFF, KingFisher is significantly more "wasteful" with the allocation of space for the fish and name index. When this flag (TINY) is specified, or the option (Tiny-RAM) is set ON, KingFisher allocates and extends the size of the fish and name index in far smaller chunks, preserving memory but making it much more probable that your memory (RAM) suffers fragmentation if you add more than a few dozen fish to the database.

The STAYBEHIND flag causes KingFisher's screen to open behind all other screens in the system and remain there, as opposed to coming to the front when initialization is complete.

Please see the 'example-scripts' directory for some ARexx scripts and review the ARexx chapter for details on
KingFisher's ARexx Port

1.21 3 CONCEPTS AND MENUS

This chapter is perhaps the most important one in the all of KingFisher's documentation. I highly recommend that you peruse as much of chapter 3.1 (Concepts) as possible: much is in there that may not be obvious, or may surprise you!

- * 3.1 Concepts
- 3.2 Menus

1.22 3.1 Concepts

This section is perhaps the most important of them all. Please take your time to learn about KingFisher's unique features. You may find quite a few surprises!

- Adding Fish
- Boolean Expressions
- Colors
- Copying between Tanks
- Cut & Paste
- Deleting Fish
- Display Database
- Exporting the Database
- Fish.index
- Fishtanks
- Fonts
- Iconify
- Insert Volume Requesters
- KingFisher.config
- KingFisher.prefs
- Modifying the Database
- Name.index
- Output Commands

Printing and Exporting

Reindexing

Searching

Verify Output Cmd

Version Select

Function Keys

1.23 Adding Fish

KingFisher has a powerful parser that makes adding new fish \leftrightarrow to your database easier than any other program currently available! Simply give KingFisher the name of the file (which may contain mail or other message headers, extraneous description text, etc.) KingFisher will find the fish descriptions within all this and add them, wherever they are in the file! What could be easier?

Under Kickstart 1.3 KingFisher uses the arp.library, under 2.0+ it uses the asl.library for its file requester to select the input file from which to read contents. The name of this input file is stored in the

KingFisher.prefs file for use as the default in future ``Add Fish`` operations. KingFisher's .prefs file will contain a line such as

```
AddFishReport=CON:80/20/580/160/Add Fish Status Report/AUTO/WAIT
```

which you may, if you wish, modify to specify a proper file on disk. That file (a console window in this case) will receive any messages that KingFisher generates during the ``Add Fish`` operation.

Reindexing is not required after adding fish to the database.

1.24 Boolean Expressions

This section serves as a small tutorial for creating boolean \leftrightarrow expressions to search for programs more selectively than by a single search string.

Boolean expressions are used by KingFisher's search algorithm for enormous flexibility in specifying complex search criteria. For information on how to control KingFisher's selection of such boolean expressions vs. the use of the more conventional simple character string, please see Searching

for more information.

A boolean expression consists of multiple character strings that are connected by boolean operators & (AND), | (OR), and ~ or ! (NOT). Operator evaluation is left-to-right, meaning that AND has no special precedence over OR. Parentheses are used to enforce specific evaluation order. Blank spaces are ignored in the expression.

A character string that is to contain blanks, parentheses, or any symbol ordinarily used as an operator (&|~!) must be enclosed in matching single (') or double (") quotes.

Ex: A relatively simple boolean expression would be the following, which searches for a game that has graphics and sound:

```
game & graphics & sound
```

In plain English this reads: ``game AND graphics AND sound'' -- thus this string will cause any entry to match which contains the string "game" AND the string "graphics" AND the string "sound"

If you are looking, however, for a game that has EITHER graphics OR sound (or maybe both) you might specify the following:

```
game & graphics | sound
```

The problem with this expression is that while most conventional computer languages use something called ``operator precedence'' KingFisher simply evaluates expressions from left to right. Using parentheses to describe what is actually evaluated first, we can immediately see that what we wrote was not what we intended:

```
( game & graphics ) | sound
```

But wait! What we have now is a perfectly valid expression for KingFisher. Let us use parentheses, therefore, to TELL KingFisher what to evaluate in what order:

```
game & ( graphics | sound )
```

That looks a lot better, doesn't it?

Ex: Let us now try to filter out any sort of demonstrations for commercial games, just in case. It is usually best to use short substrings rather than less general, long strings. So we want to add to our expression the fact that we do NOT want entries that contain the string 'demo' combined with 'commercial':

```
game & ( graphics | sound ) & ~( demo & commercial )
```

We added another AND to the expression followed by a funky symbol: ~ This symbol (you could also have used the ! instead; KingFisher doesn't care which one you use) represents what is called a LOGICAL NEGATION. If KingFisher locates the strings ``demo'' and ``commercial'' in a description it will then negate the value of that find; the AND expression will, therefore fail, and the entry from the database is not considered a match!

Boolean expressions are limited to 250 characters in length. There is otherwise no practical limit set on the complexity of the expression. KingFisher is generally able to detect which search keys are expressions, and which are just simple character strings. The overhead of expressions causes searches involving these to be slightly slower.

KingFisher's boolean expressions can also contain DOS Regular Expressions if those are enclosed in quotes. These can greatly enhance the subtlety of your search criteria. ←

1.25 Colors

KingFisher will honor color values expressed in either decimal or hexadecimal (prefixed with `'0x'`) via an entry `'colors'` in the `.prefs` file. Up to eight values can be specified, separated by commas. It is generally far easier to use Workbench's `'Colors'` tool to modify colors than it is to edit the `KingFisher.prefs` file by hand, until KingFisher provides its own color requester. ←

How to: drag down the KingFisher screen (making sure it remains the front-most screen) to reveal the Workbench screen. The Utilities drawer (Tools under KS2.0+) has a program named `'Colors'` or `'Palette'`. When you execute this program it will open its window on the front-most screen (KingFisher in this case), allowing you to modify the colors for KingFisher's screen. KingFisher will save these colors and reuse them in the future until they are changed again.

By default, KingFisher uses all eight colors. If you wish to use fewer colors, you should modify the entry `'display-depth'` in the `KingFisher.prefs` file, giving it a value of 2 or 1 to represent the number of bit planes to be used. A value of 1 uses two colors. A value of 2 uses four colors, and the default value of 3 uses all eight colors. KingFisher's visual appeal will suffer with fewer than 3 bitplanes, but some display modes (such as the A2024) are limited to 2 bit planes and (for now) you must force KingFisher to use fewer bit planes to achieve functionality with these special displays.

KingFisher can highlight keywords upon a successful search. The color used for this highlighting is selected with the entry `'key-color'` in the configuration file.

1.26 Copying between Tanks

In order to save and repeatedly restore to a favorite position in the database, you can copy the settings of a specific fishtank (q.v.) to overwrite the contents of the current tank. This would allow you to easily reposition yourself at any `'fish'` of your choosing after, for example, a

search operation.

1.27 Cut & Paste

It is possible to use programs such as ``Snap`` to cut out portions of the description. This requires that the font used for the description text is a non-proportional font. KingFisher's default font, diamond, is a proportional font and is not recognized by programs such as ``Snap``.

1.28 Deleting Fish

When you issue the ``Delete Fish`` command all fish starting with the currently displayed entry are deleted from the database. This does not physically modify the database itself, but rather modifies the indices only. If you issue a ``Reindex`` command after a 'Delete Fish' command, the deleted fish will be automatically re-added to the database. This is because the ``Reindex`` command ignores the index and looks only at the actual database itself.

1.29 The Display Database

KingFisher's default display mode is 00008000 which \leftrightarrow corresponds to a non-interlaced hires mode using the default monitor for a resolution of 640x200 or 640x256. 00008004 is the same, interlaced. The following table should cover most needs:

Mode ID	Resolution (PAL)	Description
00008000	640 x 200 (256)	Hires
00008004	640 x 400 (512)	Hires-Interlace
00008020	1280 x 200 (512)	SuperHires
00008024	1280 x 400 (512)	SuperHires-Interlace
00039020	640 x 480	Productivity
00039024	640 x 960	Productivity-Interlace
00041000	1008 x 800 (1024)	A2024-10Hz
00049000	1008 x 800 (1024)	A2024-15Hz

NOTE: Display modes that are limited to FEWER than eight colors (3 bitplanes) will not function correctly with KingFisher unless an entry in the KingFisher.prefs file (display-depth) is modify to indicate 1 or 2 bitplanes (as required; try 2 for starters.) Values from 1 to 3 are acceptable, with 3 bitplanes (8 colors) being the default.

After running KingFisher once (and quitting the program again, thereby saving the KingFisher.prefs file) edit the KingFisher.prefs file and look for a line such as

```
display-mode=00008000
```

Change this 8 digit hexadecimal number to one of the codes listed above, or another display mode value that you know. You can add ``Overscan`` to this entry if you want an overscanned display:

```
display-mode=00008004,Overscan
```

This example represents an overscanned hires interlaced display. The overscan keyword causes KingFisher to use TEXT OVERSCAN as defined by you in the Overscan Preferences.

Please see the file graphics/displayinfo.h for other examples.

1.30 Exporting the Database

Using one of the Output commands, KingFisher can generate files that can contain the results of a search (Output, Filtered) the current disk, a range of disks, or the entire database. The ``Clear`` gadget will erase the export file. The ``Start`` operation will then append its output to an empty file rather than to the end of a previously existing text.

The name of the export file is specified in the KingFisher.prefs file under the label ``Export-File``.

1.31 The Fish.index File

This file contains a 16-byte entry for every ``fish`` in the database. It is the size of this file, therefore, that determines how many fish the database contains. When you delete fish from the database, KingFisher adjusts the size of this index, but does NOT truncate the database files themselves. When you later add fish to the database again, KingFisher adds them to the logical end of file, which may overwrite no longer used portions of the database files.

1.32 Fishtanks

Eight such ``bookmarks`` are maintained, each representing a different view of the database, including its own current position in the database, a set of search strings and search range. You can quickly switch between different places in the database, perhaps for comparing similar entries.

The contents of any Fishtank can be copied into the currently active tank with the menu command

Copy from

command. One use of this function is

resetting a fishtank to a particular set of values, perhaps blank search strings, or a particular position in the database. Notice that the menu shortcuts for these commands have a striking resemblance to the actual Fishtank numbers.

1.33 Fonts

KingFisher can use proportional fonts, including scalable fonts, ↵
for the
contents description window. The
KingFisher.prefs
file will provide a
default font like `''contents-font=diamond/12''`; a value for the label
`''scroll-speed''` (if not zero) specifies the number of pixels that one line
of text uses. If you provide a value of 1 then you are asking KingFisher
to do 1-pixel smooth scrolling. This can be terribly slow.

Under AmigaDOS 2.0 or later, KingFisher's menu option

Description Font

will allow you to select a font, including styling such as bold, ↵

italics,

and underlined text. Under earlier versions you can specify styling by
adding appropriate symbols to the `''contents-font''` entry in the
KingFisher.prefs file: to select a bold and italicized Topaz/9 font,
specify it thus:

```
contents-font=topaz/9-bi
```

Notice the addition of a dash `''-''` to the entry, followed by two letters,
`''b''` and `''i''`. These represent the bold (b) and italic (i) styles.
Which letter is used for underlined text? You get one guess!

1.34 Iconifying KingFisher

Using Leo Schwab's `iconify()` function, KingFisher will ↵

iconify to the

workbench screen. The icon will always be placed where it last appeared,
and double clicking on this icon will cause KingFisher to reopen the
display.

Iconifying causes KingFisher to try to save

KingFisher.prefs

the file.

Why? It is easier to accidentally reboot the computer (and lose changes to
KingFisher's setup) when there is only an icon for KingFisher on the
Workbench screen than when KingFisher's screen is open. If you do not
agree with this idea, then turn off the menu option

Safe Iconify
, and

KingFisher will not save the KingFisher.prefs file when you iconify.

1.35 Insert Volume Requesters

These are presented when KingFisher requires a volume to be mounted (inserted) which is NEEDED but not available. It is important that you give KingFisher the disks it asks for to prevent problems with access to the database. If KingFisher gets confused, the best action is to quit the program and restart it.

1.36 The KingFisher.config File

Up until version 1.15 KingFisher used only the file `KingFisher.config`.

While this file remains preferred over the new KingFisher.prefs file when both are encountered, it is now preferred and recommended that this file is named

`KingFisher.prefs`
in preparation for KingFisher's eventual

conversion to a Kickstart 2.0-only program.

1.37 The KingFisher.prefs File

This file is expected to be stored in the default directory. If `KingFisher.prefs` is not found

there, KingFisher will look for it in the directory `ENV:KingFisher/` (which is a duplication of `ENVARC:KingFisher/`). If not found there, either, the file should be located in a volume named `CONFIGS:`, `CONFIG:`, or `CNF:` If not found in any of these, either, KingFisher will try `S:` and then give up, presenting you with a requester to this effect.

As of version 1.2 KingFisher now supplies a set of default values for this file, some of which have traditionally been required. If this file does not exist, KingFisher will use a set of defaults, which may or may not allow KingFisher to function, depending on how closely these defaults match your setup.

