KingFisher2

Udo Schuermann

COLLABORATORS

	TITLE : KingFisher2		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Udo Schuermann	August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

KingFisher2

1.1   KingFisher Release 2

KingFisher Release 2

Copyright © 1992-1995 Udo Schuermann

This document is formatted with AmigaDOS 3.x in mind. Please excuse what display problems you may encounter when running under AmigaDOS 2.x!

Table of Contents

1.2 1 INTRODUCTION

KingFisher Release 2 is a special purpose database tool designed for the storing and retrieving of information about software.

It is fully compatible with Fred Fish's Product-Info Specification v6 (which I helped design.) Ideally, information in the database is broken up into a large number of distinct fields, each of which has a specific meaning and can be examined individually.

The benefit of this storage method is greater flexibility: you can search for software by a specific author, for programs that have at least reached version 2, or for software that is not commercial or has been released after a certain date. Furthermore, how information is formatted can also be specified on a field-by-field basis, and tied to individual databases, providing you with the means to produce custom databases for other applications.

An ARexx interface is also provided, which supports the same formatting capabilities and same advanced search capabilities as the GadTools (GUI) interface. It provides a perfect interface between the databases and any application wishing to retrieve information from them.

KingFisher Release 2 is a complete revision of the original KingFisher. It no longer aimed only at Fred Fish's AmigaLibDisks ("Fish Disks"), but is meant to index also the Aminet, other CD-ROMs, as well as your club's software collection.

Based on Client-Server technology , KingFisher Release 2 allows more than one user to simultaneously access a database without the possibility of data corruption occurring. Even if you have no apparent need for this power, you can make use of it by starting multiple copies of KingFisher Release 2, one to search with, another to browse, a third to print, etc.

1.3 1.1 Components

KingFisher Release 2 consists of multiple components that work together:

KingFisher is the name of the entire package, and also the name of the GadTools (GUI) interface portion.

Behind the scenes operates the KingFisher Database Server (click here for a brief discourse on Client-Server technology) to which KingFisher connects.

Instead of (or in addition to) KingFisher, you could use RexxFisher to access the databases (using ARexx.)

The programming interface to the KingFisher Database Server is available to developers. Click here for information on how to obtain the developer kit.

1.4   1.2 Distribution rights

KingFisher Release 2 is a Shareware product made available in two versions[1]:

(1) A freely distributable version, available in archived form, which may be distributed for a price of no more than $3 (US) per disk or equivalent in foreign currency, and

(2) An unrestricted version, available only to registered users and available only from the author or authorized distribution sites.

Distribution of any portion of KingFisher Release 2 as part of a software or hardware product, including CD-ROM, where KingFisher is stored in ready to use form, such as a CD-ROM index/search tool, is not permitted without a prior license agreement with the author of KingFisher Release 2 for such use.

Such a license agreement shall cost the equivalent of 10 (ten) registered copies of KingFisher Release 2 and grants you the right to duplicate unlimited unregistered copies of KingFisher Release 2 ready to use with all your or your company's hardware or software products.

Registered copies of KingFisher may not ever be distributed by anyone except the author and licensed distribution sites.

_____

[1] A third form are beta test copies of KingFisher, but these are not distributable at all!

1.5   1.3 Registration and Updates

If you have already registered KingFisher, my thanks for your support! If not, please read the following important sections:

The unregistered, freely distributable version of KingFisher Release 2 bears some restrictions. At present, the unregistered version allows only two simultaneous connections to the KFServer (KingFisher Database Server), meaning that you could run two copies of the KingFisher GadTools (GUI) interface, but not a third.

The registered version has no such limitations, making it the search tool of choice for multi-user bulletin board systems!

Please explore the following items:

1.3.1 Why you should register KingFisher

1.3.2 What you get for registering KingFisher

1.3.3 How to register KingFisher

1.3.4 How to obtain updates and developer kits

1.6   1.3.1 Why you should register KingFisher

You received KingFisher for free. You may have downloaded it from Aminet or from a BBS. You may also have paid for a package, such as one of Fred Fish's CD-ROMs, but KingFisher came free with the package. This is because Fred and I have a deal: he gets to distribute KingFisher as part of his great CD-ROMs, and you get a program with both a past and a future.

But what do I, the programmer, get for all the time I spent on KingFisher? I cannot live off thank-you notes (although I appreciate all I get) and CD-ROMs make a poor diet (even when served with wine and cheese ;-) so you can imagine that I am hoping for something more substantial to reward me for my efforts, and I am hoping that you will show your appreciation for my work!

Now, why should you register a program that came to you free? It's simple: You get to use and evaluate it at your own pace, on your own Amiga at home, not some store's machine. You are under no pressure to buy. When you feel that you want to keep the program, you simply send me the registration fee. This registration fee is lower than what a commercial product would cost you, because you are not being charged for dealer overhead, distribution costs, or fancy packaging (every part of KingFisher is fully recyclable and wastes no natural resources :)

The original KingFisher was my return of favors to the Amiga Community. It was free, it was fun, it taught me a lot. But best of all (for you) it was free. KingFisher Release 2 is no longer free. You really should register the program.

If you live in Europe, you can register with someone I know in Germany! In North America, you can register with me directly. Those of you in other parts of the world get to choose the site most convenient to you. I will support you anywhere within the bounds of our Solar System. :) Here is how to register.

1.7   1.3.2 What you get for registering KingFisher

Here is what you get for registering:

× A KFServer that is able to allow an unlimited number of clients to connect. For you BBS operators, that means you can serve dozens, hundreds, thousands(!) of users! (Billions and billions! ;-)

× You can receive Programming Documentation so your own programs can connect to the KingFisher Release 2 Database Server. Are you the author of BBS software? Would you like to write a "Door" or other interface, rather than going through RexxFisher? Do you want to try your hand at writing a (gasp!) better GUI?

× Technical Support via electronic mail and an electronic KingFisher Mailing List. You will also be the first to receive new versions of the included software (even maintenance releases which won't always make it to the general public.)

× My pledge that as long as there is demand for KingFisher, I will continue to support it, regardless of where the Amiga is going.

× Your conscience will let you sleep better at night, knowing that you are really supporting your favorite computer. What good is your Amiga if nobody wrote software for it anymore? Support your software authors today, because tomorrow it may be too late!

As a final word of encouragement, should you feel hesitant about registering KingFisher: With the original KingFisher I hope to have established that quality software and quality support are important to me. You will receive nothing less from me in the future. KingFisher Release 2 is my first attempt to get some return on my investment of time through the good-will of you, the appreciating user. With your support, KingFisher Release 2 will grow far beyond what you see before you today.

Consider your registration a vote. Vote for the Amiga's growth! WITHOUT your support it may never achieve its full potential. Don't take that chance. Register today and support your existing investment! Here is how to register.

1.8   1.3.3 How to register KingFisher

There are two sites to receive your registration fee. If you live anywhere in Europe, you should use the site in Germany. Anywhere else in the world, such as Australia, Japan, or places even further away than that ;-) should register directly with the author, in the United States.

United States:

Send $20 (US) in the form of a personal check drawn on a U.S. bank or money order or similar form of payment to:

Udo Schuermann

7022 Hanover Parkway, Apt. C2

Beltsville, MD 20770

USA

NOTICE: The old address (6000 42nd Avenue, Apt. 405; Hyattsville, MD 20781-1518) remains active until July 15, 1995. Do not use it after that date, as no guarantee can be made that mail forwarding will be reliable ("disgruntled postal workers"...)

Europe:

Send DM30 in the form of a EuroCheque or check drawn on a German bank. Direct bank transfers (Überweisungen) are also available.

Uwe Schürkamp

Jöllenbecker Weg 4

32051 Herford

GERMANY

Members of User Groups: You get a 20% discount when you register and provide the name and address of your user group. This means you get KingFisher Release 2 for either US$16 or DM24. Pass the word!

Sending cash is discouraged for reasons that it's untracable. In the event that it gets lost in the mail, the loss would be yours to carry. If you live in a land far, far away from either Europe or the USA, then your best bet may be to send cash, as many (international) banks do not cater to the exchange of small sums. If you decide to send cash in the mail, make sure that your letter in no way reveals its contents to someone peering at it against the light, or feeling the bank notes through the envelope.

Please do not ask for technical support from any site but the author!

The European registration site is a good friend of mine, but he does not know enough about KingFisher to offer technical assistance. Also, if you absolutely must talk to him and look up his telephone number, please do him and his family the courtesy of calling only during regular daytime hours. Thanks!

1.9   1.3.4 How to obtain updates and developer kits

Updates

Updates to newer versions are available free on Aminet[1] ftp sites (requires Internet access) and, for a nominal fee, at the same addresses as listed here . You may want to specify how you want the update to be processed. Examples are:

the latest version (we keep on record the latest version you received from one of the registration sites), or the next available version within a certain time frame (say four weeks; I might be close to releasing a new version shortly and you might be happy to wait a little longer to get that, instead of the soon-to-be outdated one. We're human beings, we read your request and do our best to listen. If you are in a hurry to get the latest version, please say so.

The cost of obtaining an update:

$5.00 (US) if sent to the United States address (author.)

DM10.00 if sent to the European address.

Updates may also be available electronically on Aminet (and eventually on BBSs.) These will be patch files applicable only to the original and unmodified binaries of specific versions of KingFisher. Keep your original disk!

If you wish to obtain the Developer Package, you must specifically state this when you request an update. There is no additional cost involved for you, but the Developer Package will be included only if you ask for it[2]. Click here for a bit more information.

―――――――

[1]Aminet is a world-wide cooperative system of ftp sites that mirror each other's disk contents to reduce the need for some people to ftp halfway around the planet to retrieve information. KingFisher's update patches are stored on Aminet in the biz/dbase/ directory, in a file named KF-REGS-pch-XX.lha where the XX represents the latest version number (i.e. v2.9 is represented as 29.)

[2]It is not cost-effective at this time to include a 2nd disk for every distributed copy of KingFisher, as only a small number of people actually make use of the developer information. You may obtain the developer kit (225K of information compressed to ~90K) either on disk as part of an upgrade (does not include the "1000 Fish Disks" database, which you have already), or free by electronic means (email: walrus@wam.umd.edu; WWW: http://www.wam.umd.edu/~walrus/)

1.10   1.4 Installation

KingFisher Release 2 is best installed from the distribution disk with the standard Commodore Installer by double-clicking on the Install-KingFisher icon. This is because the full distribution version of KingFisher expands to 2MB on your harddisk, and fitting it all on a single floppy disk requires nearly everything on the disk to be compressed. The Install-KingFisher script knows how to uncompress and install the software for you, and it handles updates as well.

If you choose novice mode, KingFisher will be installed to a drawer named SYS:KingFisher2/, otherwise you get a choice. You must have 2MB of space on the disk where you install the program.

Should something go awry with the installation, it is possible to install KingFisher manually, but this requires some effort and attention to detail, as you'd need to uncompress the files on the disk yourself.

Running KingFisher Release 2 merely requires double clicking on the KingFisher icon, or typing KingFisher at the CLI when your default directory is the installation directory.

Uninstalling the program is even easier than installation.

1.11   1.5 Uninstalling KingFisher

KingFisher is installed into a single directory (and subdirectories of that.) No files are placed elsewhere in the system unless some change to the basic setup is made or you do not have amigaguide.library installed and ask the Install-KingFisher script to install it for you.

This organization makes uninstallation as easy as dragging KingFisher's drawer into the trashcan (using the Workbench), or deleting the subdirectory and all its contents (from the CLI.)

1.12   1.6 Technical support

The quickest way to get technical support is through electronic mail. This requires that you have an account with reliable Internet access. In the past, I've tried to help some people and my replies "bounced." I feel bad that my answers never made it back to them (and they may be waiting yet) but there really is nothing I can do.

Email: walrus@wam.umd.edu

WWW: http://www.wam.umd.edu/~walrus/

Postal mail is another way to get in touch with me. I've tried to be good in the past about replying to all letters, especially those that asked for some sort of response, but I confess that I have a problem allocating space on my desk and as a result things have gotten lost. 8-(

Udo Schuermann

7022 Hanover Parkway, Apt. C2

Beltsville, MD 20770

USA

NOTICE: This postal address is going to change as of August 1, 1995. The new address is not yet known; please inquire by email!

NOTICE: KingFisher's status as a shareware product means that my first priority for technical support is towards registered users. If you are not a registered user, I will still try to help you, but if your problem is complex and requires too much of my time ... well, you can probably imagine how these things go.

1.13   2 CONCEPTS

2.1 2.1 Origin and history

Fred Fish, Aquarium, and KingFisher -- Sound "fishy" to you?

2.2 Client-Server technology

A powerful multiuser database concept

2.3 Search expressions

How to formulate expressions to search for exactly what you want

2.4 Search Sets

Saving and restoring the results of previous searches

2.5 2.5 Custom Formats

Customizing visual, printed, and exported presentation of data

2.6 KFServer

The most important, though least visible, part of KingFisher Release 2

The Product-Info Specification:

Everything you need to properly describe your own projects!

Overriding .prefs defaults with CLI or TOOLTYPE parameters

Helping users of KingFisher 1.40 make the transition to Release 2

All the little known facts!

1.14   Implementation of Client-Server Architecture

The Amiga's multitasking Exec (the software that handles all aspects of multitasking, including interprocess communication) provides a highspeed method of passing large amounts of information from one task to another.

The KFServer creates a message port, a rendezvouz, to which clients deposit requests for processing. The KFServer processes these requests one after another (first come, first served) and returns the results to the sender.

And that, my friends, is all there is to it! Quite simple, really. :)

Naturally, the protocol of how exactly to ask the KFServer for information requires some attention to detail. If you are interested in writing client software for KFServer and you are a registered user, you are eligible for a nearly free package[1] to help you get started. This package includes source code, extensive documentation on the KFServer API (Application Programming Interface) and source code to isolate you from the grueling details of setting up messages with parameters for every single request sent to the server.

All code supplied with the developer package is written for and compiles with SAS®/C 6.55.

References:

AMIGA ROM Kernel Reference Manual: Libraries. Exec Chapter.

COMMODORE-AMIGA "C" include files: exec/ports.h

_____

[1] The KingFisher Developer Pack costs the same as an upgrade but you also get the most recent version of KingFisher "thrown in" as a bonus. If you ask for only an upgrade, you get the upgrade. If you ask, instead, for the KingFisher Developer Pack, you'll get both, regardless if the upgrade will actually upgrade your current version. Of course, you could always explain that you're willing to wait some amount of time for the latest release.

The KingFisher Developer Pack is available free via email directly from the author at the address walrus@wam.umd.edu

1.15   2.1 Origin and History

There are many terms used by KingFisher and by this documentation which revolve around the concept of those strange dwellers in the water: fish. The reason for this is that in late 1985, a fellow named Fred Fish began to collect freely distributable software, put it all on disks and distributed these in a coherent and reliable manner. The concept of "Fish Disks" was born. To date, Fred has released a grand total of 1000 of these disks, containing over 4500 programs.

To provide some means of wading through the ever-growing pond of "Fish Disks," (grown into a veritable ocean by this time) a man named 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.

Around the time that the 400th Fish Disk appeared, I had become the self-appointed keeper of the Aquarium Database, and I spent quite some time every month updating the database, the flags that Aquarium needed to find information, and distributing updates to the database so that not everyone had to do the same thing over again. The frustrations of this job led me to design the tool that became KingFisher. Its first release, on December 1, 1992 was quickly followed by several more versions until it arrived after seven public updates at version 1.40.

But KingFisher 1.40 still fell short of what I wanted to accomplish. A lot of people wanted features that I couldn't comfortably patch into the code, and it was a hopelessly single-minded system, unable to work with anything but a single database and a single user. Those who are using KingFisher 1.x as a BBS support tool will be especially aware of its shortcomings in this area. All this, the feedback from hundreds of users, and Fred Fish's need for an effective database tool to support his move to a CD-ROM distribution led me to design what you have in your hands now: KingFisher Release 2.

KingFisher Release 2 serves multiple databases to multiple users in any configuration, fully supports Fred Fish's Product-Info Specification v6 (which I helped design), and offers practically every feature of KingFisher's first release that can be properly supported. The databases can be accessed through an ARexx interface and a resizable and proportional font aware GadTools window that opens on the default public screen. And more is on the way ...

