

FetchRefs information

COLLABORATORS

	<i>TITLE :</i> FetchRefs information		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	FetchRefs information	1
1.1	FetchRefs information	1
1.2	Table Of Contents	1
1.3	Introduction	2
1.4	Special features	2
1.5	Distribution, my address, source code, and so on	3
1.6	Send me a note	4
1.7	Updates	4
1.8	The source code	4
1.9	Author	4
1.10	Credits	5
1.11	Quotes from users	5
1.12	Problems	6
1.13	Unsupported editor	6
1.14	Limited memory	7
1.15	FetchRefs does not understand AmigaGuide AutoDocs!	7
1.16	Do not scan E modules. Use the ShowModule output!	8
1.17	RexxMast is not running	8
1.18	GenerateIndex cannot scan <insert any file name here>	8
1.19	FetchRefs cannot pick the current word from the editor	9

Chapter 1

FetchRefs information

1.1 FetchRefs information

FetchRefs 1.3

A feature packed utility that provides you with a comfortable access to your AutoDocs and include files

(General information)

Table Of Contents

Introduction

Special~features

Distribution, my address, source code, and so on

Problems

1.2 Table Of Contents

MAIN FetchRefs information

1. Introduction

2. Special~features

3. Distribution, my address, source code, and so on

3.1. Send~me~a~note

3.2. Updates

3.3. The~source~code

3.4. Author

3.5. Credits

3.6. Quotes from users

4. Problems

4.1. Unsupported~editor

4.2. Limited~memory

4.3. FetchRefs~does~not~understand~AmigaGuide~AutoDocs!

4.4. Do not scan E modules. Use the ShowModule output!

4.5. RexxMast is not running

4.6. GenerateIndex cannot scan <insert any file name here>

4.7. FetchRefs cannot pick the current word from the editor

1.3 Introduction

To get one thing straight right away: by a 'reference' I mean a part of an AutoDoc describing a particular function or a part of an include file describing a structure, a constant, or a macro. If you do not know about AutoDocs or include files, this is not a tool for you!

FetchRefs is a programmer's toy and therefore I will not give instructions on how to operate your Amiga. You are expected to know for example how to set tool types as a programmer. Okay, enough scaring; let us get onto the actual introduction.

Unfortunately it is very time consuming to look something up in the many documentation files of various libraries that most programmers have lying around. FetchRefs was made to minimize the time wasted by looking things up in these files.

You will never more wonder where EndUpdate() is documented and have to struggle with AmigaGuide or worse viewers. You simply place the cursor of your editor at the function name when you come across a source code which calls this function. Then you press a key and an ARexx script, FetchRefs, and your editor perform a little magic -- voila! A piece of the 'layers.library' AutoDoc has been cut out and placed in a new view in your editor. Big surprise! The piece that was cut out is exactly the part describing EndUpdate().

As FetchRefs is operated through ARexx, it is very flexible. For example, if you do not want to keep FetchRefs' memory consuming index table resident all the time you can simply extend the ARexx script to load the index file on demand and flush it right after.

1.4 Special features

Though at least half a dozen other programs help you to get a hand on the same kind of references as FetchRefs does, FetchRefs has some features which make it the absolute winner (beware, I may be slightly biased :-).

- FetchRefs fits into every environment as long as the editor has a decent ARexx port - which an editor of today HAS. Therefore you will not have to learn to operate a completely new program - you just need to know what key/menu you attach FetchRefs to and FetchRefs suddenly becomes an integrated part of your editor.
 - FetchRefs understands AutoDoc files as well as include files for both C, Assembler, and E.
 - If you are uncertain as to what you are actually searching for, FetchRefs supports wild cards as well.
 - FetchRefs knows of the popular suffixes 'Tags', 'TagList' and 'A' for different interfaces to the same function. This ensures for example that you can get a reference to 'System' even though the function is documented as 'SystemTagList' in the AutoDoc.
-