An alternative name for this file is `KingFisher.config`. If KingFisher finds both `KingFisher.config` and `KingFisher.prefs` in the same directory, it will use `KingFisher.config` rather than the newer name, `KingFisher.prefs`.

If this file is ever corrupted or deleted, you may need to take steps to

rebuild
this file.

Do not modify this file while KingFisher is running. This is because KingFisher will overwrite this file when you use the `rebuild` command to

Save Settings
or when you iconify or exit KingFisher. You would lose all
your changes.

KingFisher saves this file when you quit, when you iconify the program, or when you issue the command ``Save Settings.'' If you do not wish to have this file saved when you quit, unselect the ``Keep Settings'' option in the OPTIONS menu. If you do not wish to have this file saved when you iconify the program, unselect the

Safe Iconify
option in the OPTIONS menu.

1.38 Modifying the database

The database files (Fish-xxxx-yyyy) are the most valuable ←
portion of
KingFisher. It is, therefore, not recommended that you modify them
casually, and if you find it necessary to modify the files, do so with a
BACKUP COPY first!

If you do wish to correct spellings or other mistakes, you MUST use a text editor that can work with long lines. (GNUemacs is such an editor.) KingFisher stores each fish on two (and only two!) lines of text. The first line contains the name of the program, the second line, the ENTIRE description. A completely blank line is inserted between fish disks so that the 'Reindex' command can function properly.

After you modify the database you should use the
Reindex
command.

1.39 The Name.index File

Contains one program name per text line, perfectly matching the Fish.index file. This index is read into memory, providing extremely fast searches for the name of a program.

1.40 Output Commands

Six output commands exist:
Fish
and
Fish w/Disk#
output information
about the currently displayed fish only.

The command

Disk
creates output on all fish on the current disk.

Range

generates output on all fish on all disks in the current range ←
of

disks WITHOUT REGARD TO THE STATE OF THE

Use Search Range

CHECKMARK!

Whole Database

creates output using the entire database.

The

Filter (search)

command uses the currently selected search string to
output only those fish that match this FORWARD search. If you have a
search range selected, the output operation will be restricted to those
disks, otherwise it will proceed from the current disk to the end of the
database. This behavior is consistent with a normal search operation.

Filter (search)

output does not, at present, honor the search mode
selector, i.e. it performs a full text search, rather than using the name
index.

1.41 Printing and Exporting

This command doubles for both printed output and file output.

The OPTIONS/Output Format menu provides you with a variety of choices that
affect the output you generate:

Output Width

(Determined by preferences or from KingFisher.prefs)

Output To

(Either the printer or the file in KingFisher.prefs)

Output Format

(Short (name-only), or Long form (full description))

When output goes to a file, KingFisher always appends the text to an
existing output file. This allows you to composite listings, or combine
the results of multiple searches (see

Output, Filter

.) If you wish to

start with an empty file, you can use the

Clear

gadget to erase the

file's old contents before beginning output.

1.42 Reindexing

The ``Reindex`` command assumes that the database files are the sole indicator of any contents. If you have deleted fish from the database, then added some fish again, the logical end of file may occur before the physical end of file, meaning that there is junk stuck to the end of the database which will confuse the ``Reindex`` command. You may need to issue another

Delete Fish command to remove such junk or even truncate the database files manually to that effect. There is no easy solution to this problem right now.

You can be alerted when the ``Reindex`` operation is finished by selecting the

Reindex done
alert from the options menu.

Please note that a damaged database file can cause the 'Reindex' operation to fail or subsequent operation of KingFisher to produce erratic results. If you experience unexpected problems after a 'Reindex' operation, please contact me with as much information as possible.

1.43 Searching

Search capability is active only when one of the six search gadgets is selected, thereby coloring the

Fish Select gadget in a rather noticeable yellow to match the yellow of the search string selection gadgets. If a search gadget is selected, the

Fish Select gadgets will also become yellow to match the checkmark. If you have also selected the

Use Search Range gadget, then your search will be limited to that range of disks.

The

Search Mode gadget, located to the immediate right of the

Fish Select gadget, displays either a microscope (precise search, but slow) or a magnifying glass (less detailed search, though much faster.) The less detailed search (magnifying glass) scans only the program name, providing an extremely fast method of locating a program if you know at least part of it's name. The precise search (microscope) scans through the name and the entire description, which requires more time, of course.

The

Output, Filter (search)
 command uses only the precise search, even if
 you the
 Search Mode
 is set to use the program names only.

You can be alerted when a search succeeds or when it fails, by selecting
 either or both of the
 Search ok
 or Search fails alerts from the options
 menu.

The
 Search Setup
 submenu offers several choices to configure the way
 searches are conducted. By default, KingFisher will ignore different
 letter cases (upper/lower) treating them as equivalent; also, KingFisher
 will employ the "simple" search that previous versions used. If this
 option is turned off, KingFisher assumes that the search strings you give
 it are boolean expressions (q.v) -- if what you give it is not a boolean
 expression, KingFisher will complain. A way around having to make this
 decision yourself all the time is to select the
 Auto Mode Detection
 --

KingFisher will then try to determine if the search string is a boolean
 expression or not, and use the appropriate algorithm. Due to a variety of
 factors, KingFisher's
 Auto Mode Detection
 could be fooled; turning this
 option off, and selecting the proper mode (simple or not simple) will allow
 you to properly work with problematic search strings.

While the boolean search algorithm provides a great deal of flexibility,
 the evaluation overhead exacts a speed penalty of perhaps some 10% over the
 simpler string comparison. More complex expressions have increasingly
 worsening impact.

The regular expressions that are part of KickStart 2.0 and later are
 available only in the complex (boolean) expressions, and these regular
 expressions must be quoted strings. Example:

```
editor & "#?author:#? mat(t|hew)#?dillon#?"
```

Notice especially that the regular expression is begun and ended with the
 wildcard symbol #? --- the regular expression must match the entire
 description, including the program name, which is separated from the
 description by a newline symbol (one character.)

1.44 Verify Output Cmd

By default, KingFisher will always wait for you to click on the "Start"
 gadget when you request printer output or an export operation. If you
 disable the "Verify ..." option, KingFisher assumes that you know what
 you are doing and will proceed at once without waiting for you to click the
 "Start" gadgets. Notice that you will not be given the chance to click

```
``Clear.``
```

1.45 Version Select Buttons

During the Add Fish and the Reindex commands, KingFisher will attempt to create links between programs and their more recent updates. This process relies on the fact that nearly all updated programs say something like ``this is an update to version 0.9 on disk 57``. The ``Version Select`` gadgets will not do anything if there is no previous or following version to the current entry.

Starting with version 1.2, KingFisher's 'Reindex' command recognizes and properly links multiple parts of an entry. Earlier versions failed to properly handle such special cases.

1.46 Function keys

The 10 function keys on your keyboard can each invoke an external ↔ program.

You specify a program by editing the KingFisher.prefs file in the directory ENV:KingFisher/ or wherever you have this file stored. Look for a sequence of entries such as "FKey-0=", "FKey-1=", ... "FKey-9=". Each of these specifies a function key. The keys are numbered 0 through 9 internally, but F1 through F10 on your keyboard. To edit F5, you should modify FKey-4 in the file.

The following is a list of possible commands that you can implement almost at once, and with very little effort:

1. FKey-0=AmigaGuide KingFisher.guide
This invokes
AmigaGuide
on your Workbench rather than on the
screen that KingFisher uses. On my system, Workbench is
interlaced and KingFisher is not. AmigaGuide looks nicer on
the interlaced Workbench.
2. FKey-1=sys:utilities/more t:KF.output
Invoke the MORE program on the default export file so you can
see what a search produced. Substitute an editor for the MORE
program to also have a chance at modifying the file.
3. FKey-2=type >prn: t:KF.output
Print the default export file (i.e. type it to the printer
device.) Yes, I/O redirection will work!

Here are some more examples, which require you to provide the most

important pieces, the scripts, commands, and programs:

4. FKey-3=execute KF:download-current-fish
Invoke a script (not provided) to download the current fish displayed by KingFisher.
5. FKey-4=rx KF:get-current-fish.kf
Invoke an
ARexx
script that loads your terminal program,
dials up and performs a login to your mainframe account, then
issues the relevant ftp commands to connect to a site that has
the fish disks online, transfers the file to your local
account, downloads that to your Amiga, cleans up after itself,
unarchives the file for you in the background, and meanwhile
returns you to KingFisher.

The writing of this script is left as an exercise for you! :)

1.47 3.2 Menus

KingFisher's menus may change somewhat from time to time to ←
accommodate

alterations in the overall workings of the program. Some menu entries may be grouped into submenus, or reordered in the future. It is more important that you understand various menu items' functionality, rather than memorizing their ordering or exact position in the menus.

When operating under KS2.0, you can press the HELP key while operating the menus to obtain help on the item currently under the cursor (regardless of the item's state.)

Fish

Options

1.48 3.2.1 The Fish Menu

Add Fish

Delete Fish

Reindex
Output

Fish

Fish w/Disk#

Disk

- Range
- Whole Database
- Filter (search)
- Copy from ...
- Use Settings
- Save Settings
- About
- Help
- Quit

1.49 3.2.2 The Options Menu

- User Interface
- AutoActive Gadgets
- Keyboard Control
- Wide Search String
- Keep Settings
- Description Font
- Titlebar
- Confirm Quit
 - Output Format
- Width
- Output to
- Format
 - Alerts
- Reindex done
- Search ok
- Search fails
- Background Reindex
- Verify Output Cmd

```
Safe Iconify
  Search Setup

Case Sensitive

Simple Mode

Auto Mode Detection

Allow DOS Patterns

Trim Blanks
```

1.50 Add Fish

Reads the contents of a file and parses it for fish disks and their contents descriptions. The parser is quite clever, knows where disks and entries start and end, and knows to skip headers and signature blocks.

The parser also removes hyphens from the end of lines whenever those hyphens connect two lower-case letters. This eliminates all hyphenations, while occasionally legitimate dashes are victimized by this. The benefits of one outweigh the drawbacks of the other, in my humble opinion. Future versions of KingFisher will provide for a better algorithm.

Reindexing is not required after this operation.

1.51 Delete Fish

```
Removes one or more fish starting with the currently displayed ←
one. You
will be asked if you really wish to do this. If you respond affirmatively,
KingFisher will chop the index to indicate that these fish are now gone.
```

It is important for you to realize the following:

- 1) the index is saved only with the
Save Settings
command or when
KingFisher exits properly,
- 2) that the database files themselves are not altered: when next you add
fish to the database, the new fish will overwrite where the last indexed
fish left off. If you, therefore, accidentally delete fish from the
database, you can
rebuild the index
to recover the deleted fish!

1.52 Reindex

Parses the database files and rebuilds the index, including the version information, etc. It is useful for a variety of purposes, not the least of which is undoing a mistaken Delete Fish command. It is also useful after you have modified the database files (with a text editor that can handle loooooong(!) lines.)

Modifications to the database done within KingFisher automatically adjust the index for you. The 'Reindex' command exists to make KingFisher aware of changes to the database that were made not under the program's control.

1.53 Output

Depending on the setting of the Output To menu option, the output command will produce either printed output or file output (export functionality.) The following output commands are available:

Fish
 Fish w/Disk#
 Disk
 Range
 Whole Database
 Filter (search)

1.54 Output, Fish

Uses only the currently displayed fish entry for output.

Depending on the setting of the Output Format, the output may consist either of the program's name (each on a line of its own):

KingFisher

or of the program's name, including the description:

KingFisher A specialized database tool providing maintenance and search capabilities for the descriptions of disks in the format used by this library.

KingFisher's database can span multiple (floppy) disk volumes, can be edited by text editors that support long text lines, can add disks directly from unedited email or usenet announcements, can remove disks, rebuild a damaged index, find next or previous software versions, print or export (parts of) the database, and more. Includes a database of disks 1-800. This is version 1.15, an update to version 1.11 on disk 783. Binary only. Author: Udo Schuermann

The Output commands produce different results depending on the values in the 'Output Format' option menu. You can send output to either the printer or a disk file, and output only the disk number and the name or a full textual description of fish entries. The custom output width (as opposed to Preferences-determined width) can be modified via the

```
KingFisher.prefs
file's Output-User setting. The range of acceptable values is ←
40 through
```

240.

1.55 Output, Fish w/Disk#

Uses only the currently displayed fish entry for output, but ← includes the string ``(Disk #...)'' immediately before the description.

Depending on the setting of the Output Format, the output may consist either of the program's name with disk number (each on a line of its own):

```
KingFisher      (Disk #808)
```

or of the program's name, including disk number and description:

```
KingFisher      (Disk #808) A specialized database tool providing
maintenance and search capabilities for the
descriptions of disks in the format used by this
library. KingFisher's database can span multiple
(floppy) disk volumes, can be edited by text
editors that support long text lines, can add disks
directly from unedited email or usenet
announcements, can remove disks, rebuild a damaged
index, find next or previous software versions,
print or export (parts of) the database, and more.
Includes a database of disks 1-800. This is
version 1.15, an update to version 1.11 on disk
783. Binary only. Author: Udo Schuermann
```

The Output commands produce different results depending on the values in the 'Output Format' option menu. You can send output to either the printer or a disk file, and output only the disk number and the name or a full textual description of fish entries. The custom output width (as opposed to Preferences-determined width) can be modified via the

KingFisher.prefs
 file's Output-User setting. The range of acceptable values is ←
 40 through
 240.