1.16   2.2 Client-Server technology

Client-Server architecture is a database concept whereby one unique portion of the software, called the server, is responsible for controlling access to data, while one or more clients talk to the server and request data. It is the clients that are responsible for presenting the data, perhaps alter it and then handing it back to the server for storage.

The Client-Server model provides for efficient, successful, and safe multiuser arbitration and is widely used in the computer industry.

KingFisher implements this same powerful concept to provide you with safe access to one or more databases, and to extend this access, if you wish, to a number of simultaneous users that may have access to your system through BBS software.

KingFisher's server software is the KFServer .

KingFisher itself is "merely" a GadTools (GUI) client that talks to this server.

RexxFisher, also, is a client that talks to the server.

And so is ReOrder, a little tool available in source form, meant more as examples than useful programs.

All of these together form the product named KingFisher.

If you wish to learn a little bit about how the client-server architecture is implemented in KingFisher, then click here .

1.17   2.3 Search Expressions

NOTE: If you have the Simple substrings option selected, then KingFisher will accept a simplified version of the expression syntax described here. Watch for the symbol × to to point out important sections pertaining to this topic!

To check the status of the Simple substrings option, check the following menu:

| Preferences |

| Global » |

| Display » |

| Printing » |

| Exporting » |

| Searching |_____

| | Stop on each match |

| | Case sensitive |

|___ | Trim blanks |

| Simple substrings |

| Use search masks |

|_____|

Part I. A basic expression

An expression tells KingFisher exactly what you are looking for. An expression takes the form

field comparison value

× When the Simple substrings option is selected, only the value portion is used, and KingFisher automatically substitutes a * (all) for the field, and $ (substring search) for the comparison operator.

Choices for...

field the name of any database field . An example of this would be name, description, version, or author. You may use a * instead of a specific name to indicate that you wish KingFisher to search all existing fields in the records.

comparison one or two character symbol. The following are valid:

= or == The field contents are equal to the value

<> or != The field contents are not the same as the value

<= The field contents are alphabetically less than or equal to the value

>= The field contents are alphabetically greater than or equal to the value

< ... alphabetically less

> ... alphabetically greater

$ The field contents contain the substring given by the value

The field contents are compared to the given DOS Pattern.

value a string of characters. If the string contains any of the special symbols, such as: ( ) & | ˆ or blank spaces, it becomes necessary to enclose the string in single or double matching quotes: "" or ”

Examples:

name=kingfisher

version >=2

author $ "matt dillon"

Notice that spaces surrounding the three parts of the expression (field, comparison, and value) are unimportant.

Part II. Composite expressions

Let us now combine two expressions to form a more complex one:

name = kingfisher & version >= 2

Notice the new symbol, &, that we used. This is a boolean operator that you can use to connect two expressions. The following boolean operators are valid:

& Logical AND

Both the expression on the left and on the right side of the operator must evaluate to TRUE, or else the combined expression formed from the two will evaluate to FALSE.

| Logical OR

Inclusive OR

If either or both of the expressions on the left and on the right side of the operator evaluate to TRUE, then the combined expression also evaluates to TRUE.

ˆ Logical XOR

Exclusive OR

Either one, but NOT both expressions on the left and on the right side of the operator must evaluate to TRUE, otherwise the combined expression is FALSE.

The expression above, therefore, means: if the name equals 'kingfisher' AND the version is (alphabetically) greater than '2', then we have found a record that might be interesting.

Test your new-found understanding: What does the following mean?

name $ 'aquarium' | name = kingfisher

It means if the string 'aquarium' appears as a substring in the name field, OR the name equals 'kingfisher' then this is a match.

Part III. Sequence of evaluation

Let us examine a more complex expression. Assume we want to find all the records with 'aquarium' part of the name, OR all the ones named kingfisher which have a version of at least 2. Does the following expression work?

name$aquarium | name=kingfisher & version>=2

The answer is no! KingFisher uses left-to-right evaluation, meaning that the expression first evaluates

name$aquarium

then it evaluates

name=kingfisher

and then checks if EITHER is true. Only then will it proceed to test the version. If we use parentheses to demonstrate how KingFisher actually evaluates the expression, you'll notice immediately that we had something else in mind:

(name$aquarium | name=kingfisher) & version>=2

But KingFisher does understand parentheses, so we can easily fix the expression to do what we meant it to do in the first place. We just have to remember to use them:

name$aquarium | (name=kingfisher & version>=2)

You can use many levels of nested parentheses, and it is always safer to "overdose" on parentheses than to assume that the expression really means what you hope to express.

Part IV. Negation

Any part of the expression can be negated. If you wish to find all software in which Udo Schuermann had no part, you would begin by entering the following:

name$"udo schuermann"

and then negating the expression by placing the negation operator in front of it:

~name$"udo schuermann"

KingFisher recognizes both the ~ and the ! symbols as negation operators. Which one you use depends entirely on your preference.

Part V. Expression Shortcuts

To make a variety of comparisons to a single field a little easier, an expression shortcut represents enhanced ease of entry and formulation; it does not affect the speed of searches in any way. Expression shortcuts are not available when the Simple Substring is active.

Let us assume that you are looking for a large number of programs by name. You could enter the following expression:

name$post | name$music | name$fish | name$king | name$aqua

You will quickly come to the point where typing "name$" over and over becomes rather tedious. KingFisher now allows you to enter the same expression this way:

name$(post|music|fish|king|aqua)

Observe that the expression in parentheses consists only of values and boolean operators (the | symbol in our example, signifying an OR-expression.) The field identifier ('name' in our example) and how this field is treated (as a substring, see the $ symbol) is a constant and does not change for the values given.

Consider:

name="kingfisher" | name$"post"

This expression cannot be converted to a shortcut because in one portion we test for equality (=) and in the other we test for a substring ($).

In order to impose an evaluation order, the portion in ()'s may contain multiple levels of parentheses to prioritize the evaluation order:

name$(fish&(king|net|sticks))

represents:

name$fish & (name$king | name$net | name$sticks)

Part VI. DOS Patterns

The special comparison symbol '#' serves to identify the following string as a DOS Pattern. In case you are not familiar with these, look in your AmigaDOS manual (the AmigaDOS 3.1 manual describes these in Chapter 3, pp. 16-18) under the heading Pattern Matching and/or Wildcards.

Consider:

name#"#?king#?"

This pattern is the exact same thing as

name$"king"

except that the former uses the dos.library's pattern matching, whereas the latter uses a simpler, but faster substring comparison. Note especially the difference between the '#' vs. the '$' operator to tell KingFisher how you want the string treated.

The example above is probably as unexciting as a slice of stale bread, but if you are familiar with DOS Patterns, you should have already realized that they can help you build some powerful expressions. You must understand, however, that the dos.library matches the entire expression against the field contents, whereby you will usually want to surround the string with the "0-or-more-characters" wildcard: #?

The expression name#"king" will NOT find "kingfisher", nor will it match any program that is not exactly named "king". Read the DOS documentation for more details. Enough said.

NOTE: The Trim Blanks option will affect the search string, unless it is enclosed in quotes. You are strongly urged to always enclose your DOS Patterns in quotes!

NOTE: If you specify a DOS Pattern that contains no wildcards, in other words it is a simple string constant, then KingFisher will convert the expression to an equality search. Example:

name # "kingfisher"

will not use the dos.library for pattern matching, but rather convert the expression to

name = "kingfisher"

and evaluate the equality directly and without assistance from DOS.

And that concludes the overview on expressions.

1.18 2.4 Search Sets

Search sets represent one of KingFisher's newest, and perhaps most useful features. Before searching, you must select whether or not you wish to make this an interactive or a non-stop search by checking or unchecking the PREFERENCES/Searching

submenu item. Only when the Stop on each match item is not checked, will KingFisher produce search sets that you can examine at your leisure. The value of both choices is discussed there.

Search sets need not be saved to be useful, although you can save yourself much time if you save the results of oft-repeated searches. Search sets do not require much diskspace: approximately 5 bytes per record. A thousand records, which is nearly ¼ of Fred's 1000 Fish Disks, would require about 5K on disk.

When a search set is loaded, regardless if it is shown in the listview, the gadget with representations of Fred's Fish Logo will become active. You can click on this to open and close the Search Set window. Clicking on the Search Set window's close gadget will also close the window. Neither of these actions will destroy the current Search Set!

The Search Set is only cleared from memory by one of the following actions:

Quitting KingFisher

Beginning another search

Loading a new Search Set

Loading a Search Set will, if necessary, switch to the database to which the Search Set applies, and will position you at the first record listed in the Search Set. The current Search Expression is also remembered to remind you what the Search Set represents.

1.19   2.5 Custom Formats

The format that KingFisher uses to display, print, and export fish is programmable! This means that you can customize the display just about any way you like to! This may require some persistent trial-and-error, but the results may be worth the effort if your needs are not served by the default format. Here is what the default format looks like:

@{name}@{version| }@{date| (|)}\n

@{fullname|\t|\n}

@{short|\t|\n}

@{author|\tBy\>|\n}

\n

@{description}\n

\n

@{requirements|Requirements:\n\>|\n}

@{restrictions|Restrictions:\n\>|\n}

\N

@{address|Author's Address:\n\>|\n}

@{phone|Phone:\>|\n}

@{fax|Fax:\>|\n}

@{email|Email:\>|\n}

\N

@{distribution|Distribution:\>|\n}

@{price|Price:\>|\n\n}

@{installsize|Installs:\>|\n}

@{source|Source:\>|\n\n}

@{exectype|ExecType:\>|\n}

@{construction|Construction:\>|\n}

@{tested|Tested:\>|\n\n}

@{docs|Docs:\n\>|\n\n}

@{references|References:\n\>|\n\n}

@{reference|References:\n\>|\n\n}

@{keywords|Keywords:\>|\n\n}

@{described-by|Described-by:\>|\n}

@{submitted-by|Submitted-by:\>|\n}

@{submittal|Submittal:\>|\n}

@{stored-in|Stored-In:\>|\n}

Looks gruesome, doesn't it? I agree, but computers are just so good at making sense of gruesome things, and they're terrible at working with things that we humans have no trouble understanding. This is why the formatting is described by a gruesome mess: KingFisher understands this stuff a lot easier. The bottom line is that it can display things much quicker this way, and in the end that's probably more important than a pretty behind-the-scenes format file.

So, how can we make sense of this gruesome mess?

It's actually less gruesome (nice word, eh?) than you might think. First, you may have already noticed (you're pretty quick, aren't you?) that almost everything begins with the symbols @{ followed by something gruesome and is terminated with a } Coincidence? Definitely not!

Let's look at the first line and show all four elements on that line, one at a time, underlined:

@{name}@{version| }@{date| (|)}\n

@{name}@{version| }@{date| (|)}\n

@{name}@{version| }@{date| (|)}\n

@{name}@{version| }@{date| (|)}\n

You will notice that the first item is @{name} which looks simple enough. It displays the contents of the name field!

The second line, @{version| } looks a little stranger: there is a | symbol stuck in there, along with a blank space. Let me quickly point out that the gruesome mess between @{ and } symbols can contain more than only a field name. The complete format (without the blank spaces!) is:

@{ field | prefix | suffix }

The | symbol is a separator, and the blank space is the value of the prefix portion. But what, you may ask, is the point of this weird concoction? Why not put the space outside the whole @{} construct?

The reason is that when the specified field is missing from the database or contains no information, then neither the prefix nor the suffix, if any are given, will be processed. This neat trick is used extensively and permits us to print something additional before and/or after the field contents if the field contains data, and do absolutely nothing if the field is empty.

Let's look at the third line item, @{date| (|)} which contains both prefix and suffix strings. If a date field does not exist in the database, there won't be a non-sensical " ()" shown. A content sensitive display format!

The fourth item is one that will be quite familiar to C programmers: \n is a newline. KingFisher begins the following text on a new line. This allows you to break up things into more readable sections. The end of line in the file is actually considered merely a blank space by KingFisher so that you can break things up into a more readable form.

Here is a listing of the special formatting symbols. They may be used both inside a @{} construct and outside:

\. A single . (dot) especially useful if/when such a dot is found (against normal practice) at the very beginning of a line of text and where it would then be misunderstood to represent a field-name.

\@ An @ symbol.

\n A newline, an end-of-paragraph.

\N A conditional newline (end of paragraph), inserted only if the previous line is not already a blank line.

\t A tab, which is equivalent to approximately 5 spaces.

\> A paragraph indent, which allows you to create hanging indentations of text. The indentation will remain in effect throughout the contents of a field, so it is, in effect, a temporary change of margins. Newlines embedded in a field's data will only reset back to approximately the same column as the field's first character.

\- Adds a line of dashes (------...) up to the end of the line. Makes for nice dividers!

All other text encountered is transferred verbatim to the display.

Field Option Sets

One or more options for a field can now be specified by placing them within []'s immediately before the field name. At this time the number of available options is quite small, but may grow in the future. Here is an example of an option set in use:

@{[f]author\tBy\>\n}

Notice that there are no spaces added, and that a '[' cannot, therefore, be the first character of a field name, unless you also specify an option set. The following specification is legal:

@{[][funnyfield]}

This specifies an empty option set and a field [funnyfield]. The empty option set keep the parser from thinking that the first character of the fieldname begins an option set.

Here is a list of the available options:

f,v Force FLOW or VERBATIM mode..

In Flow mode, the presence of a newline in the database record is treated as a blank space would, and only a \n sequence is regarded as a real newline (paragraph break.) In Verbatim mode, a newline in the database record is treated the same as a forced paragraph break.

0,1 Forces either 1 or no extra lines to be inserted between forced paragraphs. By default, '1' is used for the description field, which is also handled in Flow mode.

|,> Disables or forces paragraph-level indentation on a field. Indentation is never turned on, but may be desirable for the description field, for example, especially if no extra lines are inserted between paragraphs (see 0,1 above.)

You could alter the display of the description field by adding an option set [0>] (or [>0]) to it. Try it, you might like it!

The following fields are usually handled in Verbatim mode: reference, address, email, docs, run, stored-in, and author.

All other fields are handled in Flow mode.

For descriptions on format and purpose of available fields, please refer to the Product-Info specification from Fred Fish. The following is a list of the fields referenced in the Product-Info Specification v6 :

name

fullname

type

short

description

version

date

author

restrictions

requirements

reference

distribution

price

address

email

exectype

installsize

source

construction

tested

run

docs

described-by

submittal

stored-in

Click here for some example custom formats!

1.20 Examples of Custom Formats

Example1:

@{name}\n

\>@{description}@{author| Author: }\n

@{version|\>This is version |\n}

Output:

MonkeyCommand

Lure the lovestruck monster ape back to

his island. Tools include Fay Wray's torn

nightgown, a Fokker airplane (you get to

pilot it), a compass and a map. Author:

KingKong Industries

This is version 1.0

Example2:

@{name}|@{version}|@{author}|@{price}|@{short}\n

Output:

MonkeyCommand|1.0|KingKong Industries|||

Example3:

@{name}\>@{description}

@{version| This is version |.}

@{author| Author: |.}\n\n

Output:

MonkeyCommand Lure the lovestruck monster ape back

to his island. Tools include Fay

Wray's torn nightgown, a Fokker

airplane (you get to pilot it), a

compass and a map. This is version

1.0. Author: KingKong Industries.

Example4:

PROGRAM NAME:\t@{name}\n

VERSION:\t@{version}\n

AUTHOR:\t@{author}\n

RELEASE DATE:\t@{date}\n

\n

@{[0>]description}\n

\n

\-\n

Output:

PROGRAM NAME: MonkeyCommand

VERSION: 1.0

AUTHOR: KingKong Industries

RELEASE DATE:

Lure the lovestruck monster ape back to his island.

Tools include Fay Wray's torn nightgown, a Fokker

airplane (you get to pilot it), a compass and a map.

1.21   2.6 KFServer and Databases

Without the KingFisher Database Server, KingFisher is little more than gerupftes Federvieh (a plucked bird, in English, but it sounds much funnier in German :)

KFServer is the all-important portion of the software. Regardless how you access the database, through KingFisher's GadTools GUI, RexxFisher's ARexx interface, through 3rd party client software, or something you wrote yourself, KFServer will always come into play!