- FetchRefs will pop up a window with a list of all matching references when it finds out that a search does not result in exactly one reference. You are then able to select the particular reference you were searching for. This is extremely handy for things like OpenDevice() which is documented in the AutoDocs for every device or when you search using wild cards (e.g. 'Open#?' would return quite a list).
- From a reference name you can get a list of all references in the same file and thereby possibly find additional information.
- FetchRefs can consider each file it indexes as a reference itself. The effect is that you can follow a thread even if the 'SEE ALSO' paragraph reads something like <dos/dostags.h>. Fetching this reference will simply load the entire file into the new view.
- FetchRefs's index files are generated by a powerful index file generator which sports both command line options (handy for script usage) and a font and size sensitive GUI. Any type of reference may be turned on or off during index file generation to customize the index file completely.
- FetchRefs itself naturally also supports execution from both Workbench and Shell. Thus you can put it in either s:user-startup or WBStartup, as you please.
- FetchRefs comes with ready-to-use ARexx scripts for most of the current editors: CygnusEd, Edge, EMACS, FrexxEd, GoldED, and TurboText. Adapting one of them to another editor should be relatively easy if you know some ARexx. As a last resort, an additional script lets you look up references from the Shell.
- FetchRefs is free. Complete C source is available for free, too.
- And yes; FetchRefs *is* better than the built in 'Find reference' thing of GoldED. By far, thank you very much.

If you do not agree that FetchRefs is the best utility to get access to references, please tell me why - and I shall try to make it better.

1.5 Distribution, my address, source code, and so on

In short: FetchRefs is free - please spread it.

Now a bit more specific:

Send~me~a~note
Updates
The~source~code
Author
Credits
Quotes from users

1.6 Send me a note

Donations of any kind are naturally gladly accepted but are not required for using FetchRefs. I will not think bad of you if you tell me that you use FetchRefs without paying me something. It will actually make me much more happy than *not* knowing that you use it.

In fact, I would like everyone using FetchRefs to send me a post card or even just a NetMail/e-mail (address: see author) telling me what they think of it. It is nice to know that someone can use what I have made.

Too few people seem to appreciate free software...

1.7 Updates

The most recently released version of FetchRefs should always be available for free on Aminet (dev/misc).

1.8 The source code

FetchRefs is written completely in C and I compiled it using the SAS/C compiler version 6.58.

GenerateIndex was written for the DICE compiler, it has not been modified.

The distribution should have included the sources in an archive.

You can use it for learning purposes or whatever but if you make any improvements to FetchRefs PLEASE SEND IT TO ME. I do not want other people to distribute new versions as that would probably screw up the version numbering. If you think that I have stopped development and you want to take over, PLEASE ASK ME FIRST. I do not put out new versions every day so perhaps it is merely sleeping and not dead just yet.

1.9 Author

Do you not just hate it when you know nothing about the author of the program you are using? No? NO? Shame on you :-).

The original author of the program was Anders Melchiorson (tanks to him!).

I have adapted the source code of FetchRefs to SAS/C because I use this one for years now and I had a problem with the FetchRefs program, that caused a crash on my system. I didn't own the DICE C compiler, so I tried to adapt the source for recompiling it. It's done now (with some difficulties), I have also added localisation (not too difficult).

If you would want to contact me (see send-me-a-note), please do not hesitate to write. I love getting letters. Post cards are nice, too.

If there is some demands, I'll localise the GenerateIndex program too, but I need to adapt the source to SAS/C and it's a big work...

Snail-Mail: Roland Florac
6 Rue des Chardonnerets
Chez Corbin
17610 Chaniers
France

E-Mail: Roland.Florac@wanadoo.fr

1.10 Credits

sortlist.lib, Copyright by Preben Nielsen
Triton, Copyright by Stefan Zeiger
ReqTools, Copyright by Nico François

Moral support by Kasper B. Graversen, Leon Woestenberg, Simon Gramstrup, Alex Holst, and everyone else who has sent me comments.

And naturally thanks to everyone using FetchRefs for liking it.