1.56 Output, Disk

Uses the current fish disk for output. All entries on the ←
 currently shown
 disk (regardless which entry is being shown at the moment) will be output.

Depending on the setting of the
 Output Format
 , the output may consist
 either of the programs' name with disk numbers, or may also include the
 entire description of each fish. If the entire description is included,
 each entry on the disk will be separated from the next by a blank line.

The number of the disk is output at the top of the list.

The Output commands produce different results depending on the values in
 the 'Output Format' option menu. You can send output to either the printer
 or a disk file, and output only the disk number and the name or a full
 textual description of fish entries. The custom output width (as opposed
 to Preferences-determined width) can be modified via the

KingFisher.prefs
 file's Output-User setting. The range of acceptable values is ←
 40 through
 240.

1.57 Output, Range

Uses all disks in the currently selected range (if the range is ←
 checked for
 searching or not) for output. Output will otherwise conform exactly to the
 format of the

Output, Disk
 menu command.

The Output commands produce different results depending on the values in
 the 'Output Format' option menu. You can send output to either the printer
 or a disk file, and output only the disk number and the name or a full
 textual description of fish entries. The custom output width (as opposed
 to Preferences-determined width) can be modified via the

KingFisher.prefs
 file's Output-User setting. The range of acceptable values is ←
 40 through
 240.

1.58 Output, Whole Database

Uses all disks in the entire database for output. Output will conform exactly to the format of the Output, Disk menu command.

The Output commands produce different results depending on the values in the 'Output Format' option menu. You can send output to either the printer or a disk file, and output only the disk number and the name or a full textual description of fish entries. The custom output width (as opposed to Preferences-determined width) can be modified via the

KingFisher.prefs file's Output-User setting. The range of acceptable values is 40 through 240.

1.59 Output, Filter (search)

This command filters each entry through the currently selected search expression and causes it to be output only if the entry matches the search expression. In other words, this command is a "batch search" command.

There are a few points to be made clear about this command, which may not be obvious:

1. If the

Output Format is set for 'Full Text' then the string "(Disk #...)" is placed immediately ahead of the description text, as is the case with the Output, Disk command.

2. Regardless of the selection of the

Search Mode gadget, this command does not limit itself to the searching program name only.

The Output commands produce different results depending on the values in the 'Output Format' option menu. You can send output to either the printer or a disk file, and output only the disk number and the name or a full textual description of fish entries. The custom output width (as opposed to Preferences-determined width) can be modified via the

KingFisher.prefs file's Output-User setting. The range of acceptable values is 40 through 240.

1.60 Copy from ...

Allows you to copy the contents of another Fishtank into the current Fishtank's gadgets. Perhaps the best use of this function is to reset the current Fishtank's configuration to a particular set of values defined by another Fishtank.

1.61 Use Settings

This function writes the settings only to the ENV:KingFisher/ directory, thereby making them available for subsequent use so long as your Amiga is not rebooted or otherwise experiences a failure.

Together with the Save Settings option, KingFisher provides the same functionality as that given by the more common Preferences utilities, which also offer both "Use" and "Save" options.

1.62 Save Settings

Saves both the index files (if they need saving) and the KingFisher.prefs (or KingFisher.config) file to preserve the current gadget selections and menu settings. This command is automatically executed for you when you quit KingFisher, unless the Keep Settings option is off.

Regardless of that option, the indices are always saved upon exit if they need saving.

1.63 About

Displays the about window with the author's address, a picture of a kingfisher (bird) and information on where in the database you are currently located, how many fish there are in it, and how many fish disks these comprise. The size of the KingFisher executable program is also indicated here to help in detecting virus infections.

Click in the About window to get rid of it.

1.64 Help

This command produces the main HELP (Contents) screen of KingFisher's Guide file. The file KingFisher.guide must exist, and the amigaguide.library must be installed in LIBS: for this to function.

You can obtain context sensitive help by positioning the cursor over a gadget or a menu item and then pressing the HELP key. Help with menus is available only with Kickstart 2.0 or later, but the gadget help functions with Kickstart 1.3, as well.

Keyboard shortcut: HELP

1.65 Quit

Is to exit KingFisher. The index is saved (if necessary) and ← if the

Keep Settings
option is on, the KingFisher.prefs file is updated. If the

Confirm Quit
option is selected, then KingFisher requires you to respond affirmatively to a requester before actually quitting.

1.66 User Interface

This menu controls some aspects of KingFisher's User ← Interface. In particular, KingFisher can be told to keep all string gadgets active as much as possible (this is NOT recommended for Kickstart 2.0+) or offer keyboard control (highly recommended for Kickstart 2.0 users). Optionally, the search string gadgets can cause a wider string requester to appear, which offers a better view of long search expressions.

The following user interface options are available:

AutoActive Gadgets

Keyboard Control

Wide Search String

But other options are also available in the Options menu:

Description Font

Confirm Quit

Verify Output Cmd

1.67 AutoActive Gadgets

This option is mutually exclusive with the
Keyboard Control
and the

Wide Search String
menu options.

The "AutoActive Gadgets" option causes the string gadgets to remain active as much of the time as possible. They will be re-activated after every gadget or menu selection. Under KS 2.0 this can present problems, since menu shortcuts are then no longer recognized. A program named ``PatchOS`` on Fish Disk 706 provides some relief to this problem, although it only works properly with commands and not options.

As of version 1.30 of KingFisher it is recommended to use the

Keyboard Control
option instead.

1.68 Keyboard Control

This option is mutually exclusive with the
AutoActive Gadgets
option.

When the "Keyboard Control" option is selected, KingFisher's console will operate with much greater conformity to 2.0 Style Guidelines:

If a text gadget IS active:

RETURN, ENTER Deactivates the currently active gadget. This enables the use of the cursor keys to move through the database. This functions under both KS 1.3 and KS 2.0

TAB, SHIFTTAB Skip to and active next gadget. This functions only under KS 2.0

HELP Obtains context sensitive help from the KingFisher.guide file. This functions only under KS 2.0, and requires amigaguide.library to be installed in your LIBS: directory, and the KingFisher.guide file to be located in the current directory (where KingFisher is stored) or its name specified by the Help-File specification in the

KingFisher.prefs
file.

If NO text gadget is active:

RETURN, ENTER Activates the next gadget in a circular manner. Pressing the RETURN (or ENTER) key repeatedly will (slowly) move through all text gadgets. This functions under both KS 1.3 and KS 2.0

LEFT, RIGHT (alone) Move to the previous or next fish, WITHOUT REGARD TO WHETHER OR NOT THE
Select Fish
gadgets are
yellow and would initiate a search.

LEFT, RIGHT (+SHIFT) Move to the first or last fish in the database.

LEFT, RIGHT (+CTRL) Move to the first fish on the previous or next disk in the database.

LEFT, RIGHT (+ALT) Move to the previous or next version of the current program (provided there is one.) If nothing happens when you press this key (or select one of the version gadgets) then the current program has no previous or following version linked to it in the database.

UP, DOWN Scroll the text display panel one line up or down, to reveal text that is not visible. This will have an effect only if there is more text than actually fits in the display panel. If the proportional gadget (the vertical slider) does not fully fill its box, then you can slide it down with these keys.

HELP Obtains help from the KingFisher.guide file depending on where your mouse cursor is pointing. If it points at a gadget, help will be brought up on whatever gadget you are pointing at (string gadgets, buttons, etc). If you are not pointing at anything that has special help available to it, KingFisher will simply present you with the index from where you can browse to your heart's content!

CTRL-A, CTRL-E Move to beginning or end of the current range, even if the 'Use Range' button is off.

CTRL-B, CTRL-F Move backward or forward by one fish.
p, n Move to previous or next fish.

P, N Move to previous or next disk.

CTRL-P, CTRL-N Move to the previous or next version.

[,] Move to the first or last fish in the database.

CTRL-R, < Search reverse
CTRL-S, > Search forward

1, 2, 3, 4, 5, 6 Toggle selected search string.

F1, F2, ... F10 Invoke
 external command

1.69 Wide Search String

This option is mutually exclusive with the
AutoActive Gadgets
option.

The original search string gadgets were mostly sufficient for the entry of simple key strings. With the introduction of boolean expressions, however, entries can easily exceed the visible width of these gadgets. This option, therefore, causes a wider string requester to appear, which offers more than three times the width.

This string requester is terminated either by pressing the RETURN key, or by clicking on the "Accept" gadget. An added benefit of using this wider string requester is that you can modify the string and terminate it with the "Cancel" gadget instead, thereby undoing the editing.

1.70 Keep Settings

Preserves the current contents of the search gadgets, the state ↔
of their
selection, the current position in the database, and all option selections from the menus. If you turn this option off and then quit the program, the

KingFisher.prefs
file will not be updated, and the 'Keep Settings' value

is not updated there either. If you wish to make the 'Keep Settings' option permanent, use the

Save Settings
command.

1.71 Description Font

Is available only under Kickstart 2.0 because it needs the ↔
ASL font
requester to work. The font currently in use will be shown as the default, and any font you select is then used for the description text panel. At this time, no other font can be changed. If you are operating under Kickstart 1.3 you can teach KingFisher a different

font
by modifying the
``contents-font`` entry in the file
KingFisher.prefs
.

1.72 Titlebar

Allows the titlebar to be removed from the window. Under the operating system before 2.0 this removes the only standard way to depth arrange or slide the KingFisher screen up or down. The availability of menus is not affected by the lack or presence of the titlebar.

1.73 Confirm Quit

Presents you with a requester whenever you are about to quit the program so that a mistaken keystroke (rAmiga-Q) does not require you to laboriously reload KingFisher. Most useful for Floppy Disk installations.

1.74 Output Format Options

The output format menu defines how the Output menu will operate: \leftrightarrow
either as
a printing or an export function. The following options are available:

Width

Output to

Format

1.75 Output Format, Width

Specifies the width of your output. For printer output it is \leftrightarrow
usually best
to use `''Preferences''` as this examines your printer preferences. `''n
columns''` is best used for export to a file.

You can modify the actual number of columns used by changing the entry
'Output-User' in the

KingFisher.prefs

file. A value from 40 through 240

is acceptable.

1.76 Output Format, Output to

Determines if you send output to the printer (the PRT: device) or a disk file (defaults to `''t:KF.output''`.) In essence, this determines if the `''Output''` commands function as Print or as Export commands.

1.77 Output Format, Format

Selects between the short format that produces single-line listings of the fish disk number and the name of the program, and the full format that most are used to seeing from the Contents files and Usenet postings.

1.78 Alert Options

Alerts use the system's DisplayBeep() function to alert you to the conclusion of an operation which can take longer than you may be willing to sit and watch it happen. The Sound Preferences (WB 2.1+) offer the means to attach a digitized (or simple beep) sound to the DisplayBeep() function, thereby providing an audible cue.

The following alerts are available:

Reindex done

Search ok

Search fails

In addition to the Reindexing alert, the Reindex command can be made to operate in the background. In all honesty, this only saves you the effort of depth adjusting KingFisher's screen once the reindexing operation has begun:

Background Reindex

1.79 Alerts, Reindex done

Produces a beep (display flash) when reindexing is completed. This is especially useful if you are doing other things on your computer and wish to be alerted when the index has been rebuilt.

1.80 Alerts, Search ok

Produces a beep (display flash) when a search is successful, i.e. a match has been found.

1.81 Alerts, Search fails

produces a beep (display flash) when a search is is not successful, i.e. a search fails.

1.82 Background Reindex

Causes KingFisher to drop its screen to the back as soon as the Reindex command begins. This is a simple convenience feature so you don't have to depth arrange the screen yourself.

If you wish to be alerted when the ``Reindex'' operation is done, you can select the

```
Reindex done
alert..
```

1.83 Verify Output Cmd

Allows you to by-pass the confirmation requirement (involving the pressing of the ``Start'' gadget) whenever you use one of the ``Output'' commands to print or export data. Since some of these commands can create rather sizeable disk output, and since not everyone has a printer attached to their computer, it is recommended that you leave this option enabled to prevent mistakes.

1.84 Safe Iconify

Allows KingFisher to save the KingFisher.prefs file when you iconify the program. With floppy disk installations this may be too time consuming or cumbersome. You may also not feel paranoid enough about losing your current configuration.

1.85 Search Setup

KingFisher's search functions are somewhat configurable, according to the following menu options:

```
Case Sensitive
```

```
Simple Mode
```

```
Auto Mode Detection
```

```
Allow DOS Patterns
```

```
Trim Blanks
```

It is worth noting here that

When Simple Mode is selected and not ghosted,

- a) DOS Patterns are NOT available
- b) Trimming blanks is available

When Auto Mode Detection is selected,

- a) Simple Mode becomes ghosted,
- b) DOS Patterns are available,
- c) Trimming blanks is NOT available

1.86 Search Setup, Case Sensitive

When selected, requires a search to match exactly as opposed to considering upper and lower case characters equivalent. This functions with all aspects of searching.