Both KingFisher and RexxFisher know how to start the KFServer if it is not already running. KingFisher is, at this time, somewhat better at this because it can be told to start KFServer from a directory other than the one in which KingFisher starts itself.

The KFServer.prefs file

For the KFServer to successfully start, it must be able to read its .prefs file. This file is named "KFServer.prefs" and must contain at least the following information. All blank lines or line beginning with a hash (#) mark are considered comments:

default-database=1000Fish.kfdb

This line specifies the so-called "Default Database" which is the database KFServer will always open. Any client connecting to KFServer will have this database made the initial database until it selects a different one. In the case of KingFisher, you may not realize this happening, because KingFisher remembers the last used database and automatically switches to that before displaying the first record.

Notice that the filename, 1000Fish.kfdb, has a .kfdb extension. It stands for "KingFisher Database." The contents of a .kfdb file will be described below. First, let's examine what optional items you can place into the KFServer.prefs file:

maxclients=5

This line specifies that KFServer will not allow more than 5 simultaneous clients to connect at one time. This value must be at least 1, and cannot exceed KFServer's maximum. Unregistered versions of KingFisher have limit of 2. Registered versions have a limit which you will never be able to exceed unless you have too much time on your hands and you are ridiculously rich and can afford 256MB of RAM for your Amiga to run hundreds of millions of copies of KingFisher.

verbosity=MUTE

The verbosity value specifies how talkative you want the KFServer to be. Ordinarily you will want to set this value to MUTE to make the KFServer shut up as much as possible. Only real problems will be reported, things you should be aware of (like a database being unavailable.) If you find that something is not working, you might want to try a higher verbosity value, until you either can no longer stand the amount of output or you find the problem. The following values are available, and you can specify them in upper, lower, or mixed case:

mute Cries out in only terribly critical situations

terse Hardly sends any messages to the output window

quiet Sends occasional messages of interest to the output window

chatty Rather talkative with lots of status information

debug Produces a nearly continuous stream of information

loadindex=DefaultInRAM

Determines how a database main index is handled. In particular, it can override the INDEX entry in .kfdb files:

AlwaysInRAM Regardless of what .kfdb files specify, the main index is always loaded into memory for maximum speed.

AlwaysOnDisk Regardless of what .kfdb files specify, the main index is always kept on disk for minimal use of RAM.

DefaultInRAM If a .kfdb file fails to specify an INDEX entry, the index is then loaded into RAM. This is the default.

DefaultOnDisk If a .kfdb file fails to specify an INDEX entry, the index is then maintained on disk.

sanity-check=InRAM

Controls the server's sanity check operation, which is performed whenever a database is first opened.

InRAM Performs the operation only on an index that is loaded into memory (default)

OnDisk Performs the operation only on an index that is maintained on disk (this can be rather slow!)

Always Performs the operation on every index, no matter if it is an InRAM or OnDisk index.

keep-running=yes

By default, KFServer will automatically exit when the last client program detaches, requiring to be started again if another client then wants to use the KFServer. By setting the keep-running value to "yes" (instead of "no", or omitting it altogether) the KFServer will remain running even after no more clients seem to need its services. This behavior is best suited for situations where clients start and quit frequently, such as with a BBS.

NOTE: Earlier releases of KFServer kept running unless keep-running was set to a value of "no." This behavior has now been altered for less confusing single-user operation.

window=CON://640/480/KingFisher Release 2 Server Messages/AUTO

The output file to which KFServer writes all error messages should be set to a console window (such as given above) rather than a file, although reason could certainly be found where a file would be desirable. KFServer does not care where you send output, so long as you specify a valid file.

suppress-requesters=yes

By default, KFServer will not produce requesters asking the user to insert a required disk. Instead, the server returns the database selection request with an explanation added, thereby allowing the client to retry the operation based on possible user-feedback. This prevents the server from blocking when serving multiple users, which could cause serious problems in a multi-user environment. It is not recommended to turn requester suppression off.

The .kfdb files

The KingFisher Database file must have an extension of .kfdb, otherwise KingFisher will not be able to list them to a client, should the client wish to know what databases are available. Furthermore, the .kfdb files must be located in the same directory where KFServer is started. The following are absolutely required in all .kfdb files. All blank lines and those beginning with a hash (#) mark are considered comments:

database-name=1000 Fish Disks

Specifies the descriptive name of the database. This is the name presented by KingFisher to the user when using the Open Database command. Keeping this name relatively short is a virtue. The example is about as long as you would ordinarily want to make it.

section=00000,02500,MyFish:Fish01.data

section=02501,05000,MyFish:Fish02.data

One (or more) sections must be specified. Unlike the original KingFisher, which used a format strikingly similar, the two numeric values (0 and 2500, as well as 2501 and 5000 in our two examples) are not disk, but record numbers. The above values work for my own copy of the database used by KingFisher 1.40, but it may not work for you. As KingFisher Release 2 ships with a functional database of all 1000 Fish Disks, I do not expect this to be an issue.

The three portions of each section value are:

beginning record The first record in the database is 0, not 1. KingFisher always adds 1 to the record numbers because that is how most people view a database.

ending record The last record in this section of the database.

storage filename The exact name of the file where you wish to store a portion of the database. If the datafiles are stored in the same directory as the .kfdb file, then a specific path to the files is not required. If, however, the data and index files are located elsewhere (such as on a CD-ROM or floppy disk) then a complete path to the files is essential!

Note, that you can break up the database into as many section as you wish, or keep it all in one contiguous chunk. The organization of a new database is entirely up to you. The CLI tool 'ReOrder' can be used to effectively change these values by copying all records from one .kfdb file to a new .kfdb file with different ordering, then removing the original .kfdb file and all related files.

index-name=MyFish:1000Fish.index

The index-name specifies the name of the main index. This file will be recreated whenever something about the index changes, such as new records are added, or the database is truncated, or you alter any of the flags that are part of each record. KFServer updates this index file on disk whenever the database is closed.

The following are optional values you can place into the .kfdb file to determine how KFServer is to treat this database:

index=inram

The current implementation of KFServer loads an index file into memory, whereby the index is said to be "INRAM." While an "ONDISK" index has not been tested enough for me to make the claim that this will work, enough of it has been tested that it may actually be usable and even error free.

Please be advised that an ONDISK index may seem functional, but is not yet officially supported. If you wish to experiment with this option, feel free to do so, but please understand that the results may range anywhere from index-related access errors to a corrupted index file. Even if something goes wrong with the index, delete the index files, switch back to an INRAM index, and reindex the database. Another way to experiment safely is to make backup copies of your index files.

index-increment=100

This value defines by how many records at a time KFServer should expand an inram index whenever you add records to the database and the index need to grow. It is more efficient to grow the index in large leaps at a time, but can waste memory if, for example you are growing the index in step of 1000 index records, and after 1000 records, you merely add one additional record to the database. KFServer will then have allocated 2000 records but is only using 1001 of them.

Do not be overly concerned about this, however. The initial index size when KFServer opens a database, is always exactly what is needed, no more. Only adding to the database will bring the index-increment value into play.

keep-open=yes

With the exception of the Default Database specified in the KFServer.prefs file, KFServer will always close a database (and all its files) when no client remains that is using it. If, for some reason, you rather have the files, as well a the database index, remain open and loaded, you can set this behavior for each database with this value.

read-only=yes

Marking a database read-only signifies to the client program that certain operations, such as the adding of new records or the alteration of flags, is not permitted. The server may still accept such commands if they are not disabled by the client, but if a database can truly not be modified, because the index is located on read-only or write-protected media, then the server may produce error messages to this effect when a change needs to be written back to disk.

field-index-name=MyFish:1000Fish-Name.index

The name of the database file that will store the QuickIndex . As a convention, you might want to use the value of the index-name entry but add '-name' (for the name field's QuickIndex, see below) before the .index file extension. This permits easy upgrades in future versions when multiple QuickIndexes may become available.

field-index-field=name

Specifies the database field to which the QuickIndex applies. You should always use the 'name' field here because some functions (such as Build VersionLinks...) require a QuickIndex on that field. KingFisher will be able to make use of a QuickIndex on another single-line field for searching, but does not (and will not) support multi-line fields. Use of multi-line fields may cause undefined behavior. Unless you have very good reason to do otherwise, set this entry to name.

1.22   2.7 Product-Info Specification

A rough framework of the requirements, which eventually became the Product-Info Specification , was originally started by Fred Fish and his brother. Basing my work upon their original efforts, and then relying on their feedback as well as that of a number of other people, the Product-Info Specification is meant to address the following basic issues:

1. A text-only format makes it possible to easily create and modify these files without special software. This makes them portable between computer systems, easily transmitted, and more resilient to minor damage,

2. Aside from defining characteristics for some specifically named fields, the files are freely extensible by users to suit their individual requirements; in addition, the ordering of fields is (almost) completely irrelevant,

Main text of the Product-Info Specification

Fields defined by the Product-Info Specification

Starter file for a .Product-Info file of your own

1.23   2.7.1 Product-Info Specification: Text

This is a partial text of the 6th draft of the Product-Info Specification. This text details how database records can be transported by email to allow their easy extraction from within unrelated text.

The purpose of Data Transport Markers is to provide explicit delimiters for data that is surrounded by non-database records. The original KingFisher contained a very complex finite state automaton (sic) to extract data out of email and news files. This FSA relied on certain conventions and would fail to work if those conventions were not followed.

In order to enforce a more reliable means to encapsulate and transport data surrounded by irrelevant information, KingFisher Release 2 supports no other format for importing data but that which conforms to the Product-Info specification.

Records must be enclosed by special markers, such as shown in bold in the example below:

.BEGIN-FISH-DESCRIPTION

.name

MonkeyCommand

.author

KingKong Industries

.description

Lure the lovestruck monster ape back to his island.

Tools include Fay Wray's torn nightgown, a Fokker

airplane (you get to pilot it), a compass and a map.

.path

FishROM001:games/MonkeyCommand/

.END-FISH-DESCRIPTION

Furthermore, the data enclosed must consist of one or more actual database records, and specification for these requires the first field of every record to be the name field as shown above.

KingFisher will read files without Data Transport Markers, such as #?.pi, .Product-Info, or Product-Info files, but no guarantee can and will be made that it can successfully do so with files that start with data not relevant to the desired information.

This represents an added flexibility of KingFisher's parser, not an implied extension to the Product-Info Specification.

According to this Product-Info Specification, KingFisher Release 2 will extract the relevant information from the following sample file (and it does!):

Hi Tom,

Remember that monkey game you told my about?

.BEGIN-FISH-DESCRIPTION

.name

MonkeyCommand

.author

KingKong Industries

.description

Lure the lovestruck monster ape back to his island.

Tools include Fay Wray's torn nightgown, a Fokker

airplane (you get to pilot it), a compass and a map.

.path

FishROM001:games/MonkeyCommand/

.END-FISH-DESCRIPTION

Well, seems that one wasn't enough and they released

another one. We'll have to figure out how to finally

beat the first one, it seems, before they let us play

the next. Maybe we can look through the binary to find

that code phrase. Here's the text:

.BEGIN-FISH-DESCRIPTION

.name

MonkeyCommand II

.author

KingKong Industries

.description

Keep the captured ape from breaking through the defenses

of the prison that was erected at the conclusion of

MonkeyCommand I. The game consists of coordinating the

actions of four native tribal leaders and their vassals

in repairing the damage done by the angry beast.

.restriction

You need the secret code from the first MonkeyCommand

which you can only get if you won the game.

.path

FishROM002:games/MonkeyCommand2/

.END-FISH-DESCRIPTION

(=:Joe:=)

NOTE: When copying data to the Clipboard, KingFisher supplies these markers for you!

1.24   2.7.2 Product-Info Specification: Fields

The following are the fields defined by the Product-Info Specification v6 as designed by Fred Fish and Udo Schuermann.

.name

PURPOSE: The program's name

FORMAT: 1 line only

EXAMPLE: KingFisher

EXAMPLE: HomeBase VI

EXAMPLE: AIBB

EXAMPLE: gcc

.fullname

<<<OPTIONAL>>>

PURPOSE: The program's full (or complete) name

FORMAT: 1 line only

EXAMPLE: Amiga Intuition Based Benchmarks

EXAMPLE: GNU C Compiler

NOTES: If the .name is not an abbreviation then omit the

.fullname. No sense in giving the name twice!

.type

PURPOSE: A keyword that describes the nature of the program

FORMAT: Preferably a single word or two.

EXAMPLE: Database

EXAMPLE: Spreadsheet

EXAMPLE: Animation Player

EXAMPLE: Animation Tools

EXAMPLE: Communications

EXAMPLE: Display Commodity

EXAMPLE: Mouse Commodity

NOTES: Avoid abbreviations. Refer to the list below for

suggestions.

.short

<<<OPTIONAL>>>

PURPOSE: A one-line description, preferably not exceeding

40 characters in length. This description is to

give a single-glance insight into the program's

purpose.

FORMAT: 1 line only.

EXAMPLE: Software catalog/search/maintenance tool, multi-user.

.description

PURPOSE: A full-text description of your program, containing

anything that is NOT ALREADY available through the

other fields (see above and below.) The reader

should gain a good understanding what your program

can and cannot do. If you mention other programs

please do not forget to provide a .reference field

for each such mention.

FORMAT: Any number of lines, treated as one line.

Formatting is permitted, but generally discouraged.

NOTES: Do not indent your text if you choose to format

your text into multiple paragraphs. Do not use \t

as a tab. Leave paragraph formatting to KingFisher.

.version

PURPOSE: The program's version number

FORMAT: MAJOR.MINOR

1 line only

EXAMPLE: 37.100

NOTES: Please note that the Commodore guidelines specify

that the number after the period is NOT a FRACTION

but rather a WHOLE NUMBER! Thus, the following is

a valid progression:

37.1 37.17 37.39 37.100 37.170

The following are all vastly different versions:

37.1 37.10 37.100 37.1000

NOTES: The format given for this field is really more of a

SUGGESTION rather than a RULE. There is no reason

why you can't store "Today's Version" or "v940205"

instead of 18.173. In an ideal world everyone

would use Commodore guidelines, but there are

enough exceptions.

.date

PURPOSE: The program's official release date; not the date

it made it into the database.

FORMAT: year.month.day

1 line only

EXAMPLE: 1993.09.27

NOTES: The date format is chosen to be easily sortable.

Note the use of leading zeros in month and day.

The full year is to be given in anticipation of

the coming change to a new millennium.

.author

PURPOSE: Any and all authors who have a part in the program

FORMAT: Any number of lines, treated as one line (\n in the

text will "break up" the line into multiple visual

lines.)

EXAMPLE: Joe R. User, Tea Rexx.

EXAMPLE: J. Jones\n

Random Hacker\n

B. Clinton

NOTES: Addresses should be placed in the .address field.

There should be only one .address field for each

.author field.

If more than 1 .author field is specified, then the

same number of .address and .email fields must also

be given in a 1-to-1 relationship (i.e. the 3rd

.author field must be associated with the 3rd

.address, and the 3rd .email field.)

EX: see the example "Joe R. User, Tea Rexx" above;

Assume that Joe R. User has long vanished and no

known address, but that Tea Rexx has supported the
program for a while. If an .address and/or .email
field is available for Tea Rexx, then you must
specify EMPTY .address and/or .email fields for the
author listed BEFORE the ones for Tea Rexx.
Likewise, if the two authors names were reversed,
you would NOT have to specify blank .address and/or
.email fields for the second author. I hope that
makes sense.

.restrictions

PURPOSE: List restrictions placed upon this program. These
should indicate in which way this program has been
made dysfunctional (for demo purposes), problems
(bugs) known to exist with this program, or any
other thing that lets the user know that this
program, as seen in this distribution, may not
fully satisfy the user in some form.

FORMAT: Free form; see .description for more info.

EXAMPLE: Demo version has SAVE and PRINT options disabled.

EXAMPLE: The ReadOperatorsMind command fails to work with
CDTV units. Incompatible with the Discus Ejector
utility.