1.87 Search Setup, Simple Mode

This setting determines if search strings are treated as ↔

Boolean Expressions or ``simple`` keyword searches. This selection is overridden by the Auto Mode Detection setting.

If Simple mode is checked AND not ghosted, the input search string is treated as a simple string, regardless of the presence of special symbols, such as & | ! ~ (or)

1.88 Search Setup, Auto Mode Detection

When this option is selected, KingFisher disables the Simple Mode option and will instead try to determine whether or not to treat the search string as a

Boolean Expression . Only when this option is NOT selected will the "Simple Mode" option have any effect.

1.89 menusearchallowdospatterns

DOS Patterns are regular expressions that are part of AmigaDOS ←
 since KS 2.0
 and provide more search power than KingFisher's
 Boolean Expressions
 , at a
 speed penalty, of course. KingFisher will recognize DOS Patterns only if
 they are enclosed in quotes (single or double.)

If you wish to perform a search for a string that includes special symbols,
 such as question marks (?) or hash marks (#) inside a quoted string, then
 you must make sure that KingFisher performs a
 Simple Search
 .

Naturally, regular expressions function only if you are running KingFisher
 under AmigaDOS 2.0 or later.

The following DOS Pattern will search for a simple keyword, just like a
 simple keystroke would:

```
"#?kingfisher#?"
```

Notice that keyword is enclosed by the pattern #? to match anything before
 and after it. Notice that the ""'s before and after the expression are
 part of how KingFisher determines that this is something more than a simple
 search string.

Combining the DOS Patterns with KingFisher's "Boolean Expressions" provides
 for a great deal of power. Consider the following example:

Assume you wish to search for a replacement program for Aquarium that was
 NOT written by Olsson (so that you avoid getting Olsson's original Aquarium
 program.) Let's try the following expression:

```
aquarium & ~Olsson
```

Looks fine at first, but what if the description of a program said
 something like:

```
"JellyFish is a replacement for Olsson's Aquarium program. Author: J. Doe"
```

Our little expression didn't work too well, did it?

So let's alter it with the help of DOS regular expressions:

```
aquarium & "~(#?author:#?olsson#?)"
```

Can you figure this one out? It's a bit messy, but essentially it states
 that we do not want (see the ~ symbol?) what is enclosed in the ()'s, which
 is a regular expression stating anything that contains the word "author"
 FOLLOWED by "olsson".

This brings up an important point: KingFisher's boolean expressions are not
 position dependent, while a regular expression's components are! This
 means that if you searched for

aquarium | olsson

the word "olsson" may occur before or after "aquarium."

1.90 Search Setup, Trim Blanks

When this option is selected, and a simple search (as opposed to a boolean expression search) is performed, trailing blanks are removed from the search string. This is a handy feature, because blanks could accidentally be appended to a search string and not be visible in the string gadget.

1.91 Clear output file

Output is always appended to the output file (whose name defaults to "T:KF.output" and can be changed in the KingFisher.prefs file.) The "Clear" gadget erases the output file so that no other text occurs before the one about to be written to the file.

1.92 4 THE DATABASE

KingFisher's database files are by far the most important part of the system. You can more easily obtain a new copy of the KingFisher.prefs file or the index files, but the names and descriptions in the data files are not so easily replaced.

4.1 Recovering the Database

4.2 Volume References

4.3 Technical Background

4.4 Updating your database

1.93 4.1 Recovering the Database

If KingFisher reports ``seek errors'' and/or displays junk instead of clean text, try first to Reindex

the database. This operation can be performed without loss of any data.

``Seek errors`` may also be caused when a corrupted database is reindexed. The problem is usually encountered in the last few lines of a particular database file. Please see section ? for more detail on the format of these files.

Another cause of ``seek errors`` is that KingFisher is unable to locate its datafiles. In that case you should exit the program and locate files that are named something like ``Fish-0001-0501`` --- note especially the numbers, in this case 0001 and 0501, and the disks' volume names where you found these files.

Create or modify a the
KingFisher.prefs
file by running KingFisher and allowing
the program to continue if it finds that it cannot locate the
KingFisher.prefs file. When next you save and/or quit the program,
KingFisher will write all settings to the KingFisher.prefs file, thereby
(re-)creating this file. The file will contain text similar to the
following (you may need to look for it somewhere within the file):

```
-----
FishData=0001,0500,KingFisher-1:
FishData=0501,1100,KingFisher-2:
FishData=1101,1700,KingFisher-3:
FishData=1701,2300,KingFisher-4:
FishData=2301,2900,KingFisher-5:
FishData=2901,3500,KingFisher-6:

FishFileNames=Fish-%04d-%04d

FishIndexFile=KingFisher-1:Fish.Index
NameIndexFile=KingFisher-1:Name.Index
-----
```

Explanation:

The first lines (FishData) describe which range of fish disks are stored on what disk volume. In this case, fish disks 1-500 are stored in some file on volume ``KingFisher-1``.

The second portion (FishFileNames) describes the exact name format of these datafiles. Those familiar with the ``C`` language will have recognized this format already. Suffice to say that this format should contain two (2) items such as ``%04d`` which are later filled in with the actual range of disks. The above entry, thus specifies that fish disks are stored in files that start with ``Fish-`` followed by a 4-digit, 0-padded decimal number, followed by a dash (``-``) followed by another 4-digit 0-padded decimal number.

The last portion (FishIndexFile and NameIndexFile) specify the name of the fish index file, and the name index file. Both files, if missing or corrupted, can be recreated with the ``Reindex`` command.

All other contents of the KingFisher.prefs file are recreated from certain

default settings stored within KingFisher itself.

1.94 4.2 Volume References

KingFisher requires that its KingFisher.prefs file is stored in the current directory, the directory ENV:KingFisher/ or the volumes CONFIGS:, CONFIG:, CNF:, or S:

If it cannot find the file after this, it will ask you to insert a disk with that file. If you tell KingFisher to retry the operation it will start again, beginning with the current directory. Canceling the operation causes KingFisher to use default values, which may or may not work for you, depending on how closely your setup matches these defaults.

The file KingFisher.prefs contains references to the database files, which can include volume names. If you wish to start KingFisher from a directory other than where its datafiles are stored, you must modify the KingFisher.prefs file.

Use a text editor to modify this file (make a backup first!) looking for entries such as the following:

```
-----
FishData=0001,0500,
FishData=0501,1100,
...
FishIndexFile=Fish.Index
NameIndexFile=Name.Index
-----
```

Modify these entries by adding volume (or directory names.) Assuming that you have stored KingFisher and its datafiles in the directory ``DH0:KF`` you should modify these entries to look like this:

```
-----
FishData=0001,0500,DH0:KF/
FishData=0501,1100,DH0:KF/
...
FishIndexFile=DH0:KF/Fish.Index
NameIndexFile=DH0:KF/Name.Index
-----
```

Now, if KingFisher is in your execution path, you can run KingFisher from any directory on your harddisk and it will find its datafiles in the proper directory.

1.95 4.3 Technical Background

The following section may be of help understanding the interaction of KingFisher's datafiles. I hope that any additional products, such as AREXX servers using KingFisher's database without the program, benefit from this

section:

Fish.Index

The only ``binary'' (non-text) file used by KingFisher. Each record is 16 bytes in length (this may change in the future) with the following structure:

```

struct FishINDEX {
    USHORT FishDisk; /* fishdisk this contents record is on */
    USHORT TextLength; /* size of the text chunk */
    ULONG OffsetText; /* byte offset into database file */
    ULONG PrevVersion; /* previous version of this program */
    ULONG NextVersion; /* next version of this program */
};

```

Worth noting is that the ``OffsetText'' is relative to the beginning of the file in which a particular fish is stored, and NOT with respect to the first fish. Rearranging the way the fish disks are distributed in various files requires a ``Reindex'' operation, therefore.

Also to be noted is that the ``PrevVersion'' and ``NextVersion'' are fish numbers, or (to be more precise) indices into the array of these index records. By these definitions, it will become obvious that KingFisher can work with up to 65535 Fish Disks and 4294967295 fish in the database.

It should be understood that since KingFisher's database is textual, there is no need to use the Fish Index except when it is necessary to determine the exact position (and size) of a particular fish therein.

Searching may be achieved with greater efficiency by loading one line from the database at a time, treating them as pairs (name plus description) except when the name proves to be blank in which case a disk boundary has been reached.

KingFisher loads this index into memory. This file is rebuilt by KingFisher's

```

Reindex
operation.

```

Name.Index

This is only a text file, one line per program name. It is effectively an extract of the database itself, collecting only the names of programs. The names of about 800 fish disks require less than 32K of storage.

KingFisher loads this index into memory. This file is rebuilt by KingFisher's

```

Reindex
operation.

```

Fish-xxxx-yyyy

These files represent the actual database contents, and are the most valuable portion of the entire database. Through a simple and storage efficient methodology, these files contain all the information needed for the

Reindex
command to rebuild the quick-
access index files.

These files contain two lines per database entry (the first is the program title, the second (long!) contains the description. A completely blank line is placed after each disk, which allows the

Reindex
command to determine the divisions between disks.

The input datafiles

Perhaps one of KingFisher's greatest strengths lies in its ability to parse an input file in such a way that you need neither to edit out extraneous information, nor worry about duplicate disks, or missing disks. The parser recognizes Usenet news and electronic mail headers, and can recognize and skip many signatures that are commonly appended to such messages.

Each disk in the input begins with one of the strings

```
CONTENTS OF DISK ...  
or  
THIS IS DISK ...  
of  
NEW/AMIGALIBDISKS/DISK...
```

followed by a second line that is ignored. If this second line is blank and then immediately followed by a description (see next) then KingFisher needs to backtrack for this, which somewhat slows the adding process, is, however, a case that has been foreseen.

The descriptions are always separated by blank lines. Again, KingFisher is usually capable of dealing with blank lines that contain spaces and tabs (i.e. they LOOK blank, but technically aren't) and with missing blank-line separators between individual "fish."

A description always begins with a word in column 1, which is then followed by the first line of the program's description. This description begins either at or after column 16 (first blank after column 16, if the program name exceeds 16 characters in length) or after the first tab character that follows the program title.

Confused? Check out this example:

```
Hello World This is your typical introduction to a programming  
language type demo. Author: everyone and nobody.
```

This example could also have been stored like this:

```
Hello World This is your typical introduction to a programming  
language type demo. Author:
```

everyone
and
nobody.

Do you notice the indentation for the description? This is necessary because any non-blank text in column 1 is taken by KingFisher to represent something that is not part of descriptions.

1.96 4.4 Updating your database

One of KingFisher's greatest strengths lies in its ability to ←
parse an
input file to extract new program descriptions. A central maintainer, such
as was my self-appointed role with respect to Aquarium, is no longer
required or even desired.

There are numerous sources for the descriptions you can use to keep your
KingFisher well-fed and up to date. The "Contents" files from the fish
disks themselves are good sources, as are the listings that Fred himself
posts to the Usenet newsgroup comp.sys.amiga.announce; Aminet sites also
carry the listings (often named something "fish/docs/Fish841-850.txt".)

If you are not sure if KingFisher will accept a file with descriptions or
not, you could simply "feed" it the file and see what happens. If the
operation fails, or you end up with garbage in your database, you can
easily

delete
the junk from the database again and try another method.

1.97 5 QUESTIONS & ANSWERS

This chapter tries to address what problems you may encounter. ←
My hope is

that it helps you solve such problems or gain an understanding for what may
be happening if things don't go by plan. If this section fails to be
helpful, please send me mail (electronic or otherwise) describing as best
you can what your

problem
is. I make every effort to respond to everyone

who writes me!

Q: KingFisher keeps claiming that the database is empty and advises me
to use the

Add Fish
command. Where are my fish?

A: KingFisher cannot find the ``Fish.index'' file or the file is
empty. KingFisher reads the

KingFisher.prefs
file to determine

exactly where its datafiles are located. The floppy disk
configuration file that KingFisher ships with uses volume names

such as ``KingFisher-1:'' but the harddisk configuration specifies no path at all. In that case you must start KingFisher from the directory where its datafiles are stored.)

A way around this is to modify the KingFisher.prefs file. Please see section ? for a description of what needs to be changed.

Q: When I quit, KingFisher says ``Unable to update index (...)''

A: KingFisher saves a modified Index either on request (via the menu item

Save Settings

) or when you quit the program. If the proper

disk is not made available, the program cannot update the index.

If you have added fish to the database, you should run KingFisher again and use the

Reindex

command to recover the index. Either

that, or use the

Add Fish

command again.

If the filename in the parentheses of the complaint is null, such as in: ``Unable to update index ()'' then the KingFisher.prefs file is not providing the name of the index file. Edit the KingFisher.prefs file and look for (and if missing, add) lines such as the following:

```
FishIndexFile=Fish.Index
```

```
NameIndexFile=Name.Index
```

Ideally you should store the Name.Index file in the same place where the file Fish.Index is stored.