EXAMPLE: Crashes if iconified while loading a sample or
image larger than 64K.

EXAMPLE: Requires a PAL display.

EXAMPLE: The program is in German but the documentation
offers translations into English and Swahili on
a menu-by-menu and gadget-by-gadget basis.

NOTES: Do NOT use this field for things like "won't work
with KS 1.3" or "won't run with less than 2 Megs
of RAM."

.requirements

PURPOSE: List requirements for your program. These should
give the reader enough information to determine if
the software will run on his/her system or not.
Be sure to specify operating system versions,
(hard)disk space requirements, etc. If your

program requires any external libraries that are

not part of the system software, it would be nice

to list them here and comment on whether or not

they are included in the archive.

If your program is known to run on every existing

(Amiga) platform, state this in this field!

FORMAT: Free form; see .description for more info.

EXAMPLE: 68020, 68030, or 68040 CPU; 3M free RAM; 18M disk

space; at least 640x480 display capabilities!

EXAMPLE: Requires WB2.1 (V38)

EXAMPLE: Requires 1024x768 (or larger) display capability.

EXAMPLE: Works only with 4096-channel, 230db BLAZETHUNDER

Audio board.

EXAMPLE: Requires MUI (MagicUserInterface) version 5.

.reference

<<<OPTIONAL>>>

PURPOSE: Full path to where this program's files are stored,

as well as the version that is stored there.

FORMAT: 2 lines per reference: the first line specifies

the full path (with trailing slash) and the second

line, the version.

NOTES: Multiple such fields may be provided to reference

previous versions of this program, as well as

other programs that might be of interest. The

versions should be listed in reverse chronological

order and SHOULD include the CURRENT entry.

Please note that it is VERY VERY VERY important

that you specify the CORRECT PATH! Without a

correct path, this entry will be nearly useless!

SPECIFY THE PATH WITH A NEW SUBMISSION ONLY IF YOU

KNOW WHERE IT IS STORED; NEW SUBMISSIONS WILL HAVE

A PATH ASSIGNED HERE AUTOMATICALLY. YOU SHOULD

PROVIDE THE PROPER PATHS TO KNOWN AND EXISTING

SOFTWARE.

EXAMPLE: FishROM-0002:Productivity/Databases/HomeBase VI/

417.0

FishROM-0001:Productivity/Databases/HomeBase VI/

415.12

.distribution

<<<OPTIONAL>>>

PURPOSE: Describes the distribution and ownership status
of this software. Please see below for a list of
common (and recommended!) terms to use.

FORMAT: 1 line

EXAMPLE: Shareware

NOTES: Please see the table below for descriptions of the
recommended terms.

.price

<<<OPTIONAL>>>

PURPOSE: Describes the cost of this program to the user.

FORMAT: Any number of lines, treated as one line.

EXAMPLE: $50(US), DM75.

NOTES: In order to make this field more useful, it is
STRONGLY recommended that the FIRST currency
listed is United States Dollars as shown in the
EXAMPLE above. This allows a search to be limited
to a common price base. If you charge no money
for this program, omit this field!

.address

<<<OPTIONAL>>>

PURPOSE: Describe a full postal address of the author, to
be used if it becomes necessary or desirable to
contact the author. Do not specify the author's
name, as this is already in the .author field.

FORMAT: Multiple lines; formatting symbols \n are not
required, as physical line breaks are equivalent.

NOTES: SEE THE .author FIELD FOR IMPORTANT INFORMATION

.email

<<<OPTIONAL>>>

PURPOSE: Describe a full electronic mail address. Make
sure that this address is complete and reachable
even from less well-connected sites. The author
of KingFisher, for example, can be reached as

walrus@wam.umd.edu

It would be an error to specify only "walrus" or

"walrus@wam" even though these will work within

the particular organization where this address

is valid.

FORMAT: Multiple lines; formatting symbols \n are not

required, as physical line breaks are equivalent.

Do not specify more than one email address per

line. The more you abide by RFC-822 specifications

the better.

EXAMPLES: walrus@wam.umd.edu (Udo Schuermann)

Udo Schuermann <walrus@wam.umd.edu>

"Udo Schuermann" <walrus@wam.umd.edu>

<walrus@wam.umd.edu> Udo Schuermann

NOTES: You may specify multiple electronic mail addresses

in order of decreasing reliability and permanence,

each on its own line.

SEE THE .author FIELD FOR IMPORTANT INFORMATION

.exectype

<<<OPTIONAL>>>

PURPOSE: Describe the type of executable(s) that make up

your program. Examples: 68xxx, AMOS, Script,

ARexx, Compiled basic, Amigabasic, etc.

FORMAT: Free form; see .description for more information.

EXAMPLE: AMOS

EXAMPLE: 68000, 68020, and 68040.

EXAMPLE: Compiled BASIC

EXAMPLE: Compiled ARexx

NOTES: AMOS-based software has been said to not work on

some systems at all; this entry allows a user to

determine if the software is worth obtaining in the

first place.

.installsize

<<<OPTIONAL>>>

PURPOSE: Indicate the minimum and maximum sizes of the

executable as it is installed. The minimum size

should give an indication of how much diskspace

is required for a minimal installation (perhaps

lacking help files and miscellaneous tools) while

the maximum size should indicate the absolutely

highest amount of diskspace required by the

program.

FORMAT: 1 or more lines; Only the first line has a fixed

format, the rest are free-form. See examples.

Always indicate the number scales with a capital

K (for kilobyte) or M (for megabyte)

EXAMPLE: 220K - 2M

Most of the database files can be kept on floppy

disks, so valuable harddisk space is not wasted.

EXAMPLE: 18K

EXAMPLE: 38K - 500K

Lots of documentation and example scripts make up

the bulk of the installation.

.source

<<<OPTIONAL>>>

PURPOSE: Describe what source code is available with this

program. If source code is not available then

omit this field. The .construction field often

helps further identify the type of source if you

omit details here. How large is the source?

FORMAT: Free form; see .description for more information.

EXAMPLE: SAS/C,Manx,DICE source (750K) available for $15

EXAMPLE: Oberon source included. 85K

EXAMPLE: Limited C source (15K) included.

EXAMPLE: All source plus custom libraries, included: 12MB

.construction

<<<OPTIONAL>>>

PURPOSE: Describe the type of language(s) used to create

this program and the methods used to build the

final executable. If possible, include the

compiler version(s) and possibly important

options, such as optimization.

FORMAT: Free form; see .description for more information.

EXAMPLE: SAS/C++ 6.5 with full optimization.

EXAMPLE: AdaEd.

EXAMPLE: Fortran with self-made compiler.

EXAMPLE: AMOS

NOTES: This is usually closely related to the .exectype

field but differs from it in that the .exectype

might be "Compiled C" but the compiler used was

"RottenC 0.97"

.tested

<<<OPTIONAL>>>

PURPOSE: Give an indication of which configurations have

served as test environments.

FORMAT: Free form; see .description for more information.

EXAMPLE: A500(512K Chip, 0K Fast, 1 Floppy), A2000(1M Chip,

2M Fast, 40M HD, 1 Floppy); not tested on 68020+

CPUs.

EXAMPLE: A1000, A500, A600, A2000, A2000/30, A3000, A1200,

A4000/30, A4000/40 with various amounts of Chip

and Fast RAM, with and without MMU or FPU. Found

to be free of Enforcer hits and able to work with

virtual memory products; compatible with Retina,

EGS/Spectrum, and Picasso software. Also tested

under V33 through V40 system software.

.run

<<<OPTIONAL>>>

PURPOSE: Specifies how to start the program.

FORMAT: visible=type,command

Where 'type' is either WB or CLI to indicate the

required startup environment.

EXAMPLE: HomeBase VI=WB,HomeBase VI

HomeBase VI=CLI,ExecuteMe.HB6

HomeBase VI Fixer=CLI,ExecuteMe.HB6Fixer

EXAMPLE: FishTub=WB,ExecuteMe

NOTES: KingFisher requires that this entry strictly

follows the above format.

The user is shown all text up to the first equal

sign (the 'visible' portion.) The 'type' portion

must be terminated with a comma (,) and following

it will be the command to be executed.

Selecting it will either invoke the program from

the Workbench (invoking it as if double clicked on

its icon (if the .info file exists), or execute the

indicated shell command line as if it has been

typed at an open console window.

.docs

<<<OPTIONAL>>>

PURPOSE: List all documentation files, possibly for viewing

from within KingFisher for more detailed info.

FORMAT: 1 line per file

EXAMPLE: HomeBase.guide

HomeBase.dvi

HomeBase.doc

NOTES: KingFisher examines the EXTENSION and invokes the

appropriate viewing tool: MultiView/AmigaGuide for

.guide files, ShowDVI for .dvi files, more for

anything else. These files can also be sent to the

printer via KingFisher (i.e. print .ps or .doc

files.) KingFisher will honor the PAGER

environment variable (defaults to 'more') to

display standard text.

NOTES: Omit any path to these files, unless it is a

relative path from within the program's CD-ROM or

disk directory. Do not specify these files if

they are located within archive files; remember:

the files must exist as they are given here!

.described-by

<<<OPTIONAL>>>

PURPOSE: Specifies who created the description (Product-Info

file) for the program.

FORMAT: Free form; should include an electronic mail

address, too, if available.

EXAMPLE: Fred Fish (fnf@fishpond.cygnus.com)

EXAMPLE: Udo Schuermann <walrus@wam.umd.edu>

.submittal

<<<OPTIONAL>>>

PURPOSE: Identifies who submitted the program to Fred or

else how this program came to be on the reference

disk.

FORMAT: Free form; usually one line.

EXAMPLE: Submitted on disk directly by the author.

EXAMPLE: Downloaded from wuarchive.wustl.edu in pub/aminet/util/misc

----

.stored-in

PURPOSE: Specifies where and especially HOW the application

is stored. This field should specify EITHER the

name of a directory (ending with a : or a /) OR the

name of a file (one that does NOT end with : or /)

FORMAT: 1 or more lines.

EXAMPLE: FF1000:Disks701-1000/Disks941-960/Disk950/Enforcer/

FF1000:BBS/Disks501-1000/Disks941-960/Disk950/Enforcer.lha

NOTES: It is up to the particular application to decide

how to handle this information. If the extension

on the file is .lha, .lzh, .Z, .zoo, .pak, .zip,

etc. then you could, for example, call upon the

archiver of choice to unpack the application into a

temporary directory and let the user run the

program or list the files, or whatever.

1.25   2.7.3 Starter .Product-Info

INSTRUCTIONS: Using MultiView/AmigaGuide's SAVE AS command (menu), write this page to a file. Call it .Product-Info. Fill in what you need based on the description of Fields given in the previous section. Not all fields are required, and some may need special formatting.

Ship the resulting file with your product!

Acceptable names for the file (in increasing order of desirability) are:

.Product-Info

Product-Info

myproject.pi

------------------(Delete this line and all text above)------------------

.name

Program's Name

.fullname

Long/full name, if any

.type

Type of program (see below)

.short

Short (40 character) description, à la Aminet

.description

Long, possibly quite verbose description

.version

Release.Version

.date

Release date (yyyy.mm.dd)

.author

Author's name

.restrictions

Restrictions (perhaps crippleware info)

.requirements

Special requirements (such as MUI)

.reference

Reference to other related programs, two lines each (1: path, 2:version)

.distribution

Distribution type (see below)

.price

Price (if any)

.address

Author's postal address (not including author's name)

.email

Author's email address

.exectype

ARexx, shell script, binary, interpreted BASIC, ...

.installsize

How big is this thing, approximately?

.source

Type (language) of source code, if any

.construction

How built? AMOS, SAS/C, DICE, Modula-2, Oberon, Assembler, ...

.tested

Tested on what type of systems

.run

See above

.docs

Filenames of documentation

.described-by

Who wrote this description?

------------------(Delete this line and all text below)------------------

Suggested keywords for the TYPE field:

Action Game Animation Animation Player

Animation Tool Archiver CLI Tool

Communications Compiler Compression

Database Disk Tool Display Commodity

Drawing Image Conversion Image Processing

Library Mouse Commodity Music Composition

OS Utility Painting Picture

Printing Sound Analysis Sound Editing

Sound Playing Spreadsheet Strategy Game

Text Text Editing Text Viewer

Thinking Game Word Processing Workbench Tool

Keywords for the DISTRIBUTION field:

Commercial Commercial software is owned and distributed

through licenses. It costs money to individual

end-users and is not freely distributable.

SUCH PIECES SHOULD NOT APPEAR ON DISKS THAT ARE

MEANT FOR FREELY DISTRIBUTABLE SOFTWARE!

Commercial Demo Represents a demonstration of a commercial

package. As such, commercial demos are freely

distributable and may have limitations as

described in the .limitations field.

Shareware Such software is owned and copyrights are

held by the author(s). The software may be

distributed freely, but not sold for profit,

of course. Shareware often specifies a limit

of some time after which you are requested or

required to register the software (i.e. pay

for it.) This provides you with the means to

evaluate the software thoroughly before paying

for it.

Freeware Such software is owned and copyrights are

held by the author(s). The software may be

distributed freely, but not sold for profit,

which would mean the software is no longer

FREEware. No payments are required for such

software.

Public Domain Software labeled PD (Public Domain) belongs to

the public, i.e. ANYONE. Some people release

their software into the public domain with the

mistaken idea that they can continue to own

and control the program. Not so. Software

that is labeled Public Domain (or said by the

author to be released into the public domain)

truly belongs to anyone and everyone. It is

quite legal for someone to take such a program

and sell it for profit as is. Likewise, it

it perfectly acceptable to modify public domain

software to build a better product (or whatever)

out of it and then sell it for profit.

GNU Public License The terms and conditions of this license

are long and not easily reproduced here. Suffice

to say that software released under the GNU Public

License cannot be sold for profit and must be

distributed with source code. They are not

public domain, however.

1.26   2.8 KingFisher Tooltypes

KingFisher first processes the contents of the KingFisher2.prefs file for which it looks in the current directory first, then in ENV:KingFisher/ and last in S:

Once this file has been processed, KingFisher will process command line arguments (if invoked from the CLI) or Icon Tooltypes (if invoked from the Workbench.) The format of both tooltypes and CLI arguments are the same, and can be anything you find in the KingFisher2.prefs file (which is written each time you quit KingFisher.) You may also use the following:

SERVERNAME=volume:path/KFServer

If the KFServer is not currently running, KingFisher will attempt to start it by running "KFServer" in the current directory. If this is not how you have configured your system (the installation script set up things this way, so you ordinarily should not have to worry about this) then you must specify the full path and filename of the KFServer executable.

NOOUTPUT

A flag that tells KingFisher not to print the initial copyright and welcome banner. When invoked from the Workbench this banner causes a console window to open, which may well be undesirable. This option is set by default in the icon supplied with KingFisher.

1.27   2.9 Differences from KF 1.40

With this section I hope to present some information that will help you if you are a user of KingFisher 1.40 or earlier. There are some important changes that may not be always apparent:

The obvious:

KingFisher Release 2 requires at least V37 (KS 2.04); V38 (KS 2.1) is recommended because you get faster ASL requesters and, more importantly, a Screen Mode requester, without with the selection of custom screens may be a difficult undertaking. KingFisher will not work with V36 (early KS 2.0) software, nor will it function with KS 1.3. KingFisher 1.40 is the latest version that will work with those earlier operating system releases.

KingFisher Release 2 opens on a resizable window on the default public screen (usually the Workbench) and can be placed on a public screen if you so desire. It uses the screen's default font (although a different one can be selected) for all text, gadgets, and menus.

KingFisher Release 2 provides distinct browse and search gadgets. The search gadgets have arrows as well as question marks imprinted on them.

The not-so-obvious:

When used with CD-ROM databases, all items are indicated to reside on "Disk 1," and all gadgets referring to motion from disk-to-disk are disabled. This behavior comes from the fact that CD-ROMs are rather large and not broken into hundreds of disks, such as the 1000 Fish disks that Fred Fish has released over the years. The default Display Format includes a STORED-IN field, which will allow you to determine where on a CD-ROM the software is stored. If it references data that is part of Fred's Fish Disk collection, then you will notice a path similar to ...f9/ff951/... which references Fish Disk 951.

Search expressions have a new syntax so you can reference information from individual database fields. Briefly, this requires you to specify the name of the field plus an operator symbol (such as =, !=, $, etc) before the operational value. If you find this format too cumbersome and its functionality more than you need, you can switch KingFisher to use the older KingFisher 1.40 expression syntax with the Search Preferences menu item Simple substrings .

Instead of a fixed number (6) of Search Expression gadgets, KingFisher now maintains a behind-the-scenes history of expressions from which you can select with the Search Expression History gadget. The maximum size of this list (initially 20 entries) can be altered by modifying the KingFisher2.prefs file.

KingFisher Release 2 will no longer recognize the old "Add Fish" format used by Release 1. Instead it recognizes only the format specified by the new Product-Info Specification v6 which has specific provisions for storing database records so that they can be extracted from within other text. KingFisher's Export command can be configured to either use the Export Format (to be used in a document or letter, for example) or this re-importable format which is recognized by any Product-Info compliant software. The Use importable raw format Export Preferences menu permits this switch.

KingFisher Release 1 would highlight the keywords in the displayed text when a search found a record. As Release 2 now uses standard GadTools gadgets rather than a "home-cooked" listview (with all its limitations) this functionality has been lost. I cannot project if and when it will be implemented again. :(

Further details are treated in the Concepts section.

1.28   2.10 SPECIAL

1. KingFisher (the GadTools interface) operates through the KFServer program which it starts in the background. If the KFServer is already running, KingFisher will simply attach to the server swiftly; otherwise it will start the server.

2. You can force the KFServer to keep running even after the last client has detached. This dramatically speeds up the time needed to connect to the server in a future session, as the server needs not be loaded, and needs not initialize its databases. See the keep-running preference option in the KFServer.prefs file for more information.

3. The KFServer performs a "sanity check" on any INRAM database it loads. It can do this very quickly, with little or no noticable delay. It will not by default perform this operation on an ONDISK index, however, for two reasons: the time to perform the needed disk I/O could take a great deal of time, and if you choose an ONDISK index over a faster INRAM index, you are probably doing this for the reason that the index is rather large to begin with. If you absolutely must perform the sanity check on an ONDISK index, too, you can set the configuration option "sanity-check=always.

1.29   3 MENUS

NOTE: Most, if not all menu items support pressing the HELP key while pointing at the menu item to obtain assistance on the indicated menu item.

Project

Edit

Search

Preferences

Help

1.30   The Project Menu

About KingFisher

Server status

Open database

Install database

Define database

Print

Release printer

Export

Close export file

Quit

1.31   The Edit Menu

Append fish from file

Append fish from tree

Delete fish

Reconstruct database index

Build VersionLinks

Edit expression

Edit search masks

Edit version links

1.32   The Search Menu

Select expression

Search backward

Search forward

Load search set

Save search set

1.33   The Preferences Menu

Global

Auto-save on exit

Confirm quit

Display

Load custom display format

Show all fields in record

Drop custom display format

Custom screen

Default public screen

Center main window

Frame groups

Sticky result window

Display database title

Smart refresh

Printing

Load custom print format

Drop custom print format

One fish per page

Avoid page breaks

Add index info

Exporting

Load custom export format

Drop custom export format

Choose export file

Use importable raw format

Add index info

Searching

Stop on each match

Case sensitive

Trim blanks

Simple substrings

Use search masks

Save Preferences

1.34   The Help Menu

Help (index)

Using KingFisher

Searching

Printing

Exporting

Databases

1.35   PROJECT/About KingFisher

Presents an image of the KingFisher logo as well as copyright information for the software. Also given will be the registration site most likely to apply to you and an email address where to obtain technical support.

The same window will always appear when you first start KingFisher. It will go away by itself only if it is not deactivated.

Language translations:

Dansk (Danish) Finn Kettner <flynn@scala.ping.dk>

Deutsch (German) Uwe Schürkamp <hoover@mathematik.uni-bielefeld.de>

Nederlands (Dutch) Marcel Offermans <marcel@dutw30.tudelft.nl>

Español (Spanish) Eduardo Delgado <tdatos@cpd.uva.es>

Suomi (Finnish) Janne J Kalliola <plastic@vipunen.hut.fi>

Svenska (Swedish) Jan Simonson <Jan_Simonson@augs.se>

1.36   PROJECT/Status

Requests from the server some information, which includes an estimate of what percentage of the server's total time your client has taken. It also informs you of which database you have open.

If you issue this command with a shift key or the middle mouse button (MMB) held down, you will receive the same listing as KFServer would produce when invoked with the STATUS parameter.

1.37   PROJECT/Open Database

Requests from the server a list of all available databases. This is a list of the descriptions in all files with the extension .kfdb that the server knows about. The server can see these files only in its default directory.

You get to select one of these databases based on the description for the database as stored in the .kfdb files. KingFisher will save the position in your current database and activate the newly selected database, moving to the most recently visited record in that database.

The window that lists you all the available databases becomes far more useful when you have more than one or two databases available to you.

You can cancel the selection by closing the window.

1.38   PROJECT/Install Database

Allows you to select a .kfdb file from another disk volume (such as a CD-ROM or floppy disk) and install this in the server's directory, thereby making the database available for immediate use.

There are some points of which you should be aware:

· Only the .kfdb file is copied, no index or data files. Very little space is used by the .kfdb file: only ½KB on average!

· If the .kfdb file does not specify complete paths to its data and index files, then KingFisher will add the path where the .kfdb file was copied from. In most cases this will make the database available for immediate use!

· If installing a CD-ROM database, you may wish to manually copy the index files to a writable volume and adjust the .kfdb file to reflect this. Although KingFisher will think you can then add records to the database when you cannot actually do this[1], you will still be able to make changes to the index (i.e. version links and flags can be modified!)

_____

[1] A future version of KingFisher may distinguish between the index being writable but not the database.

1.39   PROJECT/Define Database

NOTE: This command is not yet available. The following is what you need to know to setup your own database, manually:

KFServer can only serve databases that are defined by the contents of files with a .kfdb extension. The exact name of this file is immaterial but it is always a good idea to use a sensible name. Let us setup a database for your Amiga Club, using a single file to store all the information, named ClubDisk:Club.data, an index file for it named ClubDisk:Club.index, and a QuickIndex file based on the name field that is stored in ClubDisk:Club-name.index:

The name of the KingFisher Database file shall be AmigaClub.kfdb

Let us create this file with the following contents. You can use any standard text editor for this task:

database-name=Our Outstanding Amiga Club's Own Software Collection

section=00000,99999,ClubDisk:Club.data

index=inram

index-increment=100

index-name=Club.index

field-index-field=name

field-index-name=ClubDisk:Club-name.index

keep-open=no

read-only=no

For more information on these individual items, please see the last portion of the KFServer section.

1.40   PROJECT/Print

Using the currently active print format (default or custom), KingFisher will print data to the printer. If you print from the main window's menu, KingFisher will print only the current record. If you print from the "Caught Fish" window that displays all matching records in the Search Set , then KingFisher will print all records in the search set.

Notice that printing is configurable with the options of the Printing Preferences menu.

1.41   PROJECT/Release printer

This entry is active only when KingFisher has printed something, after which it will retain "ownership" of the printer device awaiting more print commands. Using the Release printer command returns the printer to the system and tells KingFisher that you are done with printing for the moment.

The current page in use will be ejected by this command.

1.42   PROJECT/Export

Using the currently active export format (default or custom), KingFisher will write data to the export file. If you export from the main window's menu, KingFisher will write only the current record. If you export from the "Caught Fish" window that displays all matching records in the Search Set , then KingFisher will write all records in the search set.

Notice that exporting is configurable with the options of the Exporting Preferences menu.

If exporting is set to Use importable raw format , then neither the default, nor the custom format will be used, and instead KingFisher will write a file that can be re-imported through the Append fish from file command.

1.43   PROJECT/Close export file

This entry is active only after KingFisher has exported something and is keeping the file open and ready for further additions through the Export command. Using the Close export file command closes the file and allows you to access it through other software.

1.44   PROJECT/Quit

The Global Preferences submenu item Confirm quit allows you to specify whether or not you wish KingFisher to ask you if you really want to quit. If you find yourself frequently quitting KingFisher without meaning to, you should turn that option on. If the "Really quit KingFisher" requester goes on your nerves, turn the option off.

If you also have the Auto-save on exit option disabled, you must make this change permanent by selecting Save Preferences .

1.45   EDIT/Append fish from file

The file you specify may contain one or more records. The records must conform to the Product-Info Specification v6

All valid records from the given file will be appended to the database. The index is automatically updated (and saved to disk when the database is closed.)

1.46   EDIT/Append fish from tree

Scans a directory tree for #?.pi, .Product-Info, and Product-Info files and adds their contents to the database. A status window keeps you informed of progress. You can interrupt the scan by closing the status window; you must confirm such an action before the scan is actually aborted.

The index is automatically updated (and saved to disk when the database is closed.)

1.47   EDIT/Delete fish

Truncates the database by deleting the current fish (record) and all that follow. You must confirm the action before it will take place.

That database files themselves are not (at this time) physically altered. Only the index is altered (and this change made permanent when the database is closed.)

Please note that if you use the "EDIT/Reconstruct database index" command after deleting records from the database, KingFisher may get confused about where the last record in the database actually ends. In the best of cases, you will simply "recover/un-delete" the deleted records; much of the time, however, you may end up with junk at the end of the database which can cause a wide variety of problems. If this occurs, your first action should be to delete those junk records that lie beyond the actual end of the database.

1.48   EDIT/Reconstruct database index

This command asks the server to throw away all indexing information on the current database and scan the database file(s) in an effort to reconstruct the information necessary to access individual records.

Both the standard index, and the QuickIndex is rebuilt by this command. The QuickIndex is usually built on the 'name' field, which makes certain other operations (such as building of version links) possible.

WARNING: The index stores the version links (previous and following version) as well as the flags (deleted, hidden, owned, marked, etc.) If you reindex a database you will lose both of these sets of information! Use the Build VersionLinks command to rebuild the version links, too.

A database can be reindexed only if you are the only client using it, and if the database is writable.

1.49   EDIT/Build VersionLinks

Cross-references all programs in the currently selected database and builds links between those programs that have the same names. This permits you to skip from one release of a program to another, using the Version Browsing gadgets.

This command requires that a QuickIndex on the 'name' field exists.

Available options[1] are:

[x] Clear existing links

Removes all previously set links rather than adding to and overwriting them. This option is checked by default only if it makes sense to remove old versions. You are encouraged to accept this default!

[x] Eliminate (Roman) numeric additions

Programs such as "bBaseII" and "bBaseIII" or "DiskSalv" and "DiskSalv2" are considered the same.

[x] Double-check by authors[2]

A program "MonkeyCommand" by Antonio Spaghetti, and "MonkeyCommand" by "Ricky Rocket" are considered distinctly different.

[x] Sort links by version

CD-ROM disk's databases may not necessarily list software in version order. This will examine the 'version' fields and order the links so that earlier versions (even if stored later in the database) are at the start of the version list.

Sorting links depends heavily on correct version numbers in the database (i.e. the .version field must be correctly formatted, according to the Commodore Style Guide) and is not available with a older Release 1 databases (the option will be disabled in that case.)

VersionLinks can be build only if you are the only client using this database, and if it is writable.

NOTE: In versions earlier than 2.10, KingFisher relied on an $O(n^2)$ algorithm to perform this operation. A vastly superior algorithm was created for KingFisher 2.10 which rebuilds the links in $O(n)$ time!

---

[1] Some or all of these options may be unavailable until implemented in later releases.

[2] Use of this option will slow down processing dramatically!

1.50 EDIT/Pack database

This command doesn't exist yet.

1.51 EDIT/Edit custom format file

Use of this command will suspend KingFisher until you are finished with the program it invokes. Selecting an existing (or new) .format file will then load this file into your favorite text editor and allow you to alter it.

If the environment variable EDITOR is not set (i.e. 'ENV:EDITOR' does not exist) then KingFisher will use 'c:ED' instead to edit the .format file.

If you are running KingFisher on a custom screen instead of the default public screen, then you should look for the editor's window on the default public screen.

NOTE: If you edit the display format file currently in use by KingFisher, you must manually reload this file. A future release of KingFisher will use file notification to assure that any alteration to the file (even when not modified with KingFisher's knowledge) will automagically update the display.

1.52 EDIT/Edit search expression

Edit the current search expression.

1.53 EDIT/Edit Masks

Search Masks allow you to explicitly exclude certain records, depending on what "Flag Bits" they have set. The Search Masks are used only if you have selected the Use Search Masks option.

There are two separate Search Masks:

Avoid Mask

This mask is used to completely eliminate a record from the search. Any record (fish) that has any of these flags set, will be excluded. By default this includes all records with the Deleted (D), Owned (O), and Hidden (H) flags.

Match Mask

This mask is used to choose from those records not already eliminated by the Avoid mask. Only records are chosen which have one or more of these flags set.

NOTE: Fish (records) without any flags set cannot be located with a Search Mask in use!

Why two search masks?

The Avoid mask allows you to specify which records you definitely don't want to see, while the Match mask handles the remainder. This provides greater flexibility. If KingFisher had only the Match mask, you would not be able to eliminate records from your search which have been marked for deletion, been marked hidden, or those you marked as already in your collection (owned.)

Click here to learn about the meaning of each flag.

1.54 EDIT/Edit Links

This command directly modifies the Version Links that KingFisher uses when you click on one of the Version Browse gadgets. The window that opens up contains two numeric gadgets, one for the preceding, and another for the following link.

1. Select this command for the program whose Version Links you wish to modify.

2. Use any other function in KingFisher to locate the matching preceding or following version to the current program,

3. Press the "Set" gadget or enter the appropriate record number,

4. Repeat steps (2) and (3) for the other link (if necessary)

5. Click on "Accept" to write the selections to the database or "Cancel" to not keep the changes you made.

The Edit Links command is not available when the current database cannot be modified (stored on read-only media or the index is write protected.)

1.55 EDIT/Copy to clipboard

Copies the current record in its raw, re-importable format to the clipboard. This creates an IFF FTXT in unit 0. The next 'Copy to clipboard' operation will overwrite the previous clipboard contents.

Multiple records can be saved to the clipboard from the Search Result window.

1.56 EDIT/Append from clipboard

The contents of each IFF FTXT found in the clipboard's unit 0 is sent to the KFServer for processing. If it contains a valid record, then it is appended to the current database as a new record.

1.57 SEARCH/Select Expression

If one or more Search Expressions have been used before, you can select one of them to be placed into the Search Expression gadget and used for the next search you begin.

1.58 SEARCH/Search backward

Begins a search in reverse direction. The Stop on each match option determines if the search will stop as soon as a match is found, or if it should continue to build up a Search Set consisting of all fish (records) that match the expression.

You can interrupt a search by closing the Search Status window.

Notice that you can press the "<" key as a short cut for this command.

1.59 SEARCH/Search forward

Begins a search in forward direction. The Stop on each match option determines if the search will stop as soon as a match is found, or if it should continue to build up a Search Set consisting of all fish (records) that match the expression.

You can interrupt a search by closing the Search Status window.

Notice that you can press the ">" key as a short cut for this command.

1.60   SEARCH/Load search set

Loads a new Search Set . Any Search Set that you have currently loaded will be cleared and is lost if it has not been saved.

When a Search Set is loaded, KingFisher may switch to the database to which the search set applies, and will also store the Search Expression to the Expression gadget to give you an idea what the Search Set means.

While loading the Search Set, KingFisher will retrieve some information from the appropriate database to be shown to you in the Search Set Window. This process requires KingFisher to read from the database. Larger Search Sets may not, therefore, seem to load instantly.

If your Search Sets are not given the extension .search on disk, then you must alter the ASL File Requester's Pattern field from the default pattern #?.search to something closer to your needs.

1.61   SEARCH/Save search set

Saves the current Search Set to a file on disk so it can be retrieved later, thereby saving you the time and effort of executing another search and having to wait for the result again. Search Sets require approximately 5 bytes per record on disk, so that 100 matching records will not require more than approximately 500 bytes on disk.