Q: KingFisher complains ``Unable to expand Fish Index Array''

A: When you add a lot of fish to the database, KingFisher may need to expand in size the in-memory copy of the

Fish.index

file. If you

do not have enough memory (perhaps your memory is fragmented) then KingFisher will complain. You should quit KingFisher to save everything and then restart the program so that KingFisher can allocate a new index with a small extra margin that may be enough to complete the operation.

Q: KingFisher complains about some fish having been truncated to 4096.

A: At the time of this writing, every fish description has to fit

within a buffer of just over 4000 bytes. It is anticipated that this is more than enough in the future. This message, therefore, indicates a problem with the

Fish.index

file, which stores the

length of each fish entry (along with other information.) If the index is corrupted, KingFisher may read misleading information and get confused.

Reindex

your database.

Q: KingFisher complains about a ``Seek error ...''

A: KingFisher is unable to read an entry from the database which the KingFisher.prefs file and the

Fish.index

together claim should be

there. The problem can be any of the following:

- o ``KingFisher'' does not specify volume names for the data files (see section ?) and KingFisher tries to open these files in the wrong places (creating empty files in the process, which you may need to clean up.)

- o ``Fish.index'' has become corrupted. You may need to use

Reindex

- o You may have shortened one of the datafiles by editing away ←
text.

Do not forget to

Reindex

after every such operation!

- o If this occurs during a 'Reindex' operation, the most likely cause of this problem is "junk" in one of the datafiles. If the database was trimmed with a

Delete Fish

command and text in

the form of new fish is later added again but does not completely overwrite the previously deleted text, then the end of the database contains a partial description which can confuse KingFisher later. This problem can occur at the end of any of the database files and requires manual intervention.

Q: While adding fish, KingFisher complains about some disk not being in a defined partition. What does that mean?

A: KingFisher needs to know what range of disks is stored where. You tried to add a disk that KingFisher did not know where to store. See section ? for descriptions of the values that you need to add to the KingFisher.prefs file.

Q: While adding fish, KingFisher complains that I first have to add some other disks.

A: Adding the current set of disks would leave a ``hole'' in the database. KingFisher won't let you do that. If the range of disks is relatively small, you should look for the descriptions and add these first. If the range of disks represents a major portion of the database, you may want to download version 1.0 of KingFisher. In KingFisher's original distribution, the database had a small problem which a program named ``FixKFData'' would fix. That program was distributed with version 1.1 and version 1.11 -- If you use version 1.11 from a fish disk, then you have a fully functional program.

Q: KingFisher refuses to let me use the Name Index. After adding fish, and after Reindexing, it complains about a mismatch in the indices.

A: The database that was distributed with versions before 1.11 had a formatting problem. Several later versions distributed a program by the name ``FixKFData'' which you should run on the Fish-0501-1100 file, or one which contains the Fish Disk 522 which is where the problem is actually occurring. After this operation, use the Reindex command again.

Q: When trying to search, KingFisher complains: ``String needs terminating with 'x'' (or some other character instead of x.)

A: Strings must be terminated with the same type of quote that they were begun with. Please re-read the entry on
Boolean Expressions

Q: When trying to search, KingFisher complains: ``Pattern ←
too

complex!''

A: Please send an EXACT copy of the string you are trying to use to

the author
along with information on what version of KingFisher
you are using.

Q: When trying to search, KingFisher complains: ``String space exhausted!''

A: Please send an EXACT copy of the string you are trying to use to the author along with information on what version of KingFisher you are using.

Q: When trying to search, KingFisher complains: ``Unbalanced parentheses!``

A: For every open parentheses in the search string you need to provide a matching one that closes the sub-expression. Parentheses inside any quoted string does not count.

Q: When trying to search, KingFisher complains: ``Error in expression!``

A: KingFisher attempted to evaluate the boolean expression and found with it a problem. You should check the syntax of your expression (see the entry

Boolean expressions
for a small tutorial).

If you are using a SIMPLE search string that contains symbols such as quotes (' or '') or boolean operators (&|~! or parentheses such as (or) then turn off the "

Auto Mode

Detection" in the ``Option`` menu's ``Search Setup`` submenu: ←
the

auto detection works by compiling the expression into tokens without checking syntax; it can, therefore, be fooled. Make sure that you also select

Simple Mode

from the same submenu!

Q: I'm trying to look for the string backup & restore in the database, but I notice that KingFisher's search window is interpreting this as an expression. Is KingFisher treating this as two strings and an AND operation?

A: Correct. If you want to have KingFisher treat an entry as a single string despite the fact that it contains special symbols such as &, |, ~, or parentheses, then there are two things you can do:

You can put quotes around the entire string so that it reads 'backup & restore' -- KingFisher will continue to regard this as an expression, but treat it as a single string, rather than two strings connected with an AND operator.

Another method is to turn off KingFisher's "

Auto Mode

Detection" and select

Simple Mode

(both options are found in the

``Search Setup`` submenu of the Options menu.)

Q: I'm upgrading from another Contents Utility (i.e. Aquarium, NewAqua, or FishCat) -- can I use my existing database for KingFisher?

A: No. KingFisher's database format is not directly compatible with the other utilities. It is a relatively simple text file that uses an index to locate specific entries within the file. The other programs use a different database format.

Q: KingFisher takes forever to load. What's happening?

A: Either you're running KingFisher from a floppy drive (just a joke!) or you have selected a scalable font and diskfont.library is busy computing an image for this font.

Q: I'm using a large (scalable) font for the contents display, but there seems to be far too much blank space between the lines. How can I reduce the amount of space?

A: Add a line such as ``text-spacing=-1`` to the KingFisher.prefs file. This tells KingFisher to reduce the spacing between lines by one. Play with that value until things look good.

Q: Why can't KingFisher scroll the text smoothly, one pixel line at a time?

A: Add a line such as ``scroll-speed=1`` to the KingFisher.prefs file. KingFisher scrolls the display by one line of text usually unless you tell it specifically how many pixel lines it should scroll.

Q: Can I speed up the process of adding fish? There seems to be an awful lot of disk activity.

A: Look for the entry ``AddFishWorkFile`` in the KingFisher.prefs file. This entry defaults to a value of ``t:KF.work`` -- If T: is assigned to a physical disk, then KingFisher will write fish to this file before adding them to the database. You can change this value to something like ``RAM:KF.work`` which will be faster,

although some fragmentation of your memory may occur.

Q: What exactly is this ``KF.work`` file used for? How big will it grow?

A: The file will contain the contents descriptions of one complete disk. This may be a single program, or it may be several programs. It is, therefore, difficult to give an exact measure, but as a rule of thumb, it will seldom exceed 5 to 10K of space.

Q: How much memory will KingFisher use if I change the ``AddFishWorkFile`` to use the RAM: device?

A: KingFisher adds the contents of an entire ``Fish`` disk to this file before transferring this information to the actual database. This assures that the database does not contain only parts of new disks due to possible problems. There is no absolute answer, but 10K of memory should cover all but the most extreme cases.

Q: How can I fix misspellings in the database?

A: If a correction involves a simple character substitution, you can use a tool such as ``NewZap.`` If you need to insert or delete even a single character, then you must use a text editor that can work with long lines and afterwards use the
Reindex
command.

Q: How can I rearrange the way the KingFisher database is split up?

A: While KingFisher may offer significant assistance with this process in a future version, for now you'll either have to live with the current setup or follow these instructions WITH A BACKUP COPY OF THE DATABASE:

1. Use a text editor that can work with long lines (GNUemacs is a good choice.) Cut from one file and insert at the appropriate place in the other one ENTIRE DISK'S FISH DESCRIPTIONS. Be sure to include the extra blank line at the end of the fish disk: The
Reindex
operation uses these to determine where one disk ends and the next begins.

2. The idea is that each fish in the database occupies exactly two
-

lines: the first line for the name, the second (long one!) is for the description. Exactly one empty line is inserted after every fish disk for the benefit of the ``Reindex`` command. Technically it is not necessary to add such an empty line to the LAST DISK in a datafile: the end of file signals the same condition: the end of a disk.

3. Now change the KingFisher.prefs file's ``FishData`` values to reflect the new partitioning of your file. Be sure to RENAME the files, also, to reflect their new contents.
4. Delete the ``Fish.index`` file and run KingFisher. KingFisher will complain that the database is empty. Judging by the missing ``Fish.index`` file, this is quite true. Use the ``Reindex`` command to scan the existing datafiles and rebuild the index. This step is necessary to rebuild file-relative offsets.

Q: I don't like it when KingFisher saves the KingFisher.prefs file to
S: --- What can I do?

A: Store the KingFisher.prefs file in the same directory from where you start KingFisher. Either that, or create an assignment for a logical volume (named CONFIGS: CONFIG: or CNF:) where you wish KingFisher to load from and save to.

If you are working with KingFisher from the Workbench, you can add a

```
Tooltype
to KingFisher's icon that specifies the
location of KingFisher's configuration file.
```

Q: I'm worried about damaging my database. How paranoid am I?

A: I won't say that you're paranoid or not, but you should know that KingFisher modifies the database files ONLY when you use the command

```
Add Fish
. If KingFisher displays information in a
garbled form or fails to operate properly in other respects, the
problem is most likely partitioning information in the
KingFisher.prefs file, or a corrupted Fish.Index file. After
verifying that KingFisher's .prefs file is not at fault, erase the
Fish.Index, restart the program (KingFisher will claim the database
is empty) and begin a
Reindex
operation.
```

If you work with KingFisher on floppy disks, it is important that you give KingFisher the disk volumes it asks for. KingFisher is not guaranteed to handle such failures gracefully. You will not lose information; however, you may temporarily lose access to your database and it may seem that the database is

damaged when in truth it is not.

Q: How can I get a list of all programs in the database that Matt Dillon wrote or had a hand in?

A: Use the

```
Output, Filter (search)
command. Make sure you select
the proper ``Output Format`` from the Options menu, including
output to the printer or a file. Then type ``Matt Dillon`` into
one of the search string gadgets, click on the check box next to
that gadget to make that gadget the current search gadgets, and
then select the command
```

```
Filter (search)
-- the output is all
programs whose text contains ``Matt Dillon`` ... by any standard,
he's got a hell of a lot of programs on the Fish Disks!
```

Q: How can I find out more about the
Display Database
and the codes
that KingFisher uses for different display modes?

A: If you have access to a set of 2.0 include files, you can find out a complete set of these values in the file graphics/displayinfo.h Other programs may also use these display info codes and have a more complete list. Keep in mind that KingFisher needs at least a 640x200 display and will not run with anything less.

Q: Why can't I just specify ``A2024 (15 Hz)`` for the display mode in the KingFisher.prefs file and have KingFisher figure out what code to use?

A: Perhaps I can implement that in the future, but there may be language differences which I don't know how to bridge yet, along with problems involving proper formatting of these modes, etc. At the same time I am not familiar enough with the Display Database yet, to make better use of it. Give me time!

Q: KingFisher's display gets very weird when I run it in Productivity mode or in A2024 display mode. What's going on?

A: By default, KingFisher opens a 3 bitplane (8 color) screen. This is not compatible with some display modes. While KingFisher uses

the 2.0 Display Database for SOME things, it is not yet capable of making inquiries to determine display limitations.

For now you can change the value of ``display-depth`` in the KingFisher.prefs file to suit your needs. A value ranging from 1 to 3 is acceptable (producing 2, 4, or 8 colors.)

Q: I wish KingFisher would be compatible with Snap so I could cut out some text from the description and paste it into other applications!

A: As long as you use a non-proportional font, Snap can do what you need it to do! The easiest starting point is to use a font such as topaz or pearl. Since KingFisher's line spacing may differ from Snap's expectations, you may need to Snap from two separate lines to allow Snap to understand that the spacing is different before you can snap multiple lines.

Q: Why do menu shortcuts not work? The screen flashes (beeps) when I use them.

A: Under Kickstart 2.x Intuition no longer processes menu shortcuts while a string gadget is active. A program named ``PatchOS`` on Fish Disk 706 provides some relief to this problem, although it only works properly with commands and not options.

Q: Is searching with
boolean expressions
slower than a ``simple``
search?

A: Yes. The greatest amount of work is performed before the search actually takes off, but even so, the evaluation of the expression (as well as searching for multiple strings) requires some time. The overhead can be quite minimal (perhaps 10%) or drastically slow down the search as the expression grows in complexity.

Q: Would it help speed up boolean expression searches if certain sub-expressions were arranged to cause short-circuit evaluation?

A: No. KingFisher's boolean expression search does not use short-circuit evaluation, even though such an optimization might well speed up the search operation. Building a parse tree requires too much overhead, so KingFisher uses a simple stack-based algorithm instead.

Q: Is there ANY way to make
boolean expressions
evaluate faster?

A: It depends on the expression. If you are familiar with boolean logic you could rewrite the expression to use fewer operands. The following logical forms use KingFisher's syntax, where ``s`` stands for any character string (quoted or not):

```
(~s & ~s) ==> ~(s | s)
(~s | ~s) ==> ~(s & s)
```

In both cases you have shortened the expression by a single negation operator, which may not seem much, but could (conceivably) speed up the search by a few percent again.

Remember, though, that it may be far preferable to use READABLE expressions and suffer a slight speed penalty, rather than construct expressions that are efficiently reduced for the machine, but make no sense to you!