If you give you search sets the extension .search, then KingFisher will automatically show you existing Search Sets when you load a Search Set!

1.62   PREFERENCES/Global

This submenu allows you to somewhat alter KingFisher's behavior:

Auto-save on exit

When checked, automatically saves all settings when you exit.

Confirm quit

When checked, requires confirmation before actually quitting.

1.63   PREFERENCES/GLOBAL/Auto-save on exit

If you enable this option, then KingFisher will automatically store all settings to the KingFisher2.prefs file in the default directory, or the file named by the SETTINGS tooltype , or the first file named KingFisher2.prefs it finds while looking in the default directory, then ENV:KingFisher/, and then S:

If you turn off this option and wish this change to become permanent, then you must use the Save Preferences command, otherwise your change will not be saved when KingFisher exits!

1.64   PREFERENCES/GLOBAL/Confirm quit

Do you hate software that just always asks you if you really want to quit, and you hear yourself mumbling "Of course, I'm sure!"

Or do you tend to click on the close gadget and then find yourself saying "oops!" but it's too late?

Whichever of these describes you, with the "Confirm quit" option you can get KingFisher to behave the way you want it to!

1.65   PREFERENCES/Display

This submenu allows you to define display-related things:

Load custom display format

Loads a display format that you have defined and stored in a file.

Show all fields in record

Displays all, even empty and non-standard fields, part of each record.

Drop custom display format

Reverts to the internal default display format.

Custom screen

Opens KingFisher's window on a custom screen of your own choosing.

Default public screen

Opens KingFisher's window on the default public screen.

Center main window

KingFisher keeps the main window always centered on the screen.

Frame groups

Nicely frames related groups of gadgets.

Sticky result window

Search Result Window "sticks" to edge of main window.

Display database title

Current database title is added to top of main window.

Smart refresh

Windows are handled in Intuition's faster SmartRefresh mode.

1.66   PREFERENCES/DISPLAY/Load custom display format

The internal default display format is merely one way of displaying the information in the database. Different types of databases, in fact, may require different display formats.

Custom display formats are kept in files on disk, and KingFisher remembers the display format in use for every database.

To remove a custom display format and revert to the builtin default, select the Drop custom display format command.

To create or edit a custom format, use Edit custom format file .

1.67   PREFERENCES/DISPLAY/Show all fields in record

Creates a custom display format based on each record's own fields and uses that to create the display of information. If you have a custom display format selected, it will be replaced by this command. To return to the default display format, select the Drop custom display format command; to return to a custom display format, select a new custom display format.

Notice that this command will display non-standard fields as well as fields ordinarily suppressed because they are empty.

1.68   PREFERENCES/DISPLAY/Drop custom display format

This command is available only when a custom display format has been loaded. It removes a previously selected custom display format and reverts back to the builtin default.

1.69   PREFERENCES/DISPLAY/Font

You may select any font for KingFisher to use. Be advised that KingFisher cannot yet fall back to a smaller font if you select an unreasonably large font.

The font you select may be either fixed or proportional.

1.70   PREFERENCES/DISPLAY/Custom screen

This option will only work if you are running Kickstart 2.1 (V38) or later because it uses the Screen Mode Requester from the V38+ ASL Library. If you are still running Kickstart 2.04 (V37) click here for assistance in building getting KingFisher Release 2 to open on a custom screen.

1.71   Getting KingFisher to open on a custom screen under V37

First of all, quit all copies of KingFisher so that your changes to the KingFisher2.prefs file are not accidentally overwritten again when you later quit the program.

You need to edit the KingFisher2.prefs file and locate the "Screen" item, which should be one of the very first ones listed. The following example uses a PICASSO:1024x768 screen with 256 colors:

Screen=CUSTOM:40020006,1024w,768h,8d,1

Notice that the first portion of the Screen specification is "CUSTOM:" If anything else shows up in the first 7 characters, then KingFisher assumes that you are actually specifying the name of a specific public screen, rather than a custom screen specification.

The next item, "40020006" is a hexadecimal representation of the screen mode ID from the Display Database. The following are standard values that you can use as a starting point.

Mode ID Resolution (PAL) Description

---------------- ---------------- ----------------------

00008000 640 x 200 (256) Hires

00008004 640 x 400 (512) Hires-Interlace

00008020 1280 x 200 (256) 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

---------------- ---------------- ----------------------

The next three items, "1024w", "768h", and "8d" are probably quite obvious to you: they are the width, height, and depth of the screen. The letters that follow the numbers are stored in the file only to more easily describe their purpose. The trailing "1" indicates that autoscrolling is enabled for the screen ("0" disables it.)

Here is an entry for an 800x600 Hires-Interlace NTSC screen using 4 colors (2 bitplanes) and with autoscroll enabled:

Screen=CUSTOM:00008004,800,600,2,1

1.72 PREFERENCES/DISPLAY/Default public screen

Choosing this item causes KingFisher to open on the default public screen. If you have no public screens open, then the Workbench is the default public screen.

If you select a different screen to be your public screen, then this command will cause KingFisher to jump to the new public screen.

1.73 PREFERENCES/DISPLAY/Center main window

Ordinarily, KingFisher opens its window always in the same position where you had it last open, meaning that you can arrange your desktop in some way for KingFisher to fit in nicely.

You may, however find that it would be more convenient for KingFisher to be always available in the very center of the screen, regardless of the screen's current resolution and KingFisher's window size.

With the 'Center main window' option, KingFisher makes an effort to always place the window in the center of the display. And if you resize the window, KingFisher will automatically recenter the window for you! If, however, you then move the window by dragging it to a new position, KingFisher will turn off the 'Center main window' option to accommodate your desire to place the window in a specific position.

1.74 PREFERENCES/DISPLAY/Frame groups

When this option is selected, KingFisher will place "recessed" frames around groups of related gadgets to more clearly denote their relationships to each other. This will cause a somewhat more "cluttered" display and may not be pleasing to everybody, which is why it has been made an option.

1.75 PREFERENCES/DISPLAY/Sticky result window

When this option is selected and a Search Result window is open (see Stop on each match) then moving or resizing the main window causes the Search Result window to be reposition, too. In effect, the Search Result Window "sticks" to the right edge of the main window as much as possible.

1.76 PREFERENCES/DISPLAY/Display database title

When this option is selected, the title of the current database is displayed above the main text display area of the main window. If you rarely work with multiple databases and you are running on a relatively small display screen, you may wish to leave this option disabled to preserve valuable screen space.

1.77   PREFERENCES/DISPLAY/Smart refresh

When this option is selected, KingFisher's windows will (whenever needed) be refresh by Intuition's own buffers. This can be significantly faster on screens with little depth, but on deep screens (such as a large 256 color Workbench) this not only slows down the refreshing but also requires Intuition to set aside enough memory for the refreshing operation to work.

Although the display may not always be updated at once when portions of the window are "damaged" by obscuring windows, turning this option off can significantly improve the display's performance and reduce the program's memory requirements. Experiment!

1.78   PREFERENCES/Printing

This submenu allows you to define printing-related things:

Load custom print format

Loads a print format that you have defined and stored in a file.

Drop custom print format

Reverts to the internal default print format.

One fish per page

Prints no more than one fish (record) per page.

Avoid page breaks

Starts a new page if a record no longer fits on the current page.

Add index info

Adds information from the index to each printed record.

1.79   PREFERENCES/PRINTING/Load custom print format

The internal default print format is merely one way of formatting the information in the database for the printer. Different types of databases, in fact, may require different print formats.

Custom print formats are kept in files on disk, and KingFisher remembers the print format in use for every database.

To remove a custom print format and revert to the builtin default, select the Drop custom print format command.

To create or edit a custom format, use Edit custom format file .

1.80   PREFERENCES/PRINTING/Drop custom print format

This command is available only when a custom print format has been loaded. It removes a previously selected custom print format and reverts back to the builtin default.

1.81   PREFERENCES/PRINTING/One fish per page

By enabling this command, each record (fish) is printed beginning at the top of a new page.

This command is unavailable while the printer is in use by KingFisher. You need to first Release printer before you can change these settings.

1.82   PREFERENCES/PRINTING/Avoid page breaks

The effect of this command is, perhaps, most easily described at hand of a little diagram to compare the effect visually. The idea is to prevent descriptions from being broken up by page breaks, forcing a record which will not fit on the current page to begin at the top of the next page:

NO Avoid page breaks YES Avoid page breaks

_____ _____

|Name: KingFisher | |Name: KingFisher |

|Vers: 2.0 | |Vers: 2.0 |

|Text: blah blah blah | |Text: blah blah blah |

| blah blah blah | | blah blah blah |

| blah blah. | | blah blah. |

| | | |

|Name: MonkeyCommand | |Name: MonkeyCommand |

|Vers: 1.0 | |Vers: 1.0 |

|Text: blah blah blah | |Text: blah blah blah |

| blah. | | blah. |

| | | |

|Name: MonkeyCommand | | |

|Vers: 2.0 | | |

|Text: blah blah blah | | |

|--------------------| |--------------------|

| blah blah blah | |Name: MonkeyCommand |

| blah blah blah.| |Vers: 2.0 |

| | |Text: blah blah blah |

|Name: B5-Images | | blah blah blah |

|::::::::::::::::::::| | blah blah blah.|

| |

|Name: B5-Images |

|::::::::::::::::::::|

This command is unavailable while the printer is in use by KingFisher. You need to first Release printer before you can change these settings.

1.83   PREFERENCES/PRINTING/Add index info

Index information is added to the printout for each record in the following format:

.INDEXINFO=|DISK=1|FISH=17|FLAGS=8001|

This format will become a standard for a future Product-Info Specification and will be recognized by KingFisher's "Add Fish..." command.

This command is unavailable while the printer is in use by KingFisher. You need to first Release printer before you can change these settings.

1.84   PREFERENCES/Export

This submenu allows you to define exporting-related things:

Load custom export format

Loads an export format that you have defined and stored in a file.

Load custom export format

Reverts to the internal default export format.

Export filename

Specifies a (new) file to which exported records are written.

Use importable raw format

Uses a format which the Append fish from file command understands.

Add index info

Adds informaiton form the index to each exported record.

1.85   PREFERENCES/EXPORTING/Load custom export format

The internal default export format is merely one way of formatting the information in the database for export files. Different types of databases, in fact, may require different export formats.

Custom export formats are kept in files on disk, and KingFisher remembers the export format in use for every database.

To remove a custom export format and revert to the builtin default, select the Drop custom export format command.

To create or edit a custom format, use Edit custom format file .

Note that the Use importable raw format option, overrides the custom export format completely.

1.86   PREFERENCES/EXPORTING/Drop custom export format

This entry will only be available if you have a custom export format loaded. It will drop the custom format and revert back to the default. Note that the use of the Use importable raw format option overrides the use of custom or default formats entirely.

1.87   PREFERENCES/EXPORTING/Export filename

By default, the export filename, if you never specify a different name, is t:KF2.output. If you prefer a different filename, this command will let you do so, and KingFisher will remember the name between sessions. An implicit Close export file will be issued for you.

1.88   PREFERENCES/EXPORTING/Use importable raw format

Forces the output to be in a special, re-importable format. The file can be transmitted via electronic mail (although national characters may not be preserved by the email transmission!) and can be added to any KingFisher Release 2 database through the Edit menu's Add fish from file command.

Notice that while this option is selected any custom export format is effectively disabled.

1.89   PREFERENCES/EXPORTING/Add index info

Index information is added to the export file for each record in the following format:

.INDEXINFO=|DISK=1|FISH=17|FLAGS=8001|

This format will become a standard for a future Product-Info Specification and will be recognized by KingFisher's "Add Fish..." command.

1.90   PREFERENCES/Searching

This submenu allows you to define searching-related things:

Stop on each match

When checked, stops on each match; otherwise builds a Search Set.

Case sensitive

When checked, upper and lower case characters become distinct.

Trim blanks

When checked, will remove trailing blanks from search strings.

Simple substrings

When checked, uses the simpler KingFisher 1.x expression syntax.

Use Search Masks

When checked, uses the search masks defined by Edit Search Masks

From here, you can also learn more about Search Expressions .

1.91   PREFERENCES/SEARCHING/Stop on each match

When you begin a search, KingFisher examines this option to see if you wish it to stop immediately whenever it finds a match. If this option is not enabled, KingFisher will build a Search Set instead, presenting you with the final list of all matches, which you can save permanently, and from which you can choose randomly.

1.92   PREFERENCES/SEARCHING/Case sensitive

When this option is enabled, upper and lower case letters are treated as distinct symbols, so that "a" is not the same as "A". If, for example, you are looking for references to Kickstart and your search string consists of "KS" (abbreviation for Kickstart) you might be looking explicitly for only the all upper case version, and have no desire to locate words like these, too: ticks or packs.

1.93   PREFERENCES/SEARCHING/Trim blanks

When blank spaces are typed into a string gadget, at the end of a string, they are usually quite invisible and difficult to detect. Their presence, however, can produce rather puzzling results because they may end up being considered part of a string constant in your expression!

Enabling this command will guard against such troubles by removing all blank spaces from the end of your expressions. This option will NOT affect quoted strings, such as "fred ".)

1.94   PREFERENCES/SEARCHING/Simple Substrings

If the Simple Substrings option in the Searching Preferences is selected, KingFisher will automatically supply a field and operator selection of "*$" to your search strings, so that the substrings you provide in the style of KingFisher release 1 expressions are treated as substrings and scanned for in every available field of the database records that will be examined during a search.

1.95   PREFERENCES/SEARCHING/Use search masks

Enabling this option causes searches to use the masks defined by the Edit Search Masks command in order to eliminate certain types of records from the search.

1.96   PREFERENCES/Save Settings

Saves all settings to a file of your choosing. Unless given a specific filename with the SETTINGS tooltype at startup, KingFisher for the following files from which to read its settings:

KingFisher2.prefs (in the current directory)

ENV:KingFisher/KingFisher2.prefs

S:KingFisher2.prefs

If it finds one of these files, it will attempt to writes its settings back to this file when you exit (provided the Auto-save on exit option is enabled) or to the first file in that list (i.e. in the default directory when none of these files have been found.

You can save settings with this command to any file of your choosing but KingFisher will not be able to find and actually use the file unless you are saving it according to the above specifications.

1.97   HELP/Using KingFisher

KingFisher uses a Graphic User Interface (GUI) to allow you to easily browse through one or more databases. The individual gadgets are usually referred to by a name which is most easily presented at hand of a picture which you can view by clicking here or by viewing the KF-GUI.iff image with an external viewer.

The gadgets in the upper left corner allow you to select a particular record in the database (if the cycle gadget is set to "Fish") or jump to the first record on any particular disk. Notice that the database must actually contain multiple disks. A CD-ROM is a single disk and rarely organized into multiple disks, so the Disk number will never change from 1, which also affects some other functions of KingFisher.

The gadgets in the upper right corner are flags which are particular to each record in the database. The first row of these gadgets is defined by KingFisher, to provide you with flags to mark records as DELETED, OWNED, or HIDDEN, or MARK them for later retrieval. The gadgets with numbers in them are for you to define.

The three pairs of gadgets in the lower left are the browsing gadgets that permit you to jump through the database either from one disk to the next (unavailable if the database records the contents of only a single disk) or by record (fish.) The pair between these two are the VersionLink gadgets, which allow you to select previous or later versions of the displayed entry if such version links have been set in the database.

If a record has no previous or following versions stored in its internal version link fields, then these gadgets will be unavailable (ghosted.) A future version of KingFisher will allow you to edit the links directly, and you can use the Build VersionLinks... command to create links automatically.

The lower right corner has search tools at your disposal. The Expression gadget stores the currently active Search Expression which is compiled when you select one of the two Search Gadgets (the arrow gadgets that also have question marks in them) to initiate a search.

KingFisher maintains a history of the most recently used search expressions. You can select from this list by pressing on the gadget with the artistically deficient image of a scroll. This will open a little window with previously used search expressions.

Click on one of these to have it automatically placed into the Expression gadget. You can press the Escape key to cancel the selection window.

Whenever you initiate a search, KingFisher examines the state of the Stop on each match Search Preferences menu to determine if you want KingFisher to stop immediately when it finds a match or instead scan the whole database before presenting you with a window from which you can select from among multiple matches. When the "Stop on each match" entry is not checked, such a Search Set is built. These Search Sets can be stored on disk and later retrieved to save you the time to perform the same search over and over again. To open and close the Search Result window, which contains a representation of the Search Set, click on the gadget with the images of Fred Fish's Fish Logo (fish bones.)

The center piece of the GUI is a so-called "listview" which is a scrolling display of text, describing the current fish (record.) The vertical slider at the right edge lets you scroll through the text.

Here are some neat things you may not realize at first:

You can resize KingFisher's window to make it as large as you like or as small as various GUI elements allow.

You can run multiple copies of KingFisher, perhaps two or three to search with, and another to continue browsing through the database.

Each copy of KingFisher can access a different database. You could run two copies of KingFisher to search two different databases, and scan a third with yet another copy of KingFisher.

You are not restricted to the display, print, and export formats that KingFisher uses by default. You can define your own Custom Format in a file and KingFisher will remember which database uses which custom formats!

1.98   HELP/Searching

The simplest way to search the database is by browsing through it one record at a time. Unless you are exceedingly patient and have too much time on your hands, you are best served by expressing to KingFisher what you are looking for:

1. In the lower right corner of KingFisher's window you will find a string gadget. Into this string gadget you should enter a Search Expression .

By altering the Simple Substrings option, you can make KingFisher use the older and simpler substring expressions. These, however, do not offer you the same flexibility as the new expressions.

Instead of entering an expression, you can also click on the gadget that is supposed to look like an scroll, provided that this gadget is not ghosted. This will open a listing of previously used expressions from which you may select one by clicking on it.

2. Once you have an expression typed into the Search Expression Gadget, you should click on one of the directional gadgets that have a question mark (?) as part of the design. This initiates a search.

If you receive an error requester instead, examine the message and make what corrections are necessary to produce a syntactically correct search expression .

3. You can search for one matching record at a time or have KingFisher scan the entire database and collect all matching records into a list that it then presents to you. If you expect many records to match your criteria you may wish to disable the Stop on each match option.

The great advantage of not stopping on each match but instead building a Search Set is that you can click on each item in the list to instantly view the record. In addition, you can save the Search Set to a file on disk and retrieve it later, thereby saving yourself the time required to search the database once again.

4. Other options are available to alter the behavior of a search.

1.99   HELP/Printing

When you print with KingFisher, the printer device is kept open so that you may add further records to the printout without having another program intersperse its output. This causes other programs to be unable to open the printer device, however, meaning that they cannot print until you quit KingFisher or use the Release printer command from the Project menu.

You can specify a different print format through the PREFERENCES/Printing menu, as well as dictate how records are broken up over page boundaries.

1.100   HELP/Exporting

Exporting records with KingFisher can mean two similar, but distinct operations: you may write records to a file the same way you might wish to print them on paper, or you export them so that they may be read back into KingFisher. Which way the Export command is to be used depends on how you configure the command through the PREFERENCES/Exporting menu.

The file to which you export can also be selected from that menu. Selecting a new file will automatically close the previously used file (if it was open.) The file to which KingFisher exports records is kept open much like the printer device is kept open between print operations. Use the Close export file command from the Project menu when you are finished with the file.

1.101   HELP/Databases

If you wish to make an existing database available to KingFisher, you have two options:

1. Install database... through KingFisher's menu command (see PROJECT menu) which handles the installation of a database fully automatically, even adjusting filename references to included the proper disk volume labels!

2. Perform the step manually -- this section also offers information on what changes may be needed to a .kfdb file to make it work when KingFisher itself fails to handle step (1) properly on its own, or when you need to create a new database.

1.102   CAUGHT FISH

This page unintentionally left blank. Please try the Index.

1.103   CAUGHT FISH/Close window

This will merely close the Search Set window. It will not lose the current Search Set. To open the Search Set window again, click on the icon which contains the "boney fish" symbols.

1.104   CAUGHT FISH/Apply Mask

This command applies a new mask to all fish in the Search Result window. More flexibility will be offered to this function in the future.

1.105   SEARCH EXPRESSION ERROR: Logical Operator Expected

You failed to provide a legal operator, meaning that KingFisher could not figure out in which way you want to combine two or more expressions .

Valid logical operators are:

& Logical AND (as in "if this AND that is true, then ...")

| Logical OR (as in "if either this OR that is true, then ...")

^ Exclusive OR (as in "if either this OR that is true, but NOT BOTH, then ...")

A unary operator may be used to reverse the value of an expression:

! Logical NOT. You may use the ~ character instead; your choice.

1.106   SEARCH EXPRESSION ERROR: Comparison Operator Expected

You failed to provide a comparison operator, meaning that KingFisher could not figure out in which way you wish to apply a value to a field.

Valid comparison operators are:

= Equality (as in "if the field contains exactly this value, then ...") You can also use two equal signs, which is what the C programming language uses for equality tests. The choice is yours.

ex: name = 'kingfisher'

>= Alphanumerically greater than or equal.

ex: version >= '2.0'

<= Alphanumerically less than or equal.

ex: date <= '1994.08.31'

> Alphanumerically greater than.

< Alphanumerically less than.

<> Not equal. This is the exact opposite of the '=' operator. You may use != instead, which is what the C programming language uses to test for inequality.

$ You should read this symbol as "contains the substring" so that the expression

ex: name$fish

This reads "If the name field contains the substring 'fish' then ..."

1.107   SEARCH EXPRESSION ERROR: Invalid comparison operator

The symbols you used do not make a valid comparison operator. Please note that the following are NOT valid:

>< => =< !< !> >$

Click here for a list of valid comparison symbols.

Click here for detailed information about creating a search expression.

1.108   SEARCH EXPRESSION ERROR: Mismatched Parentheses

Each open parentheses must be matched by exactly one closing parentheses. You have either used too many (symbols or too many) symbols. You may have forgotten to enclose in quotes the (or) symbols meant to be part of a string constant.

1.109   SEARCH EXPRESSION ERROR: Field identifier expected

KingFisher requires you to provide the name of a field in the database before you can give it a comparison or logical operator. If you need help constructing a legal search expression, click here .

1.110   SEARCH EXPRESSION ERROR: Unsupported Feature

You have hit upon an as-yet unsupported feature of the expression parser. This feature may become operational in the future to allow you to build more complex expressions with fewer symbols.

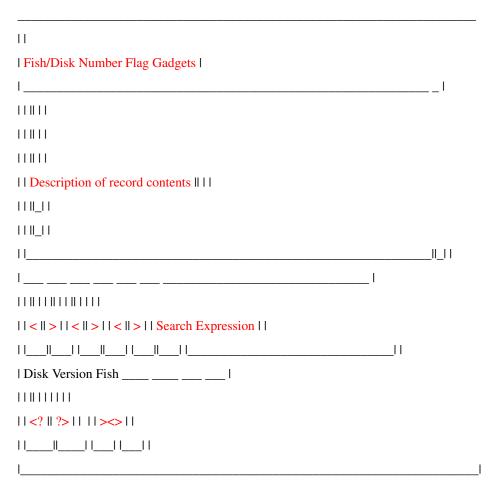## 1.111   SEARCH EXPRESSION ERROR: Internal Error

An internal error has occurred in the expression parser. Please write down the exact expression (with spaces and all characters trailing) to write to the author of KingFisher with this bug report!

1.112   SEARCH EXPRESSION ERROR: Incomplete Expression

The expression is incomplete. You must provide additional components before KingFisher can process the request.

1.113   4 GADGETS

The following is an approximate layout of KingFisher's display. The buttons below are placed to correspond with gadgets in the display. Choose any of them for explanations of their functions, or press the HELP key while the mouse pointer is over the gadget in the real window.

```
_____
| |
| Fish/Disk Number Flag Gadgets |
| _____ _ |
| | || | | |
| | || | | |
| | || | | |
| | Description of record contents || | |
| | || |_| |
| | || |_| |
| |_____|| |_| |
| ___ ___ ___ ___ ___ ___ _____ |
| | || | | || | | || | || | | || | |
| | < || > | | < || > | | < || > | | Search Expression | |
| |___||___| |___||___| |___||___| |_____| |
| Disk Version Fish ____ ____ ___ ___ |
| | || | | | | | |
| | <? || ?> | | | | ><> | |
| |____||____| |___| |___| |
|_____|
```

NOTE: yes, this looks a bit messed up, but I'll try to fix it soon... --Udo

1.114   GADGET: Home/End of Database (Buttons)

These gadgets immediately jump to the first or last record in the database. They are ghosted if you are already located on the first and/or last record.

1.115 GADGET: Browse across Disks (Buttons)

These gadgets jump over one or more fish (records) if the database index defines them to be located on multiple disks.

The original Fish Disks database that ships with KingFisher Release 2 defines 1000 such disks, but a CD-ROM usually consists of no more than a single disc, wherefore the distinction of separate disks is less clear, if even meaningless. In these cases you will find that all functionality referring to multiple disks becomes unavailable.

1.116 GADGET: Browse across Versions (Buttons)

These gadgets jump to previous or later releases of the same program, provided that information is recorded in the database index. If no such data is recorded, KingFisher has at the moment no way to provide you with this functionality.

The original Fish Disks database that ships with KingFisher Release 2 defines over 1000 linked versions, but with CD-ROM databases, KingFisher can no longer rely on quite the same format and usually has no such links available.

A future release of KingFisher will introduce more powerful VersionLink handling than earlier releases had. For the moment, you will find the links ready-set only in the original Fish Disks database.

One or both of these gadgets will be ghosted if a link in that direction is not available.

1.117 GADGET: Browse from Fish to Fish (Buttons)

These gadgets permit you to browse through the database one fish (record) at a time. These gadgets will be ghosted if you are at one end of the database and cannot move in that direction any further.

1.118 GADGET: Flag Gadgets (Toggle Buttons)

Each fish (record) in a KingFisher database has 16 special purpose flags associated with it. These are represented by two rows of 8 flags each. Their layout and appearance resemble the following diagram:

D _ _ _ _ O H M

8 7 6 5 4 3 2 1

The top row represents pre-defined flags; do not assign a different meaning to these flags than is given below:

D Deleted. This record is marked to be deleted from the database. A command available in a future version of KingFisher will use this flag to physically remove fish (records) from the database, thereby allowing you to eliminate unwanted records and recovering disk storage.

_ (undefined at this time; reserved for future use)

O Owned. You already own this item.

H Hidden. You do not wish to have this item appear in your searches (provided you set this flag in the Avoid mask and turn on the use of Search Masks .

M Marked. You could use this as a "bookmark" to quickly find records you wish to come back to later.

8 - 1 These are your own flags to define and use in any way that you may wish. KingFisher will never assign special meaning to these flags. You could use these flags as "bookmarks" in addition to the "M" flag, for example.

NOTE: These flags will be ghosted (i.e. unavailable for changes) if the index of the current database is located on R/O media, such as a CD-ROM or the media or file is for some other reason not writable.

NOTE: The implementation of these gadgets will change in the future to be more in line with the general appearance of King-Fisher and the Style Guidelines set forth by Commodore. They are fixed, internal images at the moment but will be made user-definable in the future.

Any alteration to these flags causes an immediate change in the database index. When the database is closed, KingFisher writes the index back to disk.

1.119   GADGET: Current Database Name

With the PREFERENCES/DISPLAY/Show Database Title command, the display of the current database title can be toggled on and off.

If you routinely work with multiple databases and frequently switch between them, this gadget will help you keep track of which database is the current. On the other hand, if your screen is of relatively small size, you may wish to turn off the display of this gadget to make the the central listvew larger.

1.120   GADGET: Search Expression History (Button ==> ListView)

Despite my obvious lack of artistic skill I hope that this gadget somewhat reminds you of a scroll. The idea is that you can choose an expression from a list of previously used expressions that has been recorded for this purpose.

Clicking on this gadget will open up a listview from which you can select a new expression by clicking on it. The window will go away immediately. Pressing the Escape key or closing the window will cancel the action.

1.121   GADGET: Search Result Window (Button ==> ListView)

If ghosted, a Search Set has not been loaded or generated.

This gadget opens and closes the Search Result window from which you can select individual matches, jumping immediately to the requisite database record to view the item's information. Closing this window will not lose the search set.

My thanks go to Fred Fish for his kind permission to use his "fish bones" Fish Logo! :-)

1.122   GADGET: Search Gadgets (Buttons)

These gadgets initiate a search either in reverse or forward direction. If you are creating a Search Set the current fish is included in the search, otherwise it is skipped. The Search Expression string gadget must contain a valid Search Expression ; if it does not, you will receive a diagnostic error message.

1.123   GADGET: Disk/Fish(record) Selector (Cycle)

Depending on the state of this gadget, the numeric gadget to its right will accept and show either the Disk number or the Record (fish) number of the currently presented record.

1.124   GADGET: Disk/Fish(record) (Integer)

Depending on the state of the cycle gadget to the left of this integer gadget, you are expected either to enter a fish (record) number in this gadget, or a disk number.

Notice that a CD-ROM consists only of one disk, which means that the disk gadget will always show a disk number of 1 and you cannot select a disk other than that.

The "Fish Disk" collection, however, consists of 1000 disks and over 4500 fish (records) so you can quickly jump to these positions in the database.

1.125   GADGET: Search Expression (String)

The search expression in this gadget is the expression that is used when you start a search. You may select another expression from the list of previously used expressions by clicking on the gadget beneath the search expression string gadget, that looks like a scroll.

Need help with constructing search expressions? Click here .

1.126   GADGET: Fish Description (ListView)

This gadget is sized to fit in the available space in the window and the textual description for each fish (record) is word-wrapped within it. You can scroll this gadget with the up and down arrow gadgets.

The layout of the displayed text is defined by the Display Format option.

1.127   5 REXXFISHER

RexxFisher is a Client application that attaches to and provides an ARexx interface to the KFServer. By invoking RexxFisher with the name of a port on the command line you can assure that any application can make use of the KFServer databases. RexxFisher defaults to a port name of REXXFISHER1. The second invocation will use REXXFISHER2, and so forth. Invoke RexxFisher with a parameter on the command line to select an exact portname. If a specific portname conflicts, RexxFisher will fail.

Results are returned in 'result' (use "say result" to display a value returned by RexxFisher) and errors are returned in the PORTNAME.LASTERROR variable, where PORTNAME does not include the increment number (1, 2, 3, etc.) added when RexxFisher is invoked without a specific portname. I.e. If RexxFisher's port happens to be 'REXXFISHER4' then the error variable is REXXFISHER.LASTERROR, without the '4'.

All commands sent to RexxFisher have an RF_ prefix to prevent confusion with standard ARexx commands. I think QUIT is a normal ARexx command, and so might be a few others here or there. You can use these commands in any mix of upper and lower case letters that you like. RexxFisher 1.10 and later will also understand the commands without the RF_ prefix.

The only commands you can execute WITHOUT a prior RF_HELLO command are the following; they do not access the database and in fact do not even require that KFServer is running!

RF_DISABLE

Disables a command.

RF_HELP

Returns a list of acceptable commands.

RF_QUIT

Tells RexxFisher to shutdown.

RF_VERSION

Returns RexxFisher's version tag string (without $VER:)

RF_LOCK

RF_UNLOCK

Prevent other connections from interfering.

The following command is used when trying to connect to the KFServer; it will start the server software if it is not already running:

RF_HELLO

Establishes a connection to the KFServer.

The following commands can be issued only if you have issued a successful RF_HELLO command:

RF_ADDFISH

Adds all fish from file to current database.

RF_BYE

Terminates a connection to the KFServer.

RF_FIND

Searches the database.

RF_GETFISH

Retrieves (and optionally formats) a database record.

RF_LIST

Obtains a list of available databases.

RF_OBTAIN

Obtains specific record information.

RF_SELECT

Selects (without retrieving actual text of) a specific record

RF_SET

Sets some of the current fish's data

RF_STATUS

Requests client status from server.

RF_USE

Selects a particular database for use.