Q: When I quit KingFisher, the graphics go away, but then the screen flashes and KingFisher doesn't finish closing up.

A: Some other program has a window open on KingFisher's screen. While KingFisher does not yet act as a public screen, it is still possible to open windows on KingFisher's screen. The Color tool is one of these.

Close all windows on KingFisher's screen. When the last window is closed, KingFisher will automatically complete the shutdown procedure and exit.

1.98 6 KingFisher and ARexx

KingFisher's ARexx port is named KINGFISHER. When you address ←
KingFisher
via ARexx, you should use the command

```
ADDRESS KINGFISHER1
```

for the first invocation of KingFisher and the port KINGFISHER2 for the second copy of the program, etc. Regardless of this unique port number, the error result string will always be named KINGFISHER.LASTERROR

It is important to remember that the order of parameters are important to KingFisher. While other programs may offer significantly more flexibility, KingFisher relies on the order presented in the commands detailed below.

NOTE: Contrary to earlier promises, the ARexx port is not monitored while KingFisher is iconified or when any requester or window (with the

exception of AmigaGuide) is open on KingFisher's screen! If response is positive and requests numerous enough, I may provide enhancement in the future.

NOTE: Numerous portions of KingFisher now return failure codes when invoked through the ARexx port rather than causing requesters to pop up. Most notably, the Search and AddFish functions have been so treated. Now as before, KingFisher will request that you insert a floppy disk if you request a disk that is not on a mounted volume. Keep this in mind especially for unattended (BBS) operations!

KingFisher uses three types of return codes (in the RC variable):

- 0 Success
RESULT contains returned information (numeric or string.)
- 1 Minor failure (no action required)
KINGFISHER.LASTEROR contains detail; it is not necessary to display this message, although it might be helpful.
Example: search resulted in no more fish, or
'moveto next version' command had no later version.
- 5 Unrecognized or improper command
KINGFISHER.LASTEROR contains usage information or related message. This message should always be displayed to the user.
- 10 Error (major failure)
KINGFISHER.LASTEROR contains further information. This information should always be displayed to the user, as it can hint at a serious problem.

A failure code of level 20 is not generated by KingFisher's ARexx support code.

The following commands are understood by ARexx. For more detailed explanations and examples, please select from the following list:

- ADDFISH
Add fish from file
- HELP
Obtain usage for ARexx commands
- GET
Get (retrieve) a limited set of settings.
- INHIBIT
Prevent a command functioning
- MOVETO
Move around in the database
- QUIT
Quit KingFisher

```
RESET
  Reset search parameters et al.

SCREEN
  Control both WB and KF screen positions.

SEARCH
  Configure, start or continue search

SET
  Set (change) a limited number of settings.

SHOW
  Retrieve displayable text

TELL
  Retrieve position or database stats

VERSION
  Obtain KingFisher version info
```

An

```
    example ARexx program
    is included in the original archive for your
learning pleasure :) and reproduced within this documentation.
```

1.99 ARexx Example Program

```
/* ARexx example for KingFisher:
 *
 * Lists all programs in the database and
 * indicates those that have been supplanted
 * by newer versions.
 *
 * All commands sent to KingFisher are shown in
 * UPPERCASE to more clearly indicate what is
 * going on. KingFisher does not, of course,
 * care about case, unless, perhaps, during a
 * search operation (which we're not using here.)
 *
 * BUGS: we are checking the return code only where
 *       we expect an error condition to occur.
 *       This means that we won't handle any sort
 *       of critical errors, such as an empty data-
 *       base, or a missing disk, or other problems.
 *
 * Written by: Udo Schuermann
 */
options results
address kingfisher1

VERSION
say "Using "result
say ""
```

```

say "The following is a list of all programs in the database,"
say "and information on the disk where their latest revisions"
say "are to be found:"
say ""

totalupdates=0
totalrevisions=0
ok = 1
MOVETO FIRST FISH
do while rc = 0
  /* save current position in the database */
  TELL FISH
  curfish = result
  TELL DISK
  curdisk = result

  /* now see if we can skip forward on the version link */
  MOVETO NEXT VERSION
  if rc = 0 then do
    totalupdates = totalupdates + 1
    /* follow all version links and count them */
    do while rc = 0
      totalrevisions = totalrevisions + 1
      MOVETO NEXT VERSION
    end
    TELL DISK
    msg = ' (Latest updated version is on disk #'||result|')'

    /* Return to the place from where we followed the link */
    MOVETO FISH CURFISH
  end
  else
    msg = '';

  /* now get only the title of the description */
  SHOW TITLE
  say 'Disk#'curdisk': 'result||msg
  MOVETO NEXT FISH
end

say ""
say "Statistics:"
TELL FISHCOUNT
total = result
say "Total fish in database: "total
say "Total revised fish      : "totalupdates" ("||(totalupdates*100)%total||"%"
say "Total of all revisions: "totalrevisions" ("||(totalrevisions*100)%total||"%"

```

1.100 ARexx: ADDFISH

SYNTAX

ADDFISH filename

SYNOPSIS

Adds new fish from the indicated file to the database. Notice that no requesters are produced, although the usual error output listing will continue to operate.

It is strongly recommended that this command be inhibited during unattended operation, such as where KingFisher serves the users of a Bulletin Board System (BBS.)

EXAMPLE

```
/* add multiple files to kingfisher */
options results
address kingfisher1

ADDFISH 'Disk-841:Contents.841'
ADDFISH 'Disk-842:Contents.842'
ADDFISH 'Disk-843:Contents.843'
ADDFISH 'Disk-844:Contents.844'
ADDFISH 'Disk-845:Contents.845'
ADDFISH 'Disk-846:Contents.846'
ADDFISH 'Disk-847:Contents.847'
ADDFISH 'Disk-848:Contents.848'
ADDFISH 'Disk-849:Contents.849'
ADDFISH 'Disk-850:Contents.850'
exit
```

1.101 ARexx: GET

SYNTAX

GET option

SYNOPSIS

Retrieves the definition of one of the function keys, the export file, the reporting file that is used during AddFish operations, the workfile used during AddFish operations, or one of the six currently visible search strings.

The 'option' and 'value' portion of the parameter are separated by an equal sign (=) and NO blank spaces!

The following are understood for the 'option' portion:

OPTION	Meaning
~~~~~	~~~~~
fkey-0	Function Key F1
fkey-1	Function Key F2

```
fkey-2    Function Key F3
...      ...
fkey-9    Function Key F4
```

export-file File used when output goes to a file instead of to a printer.

addfishreport File (or console window!) to which messages of progress and other information are sent during an AddFish operation.

addfishworkfile The temporary workfile that KingFisher uses during the AddFish operation. For optimum speed this file is best located on the RAM: disk. It is unlikely to ever grow larger than 5 to 10K, as it never holds more information than all the descriptions of a single fish disk.

```
search-text-1 First search string gadget
search-text-2 Second search string gadget
...      ...
search-text-6 Sixth search string gadget
```

#### EXAMPLE

```
/* set the value of the first blank (unused) function key to a value */
options results
address kingfisher1

i=0
do while (i<10)
  get 'fkey-'||i
  if length(result) == 0 then
    break; /* stop the loop! */
  i=i+1
end
if(i<10) then do
  set 'fkey-'||i=c:execute kf:example-scripts/download-current-fish.kfs
  say "Function key #"||i+1||" will now download the current fish!"
end
```

## 1.102 ARexx: HELP

### SYNTAX

HELP

### SYNOPSIS

Produces a brief listing of commands, their parameters, and explanation. This listing falls far short of proper documentation, but can serve in an emergency (such as when documentation is unavailable.)

## 1.103 ARexx: INHIBIT

### SYNTAX

INHIBIT command

### SYNOPSIS

This command will cause a command to be inhibited, i.e. disabled until KingFisher is re-started. The use of this feature is to inhibit the functionality of potentially damaging or undesirable commands during unattended operation, such as when KingFisher serves the users of a Bulletin Board System (BBS.)

It is recommended that the following commands be inhibited: QUIT, ADDFISH, INHIBIT, SET, and SCREEN. Notice that the INHIBIT command itself must be inhibited as the last one, since it will thereafter become inhibited (and therefore unavailable) itself!

### EXAMPLE

```
/* prepare kingfisher for remote operation */
address kingfisher1

INHIBIT QUIT
INHIBIT ADDFISH
INHIBIT SET
INHIBIT SCREEN
INHIBIT INHIBIT
exit
```

## 1.104 ARexx: MOVETO

### SYNTAX

MOVETO DISK n  
MOVETO FISH n

MOVETO FIRST FISH  
MOVETO LAST FISH  
MOVETO NEXT FISH  
MOVETO PREV FISH

MOVETO NEXT DISK  
MOVETO PREV DISK

MOVETO NEXT VERSION  
MOVETO PREV VERSION

### SYNOPSIS

---

Move to a different entry in the database. The "MOVETO NEXT ..." and "MOVETO PREV ..." entries (for DISK, VERSION, and FISH) correspond to the buttons in KingFisher's GUI (Graphic User Interface.) Likewise, the "MOVETO FIRST FISH" and "MOVETO LAST FISH" correspond to the pair of small buttons next to the disk number gadget that select the very first and very last fish in the database.

"MOVETO DISK n" allows you to move to the first entry on a particular disk, much like entering a disk number in the disk number gadget. The disk number must not be greater than the highest disk in the database, otherwise KingFisher will issue an error.

"MOVETO FISH n" is a special function not available through the graphic user interface. It allows you to make any of some four-thousand database records the current one. The record number starts with 1 and counts upwards from there. It corresponds directly to the value returned by the ARExx

```
TELL FISH
command.
```

This command will also execute a

```
SHOW
command for you, meaning that
```

you can immediately display the ARExx RESULT variable to produce the title and description for the new entry.

#### EXAMPLE

```
/* display all versions of a program (see the
SEARCH
command, too)*/
options results
address kingfisher1

reset
search direction forward /* default direction is backward */
MOVETO FIRST FISH /* start from first fish in database */
found = 0
search for sysinfo
if rc > 0
then say kingfisher.lasterror
else do
ok = 1
while ok do
found = found + 1
tell disk;
say "DISK #"result":"
show /* get description again; TELL altered it */
say result;
say ""
MOVETO NEXT VERSION /* try next version */
if rc > 0
then ok = 0 /* no next version! */
end /* end while */
say "located "found" versions of this program."
end /* end if/else */
```



exit

## 1.105 ARexx: QUIT

SYNTAX

QUIT

SYNOPSIS

Shut down KingFisher. This is the exact equivalent to selecting  
QUIT  
from the menu.

## 1.106 ARexx: RESET

SYNTAX

RESET

SYNOPSIS

Resets various internal states, especially with respect to the search operations. More explicitly, this command moves to the end of the database, selects title-only search, backwards search direction, and causes case to be ignored during searches.

Recommended when a user first gains access to the database to search for information. The simplest search operation, therefore might be

```
> search for sysinfo
```

Which will look for the most recent version of a program by this title (or with this string as part of the title.)

EXAMPLE

```
/* search for multiple programs, each the most recent version */  
options results  
address kingfisher1
```

```
RESET; search for sysinfo  
if rc > 0 then do  
  tell disk; say "DISK #"disk": "  
  show; say result; say "  
end
```

```
RESET; search for '"game & (graph | sound)'"  
if rc > 0 then do
```

```
tell disk;    say "DISK #"disk":"  
show;        say result; say ""  
end
```

## 1.107 ARexx: SEARCH

### SYNTAX

```
SEARCH MODE TITLE  
SEARCH MODE DESCRIPTION  
  
SEARCH DIRECTION FORWARD  
SEARCH DIRECTION BACKWARD  
  
SEARCH CASE SENSITIVE  
SEARCH CASE IGNORED  
  
SEARCH RANGE ACTIVE  
SEARCH RANGE INACTIVE  
SEARCH RANGE low-high  
  
SEARCH FOR expression  
SEARCH AGAIN
```

### SYNOPSIS

Configures and begins a search operation. These changes will remain in effect until changed again or reset by the

```
RESET  
command:
```

MODE alters the search mode with respect to which portion of the database is searched, the title only, or the entire database entry (title and description.)

NOTE: For TITLE searches, you should never supply more than one set of quotes, unless quotes happen to be part of the string you wish to search for. This is because searches on titles are NEVER

```
boolean expression  
searches!
```

DIRECTION alters the search direction, i.e. if the next search operation will proceed from the current position forward or backward. It is worth noting here that the current entry will actually never be searched in order to prevent the search getting "stuck" on the current entry. This behavior may undergo alteration in the future!

CASE alters the sensitivity of the search operation to differences in upper and lower case characters. At present, national symbols are not guaranteed to be properly recognized with respect to their upper and lower case equivalents. This may be guaranteed in a future version.

---

RANGE alters the restriction on the region within which a search is confined. The 'low' and 'high' values must be disk numbers. No spaces may be entered in this range. Instead of the dividing dash (-) KingFisher will also understand a comma (,). The limitation is not activated until the "SEARCH RANGE ACTIVE" command is received. The command "SEARCH RANGE INACTIVE" command will re-enable the default use of the entire database for searches.

NOTE: Enclose the range of disks in quotes, otherwise ARexx will interpret the values as a mathematical operation (subtraction.) Another way to express the range (for ease of use) is to use a COMMA instead of a dash (SEARCH RANGE low,high) See the SHOW command for exemplification of the difference.

NOTE: defining a range of disks to search does NOT ACTIVATE an INACTIVE range selection: you must ACTIVATE the use of a search range explicitly!

FOR defines an expression for which to search. If this expression is to contain blank spaces, it must be DOUBLE quoted! This is because ARexx removes one set of quotes, passing the remainder to KingFisher. If KingFisher finds blanks within the entry but no boolean operators, it will complain! The expression remains unaltered until changed by a

RESET  
or another "SEARCH FOR"  
command.

This command will immediately begin the search operation, and if it completes successfully, executes a SHOW command (without options) for you, meaning that you can immediately display the result.

AGAIN continues a previous search that was begun with the "SEARCH FOR" command.

This command will immediately begin the search operation, and if it completes successfully, executes a SHOW command for you, meaning that you can immediately display the ARexx RESULT variable.

#### EXAMPLE

```
/* display all programs that offer source code */
options results
address kingfisher1

reset
SEARCH DIRECTION FORWARD /* default direction is backward */
moveto first fish /* start from first fish in database */
found = 0
SEARCH FOR "!'binary only'"
```

```

if rc > 0
  then say kingfisher.lasterror
  else do
    ok = 1
    while ok do
      found = found + 1
      tell disk;      say "DISK #"result":"
      show;          say result; say ""
SEARCH AGAIN
      if rc > 0
        then ok = 0      /* no more! */
      end /* end while */
      say "Located "found" programs that offers source code."
    end /* end if/else */
exit

```

## 1.108 ARexx: SET

### SYNTAX

SET option=value

### SYNOPSIS

Alter the definition of one of the function keys, the export file, the reporting file that is used during AddFish operations, and the workfile used during AddFish operations.

The 'option' and 'value' portion of the parameter are separated by an equal sign (=) and NO blank spaces!

The following are understood for the 'option' portion:

OPTION	Meaning
~~~~~	~~~~~
fkey-0	Function Key F1
fkey-1	Function Key F2
fkey-2	Function Key F3
...	...
fkey-9	Function Key F4

export-file File used when output goes to a file instead of to a printer.

addfishreport File (or console window!) to which messages of progress and other information are sent during an AddFish operation.

addfishworkfile The temporary workfile that KingFisher uses during the AddFish operation. For optimum speed this file is best located on the RAM: disk. It is unlikely to ever grow larger than 5 to 10K, as it never holds more information than all the descriptions of a single fish disk.

EXAMPLE

```
/* change some options */
options results
address kingfisher1

set fkey-9=c:execute kf:example-scripts/download-current-fish.kfs
set export-file=PIPE:kf
set addfishworkfile=RAM:kf.work
set addfishreport=CON:0/0/640/200/KingFisher's AddFish Report/WAIT/CLOSE
```

1.109 ARexx: SHOW

SYNTAX

```
SHOW
SHOW TITLE
SHOW DESCRIPTION
```

SYNOPSIS

Returns information on the current database entry. Without options, this command returns the title followed by the description. The title is separated from the description by a newline symbol. The description is unbroken by formatting symbols, i.e. it consists of one long line that will likely look rather ugly if not processed before being displayed.

The TITLE option returns only the title of the current entry.

The DESCRIPTION entry returns the long, unformatted description, without the title. As with the SHOW command sans any options, the description is entirely unformatted.

The

```
MOVETO
and the
SEARCH
commands issue an implicit SHOW command
(without options.) There is no need to issue a SHOW command after a
MOVETO or SEARCH command unless you wish to restrict the displayed
information to only the title or the description.
```

EXAMPLE

```
/* list the disks and names of all programs mentioning "matt dillon" */
options results
address kingfisher1

reset
moveto disk 1
```

```
search mode description
search direction forward
search for '"matt dillon"' /* note double quotes for DESCRIPTION */
if rc > 0
  then say kingfisher.lasterror
  else do
    ok = 1
    do while ok
      tell disk
      curdisk = result
      SHOW TITLE
      say "DISK #"curdisk": "result
      search again
      if rc > 0 then ok = 0
    end
  end
end

moveto disk 1
search mode title
search for 'matt dillon' /* TITLE must not get double quotes! */
if rc > 0
  then say kingfisher.lasterror
  else do
    ok = 1
    do while ok
      tell disk
      curdisk = result
      SHOW TITLE
      say "DISK #"curdisk": "result
      search again
      if rc > 0 then ok = 0
    end
  end
end
```

1.110 ARexx: TELL

SYNTAX

```
TELL FISH
TELL DISK
```

```
TELL FISHCOUNT
TELL DISKCOUNT
```

SYNOPSIS

Returns the value of the current FISH or DISK in the database. The value returned by the "TELL FISH" command can be saved and later made current again with a similar sequence of commands:

```
/* some code */
options results
address kingfisher1
TELL FISH
```

```
    olddisk = result

/* move around the database */

    moveto disk olddisk

The FISHCOUNT and DISKCOUNT commands return the total number of fish or
disks in the database, respectively. Note that the
    SEARCH
    command
returns the fish number (just like a TELL FISH command would.)
```

EXAMPLE

```
/* introductory stuff */
options results
address kingfisher1

version;          say result
TELL FISHCOUNT; totalfish = result
TELL DISKCOUNT; totaldisk = result
say "We have" totaldisk " disks available with " totalfish " programs!"
```

1.111 ARexx: VERSION

SYNTAX

```
VERSION
```

SYNOPSIS

Returns the current version of KingFisher. Displaying this to a client user (on a Bulletin Board System) is probably a nice touch, as it also advertises the program that is being used to peruse the database and shifts the blame of errors to me rather than on you ;-)

1.112 Arexx: SCREEN

SYNTAX

```
SCREEN KF TOFRONT
SCREEN KF TOBACK

SCREEN WB TOFRONT
SCREEN WB TOBACK
```

SYNOPSIS

Allows an ARexx program to bring forward or drop to the back either the

Workbench screen or KingFisher's screen. This command's greatest use may be in dropping KingFisher to the background once it has been started by an ARexx script on a Bulletin Board System.

1.113 7 MISCELLANEOUS

This chapter contains subjects of more peripheral importance to most users. ↔

- 7.1 The Author
- 7.2 Development Environment
- 7.3 Problem Report Form
- *
- 7.4 History
- 7.5 The Future
- 7.6 Credits

1.114 The Author

In some respects KingFisher may look too good to be my first significant program for the Amiga. This is, however, the case. While I consider myself a seasoned programmer on another platform, I must describe myself as just past the beginner's stage where the Amiga is concerned.

It is for that reason, and no other, that KingFisher does not yet offer gobs of 2.0-features. KingFisher is 1.3 compatible not because I do not wish to implement 2.0-features, but because I am not yet familiar enough with 2.0-features to be able to implement them. Public screen support, for example, takes a ``wee bit'' more than naming the screen with a SA_PubName tag.

I can be reached via electronic mail as walrus@wam.umd.edu, or by standard mail as:

Udo Schuermann
6000 42nd Avenue, Apt. 405
Hyattsville, Maryland 20781

1.115 7.2 Development Environment

KingFisher was developed on an Amiga 2000 with GVP 68030 accelerator under AmigaDOS 2.1 using SAS/C 6.2.

Enforcer 37.28 was used to locate and eliminate problems; SetCPU 1.6 was employed to boot with KS1.3 for compatibility tests.

1.116 7.3 Problem Report Form

Print this form to paper or to a file (use the SYS:System/CMD program) or create some equivalent. This form is important only in that it asks the most important questions. Any equivalent will do fine. I'm no stickler for formalities:

To: Udo Schuermann
6000 42nd Avenue, Apt. 405
Hyattsville, Maryland 20781
(USA)

From: _____

walrus@wam.umd.edu

1. Version of KingFisher that you use. You can obtain this information by issuing the CLI command "version KingFisher"

2. From where did you obtain your version(s) of KingFisher? Some file off a bulletin board system, a public ftp site (which?), a fish disk?

3. Describe your system configuration (Kickstart and Workbench version, hard or floppy drive, amount of memory, and display type, are the most important ones.)

4. Please describe the nature of the problem in the space below. The more information you can provide, the more of a chance I have to duplicate or track down the problem. It helps me very little if you simply say "When I add some fish it doesn't properly add them." See? Not much to go by.

However, if you explained that "given as input the first chapter from the illustrated 'Exotic Fish of the Barrier Reef', KingFisher fails to properly include the description for the voracious Trigger Fish, calling it a Silly Example instead and capitalizing every fifth word", now THEN I have something solid to go by. :)

Seriously, though: some problems, when mysterious to you, are downright impossible to reproduce or figure out on my end, especially if the problem description provides little guidance. Be specific, be verbose, be elaborate. Thanks!

1.117 7.4 History

The following is a revision history of KingFisher, detailing changes and bug fixes with each release and version, in reverse chronological order:

1.40 December 20, 1993

Fixed Rexx "SEARCH RANGE from-to" command (was totally broken!)

Fixed "SEARCH MODE TITLE" command.
Added "New/AmigaLibDisks/Disk" to the recognized keystings that can start a fish disk description.

1.37 September 28, 1993

Fixed flood fill pattern problem with promoted screens, such as with the Picasso II and other cards.

1.36 August 13, 1993

Really fixed the pens now!

1.35 August 11, 1993

Added support for display pens that recognize different display depths, i.e. no accidental use of colors that match background. Thanks to Markus for tracking down the problem and suggesting the proper solution.

1.34 July 25, 1993

Missing libraries generate more proper complaints now. Fixed some possible problem with recognizing the HELP key.

1.33 July 13, 1993

Added WaitBlit() before FreeRaster() to avoid possibly yanking away the planePtr before the blitter is actually done with it. This *might* have been responsible for crashes on some systems.

1.32 June 18, 1993

Added STAYBEHIND commandline switch to prevent the KingFisher screen from popping to the front when starting up. This makes BBS based operation a whole lot friendlier ...

1.31 May 8, 1993

Added ability to drag KingFisher's windows (search, output, reindex, and big-string entry.)

1.30 April 30, 1993

Added 'Search matched' counter to search filtered output.

Added ghosting of movement gadgets to more properly indicate the available choices.

Added GNUemacs style keys to move about and invoke searches.

Added commandline options (template: SETTINGS\K,NOABOUT\S,TINY\S) to enable better unattended operation (SETTINGS, NOABOUT), and prevent KingFisher from allocating extra space for its internal structures in order to cut down on memory fragmentation if many fish are added to the database (TINY.)

Added ability to suppress reading of a SETTINGS file by specifying a tootype of "SETTINGS=" (no quotes, but no filename either!) and the ability to process tootype strings identical to those ordinarily found in the KingFisher.prefs file -- the .prefs file no longer needs to exist and a few modifications can be made to the default config via tooltypes. KingFisher will continue to save/create the .prefs file upon exit, unless the 'Keep Settings' option is OFF. Tooltypes cannot be updated from within the program.

Added F1..F10 function keys that execute external commands, such as ARexx scripts, programs that print the export file to the printer or perform some other action.

Added ARexx port with 13 commands (options provide some 40 variants on these commands.)
Added native AmigaGuide help (point at gadget/menu & press HELP.)
Added a great deal of documentation and cross-referencing in this, the KingFisher.guide file.
Added keyboard control (with qualifiers) to move through database with the cursor keys. With the "Keyboard Control" option ON, the ENTER key will now toggle between deactivating and re-activating the (next) gadget. Cursor keys function for motion only while no text gadget is active.
Added optional wide-search-string-entry requester.
Added support to select color of search-keyword highlight.
Added TAB key support [KS2.0]
Added gadget styling (inverse video) when a gadget is active [KS2.0]
Minor improvement to reindexing to increase likelihood of linking.
Fixed bug that failed to update version links if case sensitivity was selected.
Fixed bug that tried to free the default database partition list on exit if no such list was specified in the .prefs/.config file.
Fixed improper behavior of the checkmarks in the 'Output Format' menu (checkmarks could be unselected, leaving a one-or-the-other option without either choice; rather confusing!)
Fixed bug with search key highlighting that hung the program if the search key was a null string.
Fixed an obscure bug in version linking routine involving pointers to deleted records.

1.2 February 26, 1993 (Note: this version should have been 1.20)

Added case sensitive search functionality.
Added boolean expression search capability with option to disable KingFisher's auto-recognition of such expressions.
KingFisher now understands KS2+ regular expressions involving ?, #?, and other complex patterns enclosed in parentheses. These are recognized as regular expressions ONLY in QUOTED strings within boolean expressions, this distinction providing some flexibility in searching for combinations of literal strings which might otherwise be interpreted by AmigaDOS as regular expressions. (If you think this sentence is too long, just remember that I'm German and could quite comfortably take your breath away with much more complex stuff!) ;->
Search strings can now be 250 characters long instead of 64; this is to accommodate complex boolean expressions.
All of KingFisher's files now have default values.
KingFisher no longer REQUIRES the configuration file to exist.
KingFisher now also tries to load KingFisher.prefs, and also looks in ENV:KingFisher/ for its configuration file (hopefully it will find KingFisher.prefs there.)
KingFisher is now compiled with SASC 6.2
Added keyword highlighting when a search succeeds.