A demonstration script, RexxDemo.kfrx , is supplied for your enjoyment. This script assumes that you have started KFServer and RexxFisher and that they are running in the background. Enter "rx rexxdemo.kfrx" to run the demo script.

Bugs: If RexxFisher tries to startup the KFServer and this fails (because the KFServer cannot be made to startup for one reason or another) then RexxFisher may crash the system with an Illegal Instruction. This problem will be taken care of in a future release!

1.128 ARexx: RF_VERSION

Usage:

RF_VERSION

Returns RexxFisher's version tag, without the $VER: portion, of course. This will ALWAYS use the standard Style Guide compliant format such as:

RexxFisher 1.5 (8.5.94)

Example:

RF_VERSION

say "Welcome to" result

1.129 ARexx: RF_HELP

Usage:

RF_HELP

Returns a list of all acceptable commands as well as some sort of command template to help you figure out what sort of parameters you might be able to get away with.

Example:

say "These commands are available to you:"

RF_HELP

say result

1.130 ARexx: RF_QUIT

Usage:

RF_QUIT

Tells RexxFisher to shutdown. In a real environment, you might want to issue a command such as "RF_DISABLE RF_QUIT" to prevent the QUIT command from being recognized. This will also suppress the command from being listed by RF_HELP.

Despite a disabled RF_QUIT, RexxFisher will respond to a CTRL_C signal such as those sent by the c:BREAK command.

Example:

RF_QUIT

1.131 ARexx: RF_DISABLE

Usage:

RF_DISABLE command

Disables a command so that RexxFisher will no longer be able to execute it. This prevents accidental shutdown of RexxFisher, for example by a "rogue script." Disabled commands will not be part of the RF_HELP listing.

Example:

RF_DISABLE RF_QUIT

1.132 ARexx: RF_HELLO

Usage:

RF_HELLO "arbitrary identification"

Needs no previous login and will establish a connection to the KFServer. If the KFServer is not running, RexxFisher will attempt to start it in the exact same way that KingFisher (the GadTools client) tries to start KFServer. The only problem is that RexxFisher cannot (yet) be told to look in a place OTHER than the default directory for the KFServer. Start RexxFisher in the same directory where KFServer is located and all will be fine.

You should give a nice and descriptive name along with the RF_HELLO, such as:

RF_HELLO "BLAZEFISHER ARexx Script"

(my apologies, Dan! :^)

If you issue RF_HELLO when already connected, then RexxFisher will issue an implicit RF_BYE command to the server to disconnect you. RexxFisher will also do this when it is made to shutdown (either with RF_QUIT or through a c:BREAK signal.)

Example:

RF_HELLO "test script"

1.133   ARexx: RF_BYE

Usage:

RF_BYE

Sign off from the KFServer. This terminates your access to the server. If you forget this, then RexxFisher keeps the connection active for the next script, which may be confusing. RexxFisher has no idea, of course, if your script has terminated or is just idling around for no particular reason.

Example:

RF_BYE

1.134   ARexx: RF_LIST

Usage:

RF_LIST

This obtains a list of all available databases from the server. The format of this list is as follows:

"Description\1database.kfdb\n

Description\1database.kfdb\n"

Which means that there are one or more lines of text each of which begins with a nice descriptive text for the database followed by a \1 character (which is an ASCII 1, ^A symbol) and followed then by the .kfdb name which you would need to give to the server through the RF_USE function to make a selection.

Example:

RF_LIST

1.135   ARexx: RF_USE

Usage:

RF_USE database

This selects a database by giving it the name of a .kfdb file. Please see RF_LIST above for more information.

Example:

RF_USE "Miniature.kfdb"

1.136   ARexx: RF_FIND

Usage:

RF_FIND "expression"

RF_FIND AGAIN

RF_FIND OPTION x

This command initiates, continues, or configures a search operation:

RF_FIND "expression"

Allows the use of the same expression syntax as KingFisher. The expression is compiled and the function begins a search at once. If you receive an error, the result string is in the format "Error X in column Y" where the error value X is one of these:

X Meaning

1 Comparison Expected ($ = != < > >= <=)

2 Operator Expected (AND, OR, XOR)

3 Invalid Comparison (ex: <! >< ... are bogus)

4 Mismatched Parentheses

5 Field Expected (must use "field op value")

6 Unsupported Feature (no hints yet :)

7 Internal Error

8 Incomplete Expression

Example:

RF_FIND "name$kingfisher|name$aquarium"

RF_FIND AGAIN

Search onward with the previously used expression. You must have an expression compiled with a previous RF_FIND "expression" command, otherwise this will not work.

Example:

RF_FIND AGAIN

RF_FIND OPTION x

Alter the behavior of the next FIND command according to the option x:

FORWARD Search forward

BACKWARD Search backward

CASEIGNORE Upper/lower case ignored

CASEEXACT Upper/lower case important

TRIMBLANKS Trim trailing blanks off search-strings

NOTRIMBLANKS Do not trim blanks

SIMPLEEXPRESSION Uses original KF1.40 expressions

COMPLEXEXPRESSION Uses new KF2.0 expressions

SHOW List current options

Example:

RF_FIND OPTION FORWARD

RF_FIND OPTION CASEIGNORE

RF_FIND OPTION TRIMBLANKS

RF_FIND OPTION COMPLEXEXPRESSION

RF_FIND OPTION SHOW

say result

NOTE: There is no way to interrupt a search in progress. Depending on user-feedback, a future version of RexxFisher may accept a break signal to interrupt a search.

1.137   ARexx: RF_GETFISH

Usage:

RF_GETFISH fishnum

RF_GETFISH fishnum width

RF_GETFISH fishnum width displayformat

Retrieve a specific fish by record number. The command has a second, optional parameter that determines if the resulting string is formatted or retrieved in raw form. A positive number for the 2nd parameter indicates the column width of the display that the text should fit. The resulting text, when formatted, will have an appearance much like that in KingFisher's ListView.

A 3rd parameter specifies a display format other than the default.

Notice that a record number of 0 retrieves the most recently retrieved record. It is best not to rely on this functionality, especially after a search operation but may be useful in some cases:

Example:

RF_GETFISH 3693

Retrieves record 3693 without special formatting.

RF_GETFISH 3693 75

Retrieves record 3693 without special formatting, but wordwraps the buffer to fit within 75 columns of non-proportional text.

RF_GETFISH 0 75 "NAME=@{name}\nAUTHOR=@{author}\nDESCRIPTION=@{description}"

Retrieves the last used record (0 has that special meaning), wordwraps the buffer to fit within 75 columns of non-proportional text, and uses a special custom format (the strange mess in ""'s) to determine what is included and how it is formatted.

1.138   ARexx: RF_ADDFISH

Usage:

RF_ADDFISH "filename"

Adds all fish from the indicated file to the current database.

Example:

RF_ADDFISH 't:new-fish.txt'

1.139   ARexx: RF_OBTAIN

Usage:

RF_OBTAIN what

Obtains a variety of information from the server, according to the parameter given:

DISK The current disk number.

FISH The current fish number, usable as the 1st parameter to the RF_GETFISH command.

FLAGS The flag bits of the current fish; the values currently defined, although not necessarily setup for each fish, are:

0x0100 Marked for retrieval

0x0200 Marked for ownership

0x0400 Marked to stay hidden in searches

0x8000 Marked to be deleted

Bits in the range 0x0001 through 0x0080 are user defined.

PVER The fish number of the PREVIOUS VERSION; the value 0 is returned if no previous version exists.

NVER The fish number of the NEXT VERSION; the value 0 is returned if no next version exists.

DBNAME The descriptive name of the database in use.

DBFILE The filename (ending with .kfdb) of the database in use. Such a filename can be passed to the RF_USE command.

DBSIZE The number of records in the current database, which is also the highest fish number you can pass to the RF_GETFISH command.

Example:

RF_OBTAIN DISK

RF_OBTAIN FISH

RF_OBTAIN FLAGS

RF_OBTAIN PVER

RF_OBTAIN NVER

RF_OBTAIN DBNAME

RF_OBTAIN DBFILE

RF_OBTAIN DBSIZE

1.140   ARexx: RF_SELECT

Usage:

RF_SELECT fishnumber

Selects a specific database record, but does not actually retrieve information. This is faster if all you need is use the RF_OBTAIN command next, perhaps to locate a record with specific flags, or to follow a list of records along the VersionLinks.

1.141   ARexx: RF_SET

Usage:

RF_SET PVER fishnumber

RF_SET NVER fishnumber

RF_SET FLAGS flags

NOTE: In prior versions this command was RF_SETVLINK.

Sets the previous or next version link for the current fish. A value of 0 for the fishnumber indicates no previous or next version. It is up to you to assure that records are properly linked in both directions, as the KFServer will not interfere with your selections. Furthermore, if a record already has a link number set, this value will be replaced with the new selection.

Example:

/* link record 17 to 76 and vice versa */

RF_SELECT 17

RF_SET NVER 76

RF_SELECT 76

RF_SET PVER 17

NOTICE: In older versions of this example, we used RF_GETFISH instead of RF_SELECT for improved speed.

NOTICE: Links need not be sequential. Thus, record 17 may have record 76 as its predecessor. Versions of KingFisher before 2.9 do not properly handle such cases, however.

1.142 ARexx: RF_STATUS

Usage:

RF_STATUS

Retrieves status information from the server. This is effectively the same as what KingFisher displays in the Status command (rightAmiga-I) except that it applies to RexxFisher.

Example:

RF_STATUS

say result

1.143 ARexx: RF_LOCK and RF_UNLOCK

Usage:

RF_LOCK

RF_UNLOCK key

Locking RexxFisher may be necessary to prevent other programs for inadvertently connecting to it and fouling up the current status while your ARexx script is passing information to another program, thereby having relinquished access to RexxFisher (i.e. ADDRESS REXXFISHER1; ... ; ADDRESS VLT ; ...)

RF_LOCK returns (in the Rexx RESULT variable) a unique value which you should store. Pass this value to the RF_UNLOCK command to unlock RexxFisher again. Without this key, RexxFisher will remain locked. ALWAYS call RF_UNLOCK before exiting your script, else the key is (probably) lost and RexxFisher must be shutdown with a CTRL-C signal and restarted!

Example:

OPTIONS RESULTS

ADDRESS REXXFISHER1

RF_VERSION

say RESULT

RF_LOCK

myKey = RESULT

ADDRESS VLT

. . .

. . .

ADDRESS REXXFISHER1

RF_UNLOCK myKey

. . .

. . .

RF_BYE

1.144   6 TROUBLE SHOOTING

This section is woefully incomplete, and I apologize. If you experience a problem of any kind with KingFisher, please write me (by postal or email) and I'll try to help. Your suggestions will make their way into this section as I learn more about what to expect and what doesn't work. Of course, bugs and conceptual problems will be fixed and smoothed out to make KingFisher easier to install and operate.

If you find that KingFisher is deficient in some way that prevents you from getting something done, or you find yourself frustrated and wish an easier way existed, or you come across a serious problem with KingFisher, please attempt to resolve the problem with me first before crying out in public. A reputation for quality is difficult to build and easy to lose. I have made every effort to assure that KingFisher Release 2 is as stable and bugfree as can be expected of a product as complex (and new!) as this. If, however, you experience a problem, allow me the chance to help!

6.1 Making a new (CD-ROM) database available to KingFisher

6.2 KingFisher is losing memory!

6.3 KingFisher 1.40 was so much easier to use!

6.4 The search window pops up far too briefly

6.5 Why don't my search expressions work?

1.145   6.1 Making a new (CD-ROM) database available to KingFisher

It is the KingFisher Database Server (KFServer) which is responsible for actually reading and writing databases on disk. KF-Server knows only about databases for which a file exists whose name ends with .kfdb . This file must exist in KFServer's default directory, i.e. the directory where the KFServer is located. KFServer will not recognize .kfdb files that are stored in any other place.

It is the purpose of these .kfdb files to describe all components of a database, i.e. what the name of the index file is, which records are stored in which data files, and less importantly the complete name of the database to be presented to the user. More detail on the contents of these .kfdb files is given here .

In most cases it may be sufficient to copy an existing .kfdb file from a CD-ROM to the directory where the KFServer program resides (most likely the same as where KingFisher is installed.) You need not restart the Server when you add a new database.

1.146   6.2 KingFisher is losing memory!

I am aware of the problem but have only been able to determine that it is the KFServer that is losing about 20K of memory, apparently through uncontrolled creation of an additional console task that never gets removed. The solution to the problem is still eluding me, as it is not my code that seems to be creating this console task... :(

If you are in the habit of starting and stopping KingFisher frequently, you may wish to change the value of "keep-running" (in KFServer.prefs) to "yes" to prevent the server from shutting down when the last client detaches. You can always shutdown the server with a CTRL_D or CTRL_C signal (see the AmigaDOS 'break' command.)

1.147   6.3 KingFisher 1.40 was so much easier to use!

Only a very few concepts have actually changed, and if you pretend to be color-blind for a moment, then the interface isn't actually all that much different either:

1. Instead of 6 search expression gadgets you now have only one. But the last 20 entries[1] are remembered and can be recalled by clicking on the gadget that is hoped to resemble a scroll. The list is sorted in order of the most recently used.

2. Search expressions now come in two flavors: the original KingFisher 1.40 expressions and the new KingFisher Release 2 expressions, which allow you to limit their search scope to specific fields. You can use the old format by making sure you have the "PREFERENCES/Searching/Simple substrings" entry checked.

3. Instead of activating a search gadget with the yellow checkmarks, you get a dedicated set of search gadgets (they have the big question marks (?) on them.)

A more detailed discussion of differences is presented here .

————————

[1] This value (MaxHistory) is adjustable in the KingFisher2.prefs file.

1.148   6.4 The search window pops up far too briefly

Most likely you have forgotten to turn off the use of Search Masks . When this option is selected, KingFisher ignores all records that do not match the selected Search Mask. If no records match (such is always the case when none of the records in the database have any flags set) then the search will be over almost as quickly as you can blink your eyes!

Use of the Search Mask will do you no good unless at least some of the records in your database have one or more flags set.

1.149   6.5 Why don't my search expressions work?

KingFisher Release 2 uses a new expression syntax, which is an extension of the one used by KingFisher 1.40 and earlier. The extension involves having to specify to which field a value is to be applied. This gives you the ability to search for "all database programs released in 1993 or later (provided a date is available)":

type$database & (date>=1993.01.01 | date="")

You can force KingFisher to drop back to the original expression syntax by selecting the Simple substrings option from the PREFERENCES/Searching menu but that will effectively search all database fields, thereby eliminating KingFisher's ability to make use of the QuickIndex for ultra fast searches.

1.150   7 THE FUTURE

Numerous enhancements are already planned for KingFisher Release 2, changes to the interface, additions to functionality, and improvements in speed. The following is a brief list of only some of the features you can expect to see in a future release, and if you wish to cast your vote for others, feel free to drop me a line:

× VersionLinks will be editable on an individual scale or by performing algorithmic scans on the database,

× A database may be reorganized, duplicated, or packed (deleted records removed),

× Support for various alerts (i.e. "search done") and iconification,

× The keyboard, including menu shortcuts, will become customizable.

And lots more...

As always, if you have requests or suggestions, please let me know! KingFisher is a user/need-driven project. Without your support and your feedback, I am simply left to implement things at my own pace, as my own priorities dictate.

1.151   8 THANKS

I would like to extend my thanks to the following people whose feedback, help, input, criticism, requests, and support have helped grind the rough edges off KingFisher and have helped make the program a more polished product than it otherwise would have been:

AMIGA

"Baby, you were born to run!"

Mike Schwager,

Jim Ventola.

And thanks, also, to the efforts of my translators, by whose efforts you can (hopefully) enjoy KingFisher in your native language:

Eduardo Delgado (Español/Spanish),

Janne J Kalliola (Suomi/Finnish),

Finn Kettner (Dansk/Danish),

Marcel Offermans (Nederlands/Dutch),

Uwe Schürkamp (Deutsch/German),

Jan Simonson (Svenska/Swedish)

And last, but not least, my thanks to you who have registered KingFisher and shown your very real support for my efforts. You know who you are. Cheers!

.__. Udo Schuermann

( ) walrus@wam.umd.edu