1.15 January 3, 1993

Bug with international symbols (upper half of character set) fixed (the 'char' datatype was treated as a signed value -- oops!)
This bug would cause KingFisher to lock up (hang) when about to display an entry with such symbols in it.
Fixed bug in word-wrap code that could cause a loop with very wide fonts -- now can break an "unbreakable" word to force it into

available space. Honors hyphen and slash for line-breaking. A bug was fixed that would refuse to properly cancel the save operation if KingFisher.config could not be written for one reason or another.

Added FirstFish/LastFish gadgets.

Added ASL Font Requester support for 2.0. Selected font now allows use of styles (bold, italic, underline.)

Added 'Safe Iconify' option to allow turning off the saving of the KingFisher.config file when using the Iconify gadget.

Added variable-width export (as opposed to 72 columns, fixed.)

Added 'Copy from' menu item to reset current setup to the contents of another Fishtank.

Added support for 1 and 2 bitplane screens as part of the effort to allow KingFisher to function on Public Screens. This should (at last!) provide functionality for A2024 and Productivity screens.

1.11 December 20, 1992

Bugs with name index fixed.

Under 2.0 the program now uses the Display Database to build a display, allowing compatibility with A2024 displays and oversized, scrolling screens in the system.

Broke up Alert for Searches into separate alerts for success and failure.

1.1 December 8, 1992

Added name index.

Added capability to collect size information on other system screens to determine screen size for overscan, PAL, and interlace display.

1.0 December 1, 1992

First public release.

1.118 7.5 The Future

Even now I don't consider KingFisher complete. It is a functional and useful program, true, but my list of enhancements contains nearly a dozen items so far, and I receive numerous suggestions all the time. As I find time, I will continue to enhance KingFisher and release new versions. Foremost on my list are the following items:

- o Screen-less KingFisher (i.e. usable on Workbench or on public screens) in a sizable window.
- o AppIcon on Workbench for "Drop-and-Add" functionality of new fish, and the same functionality by making KingFisher's window an AppWindow.
- o Screen mode requester and on-the-fly adjustment of display modes.
- o VersionLink editor to supplement the automatic version linking that occurs during 'Add Fish' and 'Reindex' operations.
- o FishHooks to facilitate hiding and marking of individual fish in the database.
- o Clipboard support for both input and output.
- o More flexible search-result display (dynamically built list from which entries can be selected for viewing.)
- o Better printing/export support.

If you have suggestions I welcome your feedback!

Not all the improvements listed above will make it into a 1.3 compatible version of KingFisher, perhaps none. So long as I need to make no serious sacrifices with regard to implementing 2.0 features, compatibility with 1.3 will not be ignored. This is bound to have its limits, however:

During spring 1994 KingFisher will turn into a Kickstart 2.0+ product, thereby requiring you to be running at least Kickstart 2.04 (V37) and preferably Workbench 2.1 (V38) software. For those of you who are still using Kickstart 1.3 (V34): you are cheating yourself out of a lot of wonderful features, and serious software developers will leave you in the dust. The upgrade is worth it! Do it now!

KingFisher 2.0

KingFisher 2.0 will hopefully be available sometime in Spring 1994 and will offer a number of new features, especially with regard to the functionality of CD-ROM disks. Yes, that means especially Fred's "Fresh Fish" CD-ROM. I will say nothing more at this time about the planned features, except that I hope to please even the most critical user out there.

1.119 7.6 Credits

I would like to express my gratitude and appreciation to the following individuals, who have contributed one way or another to KingFisher:

Peter Cherna, Commodore-Amiga

For his prompt and thorough help obtaining a workaround to a V34 bug in the Request() function. Great support!

Fred Fish

For his many years of service to the Amiga community, and especially for the collection of software that has come to be known as the "Fish Disks." Fred's efforts have set him apart as one of the Amiga Community's most important people.

George Gibeau

For his efforts at beta-testing KingFisher and for making numerous suggestions and for finding a bunch of misspelled words in this .guide file.

Paul Huxham

For information on how to make a window draggable without causing Intuition to render the dragbar and destroying the window contents.

Maurizio Loreti

For the example code of Display Database and Screen Mode requester. I will be using it as part of KingFisher sometime soon.

B. Lennart Olsson

Author of the original Aquarium program which has been an invaluable aid in the past, and whose only true fault was its failure to expand the database beyond a floppy disk.

Don Mac Phee

His initiative, dedication, and skill as a beta tester turned out to be an invaluable aid! Excellent work!

Kyle Saunders

For his efforts in producing a concordance index for KingFisher that I ended up not using (for several reasons.) Also, Kyle offered many suggestions for KingFisher during the early stages of development, which helped shape the program into what it is today.

Bill Sorenson

For the User Interface Style Guide and the little walrus! :-)

Uwe ``Hoover'' Schürkamp

For the original greyscale and the new color images of the kingfisher that appear in the ``About'' window and the icons, as well as his long-term efforts as a beta-tester that pointed out many dumb things that I (or KS 1.3) did.

Leo L. Schwab

His iconify() function took less than 30 minutes to figure out and get working! Thanks Leo!

Michael Sinz, Commodore-Amiga

For Enforcer 37.28 (a truly wonderful debugging tool!) and his help with regard to using SimpleRexx.

Micheal Zielinski

For providing me with a prototype of this AmigaGuide@ document.

In addition I would like to thank all who gave me encouragement over the years for my efforts to keep the Aquarium database up to date. Not only have your suggestions caused me to streamline the update process but they have also pointed out the widespread usefulness of a tool such as Aquarium, which has, therefore, directly caused me to create KingFisher.

Furthermore, I would like to extend my thanks to all those who have sent me their appreciation in the form of \$10 checks. I was pleasantly surprised by those gestures, and assured that KingFisher's usefulness is more than a passing fad. Thank you for your support!

I would also like to offer my gratitude to the over one hundred individuals from all over the world who sent me messages and mail expressing their liking of KingFisher (or gave suggestions for improvements.) The most enthusiastic of these literally had me dancing for joy! :) It's nice to have my work appreciated. Thank you all very, very much!

I wrote KingFisher for you. I hope that it lives up to its namesake and becomes a worthy successor to Aquarium.

._. Udo Schuermann
() walrus@wam.umd.edu

1.120 Edit Search String

Any one of these six gadgets can provide a string or a boolean expression to serve in a search. Only when the checkmark to the left of one of these strings is marked (the checkbox shown in yellow) will the Select Fish buttons function as search mechanisms.

Search mode is active only when the Select Fish buttons are yellow.

A wider entry gadget is available for each of the six search string gadgets by selecting

Wide Search String
option.

1.121 Select Fishtank

These eight gadgets, numbered "1" through "8" provide a means of saving the current position in the database, including search expressions, and moving around elsewhere without losing that particular place. These could be called "Bookmarks" because one of their uses is to keep a place in order to come back to it later.

Other uses might be to provide a default setting in one, while you use another to work with. Each time you wish to reset your search expressions and your position in the database, you simply copy the contents of the default "Bookmark" (to the currently selected one.)

For this purpose, I suggest the use of "Bookmark" #8 as the default setup, for it is easy to select with one hand, using the menu shortcut rAmiga-8. You thus have six others to use as "permanent" place-holders and one to work with and play with as you please!

1.122 Select the Use of the Search Range

This gadget determines if the Search Range gadgets will be used during a normal search, or if they will be ignored. Only when this gadget is checked, will the search be limited to the disks in the indicated range.

You should be aware of the fact that the Output command "Range" ignores the setting of this gadget as a matter of convenience.

1.123 Search activation and string selection

At most one of these buttons can be selected at one time to determine which

of the six search strings will be used to provide the search expression.

When one of these buttons is selected, the

Select Fish
buttons will turn

the same color (yellow) to indicate that they now function as initiators of a search.

Selecting the checked button again will uncheck it, and turn off the yellow color of the Select Fish buttons, indicating that they have returned to their normal browsing function, moving only one fish at a time forward or backward.

Keyboard shortcut: '1' .. '6'

1.124 Select a Specific Disk

Entering a new disk number in this gadget will immediately position you on the first entry of that disk. This is, perhaps the swiftest way to find a program if you know the correct disk number.

If you select the last disk in the database, and this disk contains more than one entry, you will not in actuality be placed on the last entry of the database, but rather on the first entry of the last disk.

1.125 Select Previous or Next Disk

This gadget allows you to move to the first entry of the previous or next disk in the database. You should understand that this may fail to position you on the very last entry of the database if the last disk contains more than one entry.

Keyboard shortcut: LEFT and RIGHT arrow keys with CTRL held down or 'P' and 'N' (upper case!)

1.126 Select earlier or later Version

This gadget follows the
Version Links
that KingFisher has stored in the

Fish Index
file. If you select one of these gadgets, and nothing seems to happen, then a version link did not exist for the current fish.

A future version of KingFisher will properly enable and disable the Version gadgets, according to the existence of the respective Version Links.

Keyboard shortcut: LEFT and RIGHT arrow keys with ALT held down or ^P and ^N (CTRL-P, C-p, and CTRL-N, C-n)

1.127 Select Previous/Next; Search backward/Forward

Depending on the color of these gadgets, they function either as a ↔
browsing
mechanism (grey), or to initiate a search (yellow.) The gadgets will be
yellow (to search) only if one of the six
Search String Selection
gadgets
is selected.

Keyboard shortcut: LEFT and RIGHT arrow keys or 'p' and 'n' (lower case!)
or to search: '<', ^R, (reverse) or '>', ^S (forward)

NOTE: The keyboard shortcut does not invoke search mode when selected.
This is a perfect way to move through the database one fish at a
time WHILE A SEARCH STRING IS SELECTED AND THE GADGETS THEMSELVES
WOULD INITIATE A SEARCH!

If you wish to search, use '<' or ^R (CTRL-R, C-r) for reverse
search and '>' or ^S (CTRL-S, C-s) for forward search. Note that
one of the six search strings must be active for a search to be able
to work.

1.128 Toggle Search Mode

This cycle gadget can be changed regardless of its color (which ↔
indicates
whether or not the
Select Fish
gadgets initiate a search (yellow) or
merely function to browse through the database one entry at a time (grey.)
Only during a
search
does this gadget function to determine the type of
search to be used:

Magnifying Glass: The swifter search of the two, providing little accuracy
but a great deal of speed. The titles of 4000 fish can
be searched in just a few seconds or less.

Microscope: The slower search of the two, providing a great deal of
accuracy and much flexibility (such as
Boolean Expressions
)

1.129 Modify Search Range of Disks

These two gadgets serve a dual purpose:

1. During a standard search, they will be recognized and used if the

Select Search Range gadget is selected, determining which range of disks is to be searched. The lowest disk number is 1, the highest (at this time) is 9999. Selecting a disk number higher than the latest in the database causes the search to proceed until the end of the database.

One use for these gadgets is, for example, the limiting of a search to the contents that fit on a single floppy disk. This cuts down on the number of disk-swaps that KingFisher requests during a repeated search.

2. When using the

Output, Filtered (search) command, the disks specified in these two gadgets determine the range actually used. It does not matter if the "Select Search Range" gadget is selected or not. This is a convenience override.

1.130 Reveal additional text

If an entry's description is too long to fit in the display panel, this proportional gadget will indicate (by its shrunken size) that there is more text to be revealed by grabbing it with the mouse and dragging it up or down.

Keyboard shortcut: UP and DOWN arrow keys.

1.131 Reveal text line-by-line

These gadgets function merely as conveniences. Instead of grabbing the proportional gadget (the tall bar at the right edge of the display panel) and sliding it up or downwards, you can cause it to move effectively by one line of text at a time with these gadgets.

Keyboard shortcut: UP and DOWN arrow keys.

1.132 Iconify KingFisher

When you select this gadget, KingFisher "iconifies" to an icon on the Workbench screen. This function causes KingFisher to close its display, thereby saving as much as 46K of Chip RAM (as much as 10% of the total available custom-chip accessible memory.)

The icon on the Workbench will offer you the familiar image of KingFisher's namesake bird: a kingfisher. You can drag this icon anywhere on the screen that you like (it acts like a tiny window) and when you double-click on it, KingFisher will attempt to re-open its display.

If KingFisher fails to re-open its display, it will flash all system screens as an indication of error, and then save and exit normally. Such a

situation can occur when too little Chip RAM is available when you attempt to re-open KingFisher, or Chip RAM is too fragmented to offer large enough portions in contiguous chunks.

Keyboard shortcut: ^Z (CTRL-Z, C-z)

1.133 Jump to first/last Fish

These two gadgets allow you to jump quickly to the first or last fish in the database. Notice that this is slightly different from the first and last DISK in the database, as the last disk could have multiple entries (fish) on it. Entering in the gadget next to it the number of that last disk would carry you to the first entry on the last disk, not the last entry on the last disk (i.e the last entry in the database.)

Keyboard shortcut: [and]