1.11 Quotes from users

I have taken the time to collect a few comments from some of the e-mails I have received about FetchRefs. Just to show off, here they are. I have kept them as I received them which means that most of them are in Danish.

"Some fancy stuff you released there! FetchRefs is the perfect solution for spoiled programmers that use the best tools available :-)"

"Jeg har lige siddet og kigget på din Fetchrefs1.1...FEDT program :-)"

"Jeg har nettop installeret FetchRefs programmet ditt og jeg synes det var meget bra, men det er rom for noen smaa forbedringer:"

"Gode features du har kommet i!"

"Det er jo temmeligt genialt! :)"

"Ellers et virkelig smart program! Keep coding!"

"Jeg vil bare lige udtrykke min ekstremme fornøjelse over dit FetchRefs."

"Det er vaesentligt hurtigere end hvad jeg har set tidligere:)"

"I can't think of any improvments, so this is just to let you know that your program has greatly increased my development time."

"JEG MENER at FetchRefs bliver mere genial for hver dag jeg bruger den"

"Joda, faktisk MEGET behageligt at have installeret."

"Jeg havde set det kører på en anden maskine, så derfor MÅTTE jeg bare have det på min."

"Det er FEEDT."

"Jeg bruger plaintext autodocs og FetchRefs til at læse dem med - meget bedre kan det ikke blive :)"

1.12 Problems

Below is a list of some problems I figured you could encounter. Most of them are based on questions I have received. Also, most of the information presented here is repeated somewhere else in the documentation. As a funny consequence, most questions are answered elsewhere in the manual - but this section keeps growing anyway. I do not mind answering questions and looking up things in the manual for you but it might be faster if you did it yourself!

I would very much like you to send-me-a-note if you come across a problem that you cannot solve by reading the entire documentation. Only then can I improve that part and save others from the troubles you have.

```
Unsupported~editor
Limited~memory
FetchRefs~does~not~understand~AmigaGuide~AutoDocs!
Do not scan E modules. Use the ShowModule output!
RexxMast is not running
GenerateIndex cannot scan <insert any file name here>
FetchRefs cannot pick the current word from the editor
```

1.13 Unsupported editor

The greatest problem is most likely that you use an editor which I have not provided a script for. I cannot really help you with that one, because if I had had access to that editor I would have made an ARexx script for it. If it is a freely distributeable editor you can send me a copy and I will provide the script (remember that I cannot make the script if I do not know the ARexx commands of the editor so I also need a copy of the documentation).

If your editor does not have an ARexx port or the port is very limited, you are probably out of luck and should look around for another editor. The functions that the port should be capable of doing are:

- Get the current word (the word the cursor is at)
- Open a new window/view
- Load a file into the newly created window

A function to jump to a specified line is handy for include file references and it will give a more integrated system if there is a way to

tell if anything went wrong. However, this is not really required. If the editor does not support the above mentioned functions, I honestly cannot see how it should work. It is, however, some of the most basic functions I can think of, so if the editor has an ARexx port, it probably supports them in some way.

I have been able to make scripts for all the editors I have tried. By reading the various scripts you may be able to learn a few tricks though you should not get any problems.

If you should succeed in creating a new script I would be happy to get a copy of it so that I can include it in a future release. In that case it would be nice if you could keep it simple (like the ones I have provided) and not specific to your system. I cannot modify the scripts much as I do not have your editor to test it on!

1.14 Limited memory

The index file that FetchRefs uses is quite big to keep in RAM all the time - considering that it is not used very often.

A solution would be to make a script that invokes the FR_CLEAR and FR_ADD ARexx commands (see 'FetchRefs_FR.guide'). This way you can flush the index whenever you do not use it. With a relatively fast harddisk this will make each loop-up take just a little longer and you only use the RAM for the index file when you actually need it.

I have not implemented this into the ARexx scripts provided as I wanted to keep them quite simple - by getting sophisticated, more people will experience problems with their particular setup. Besides, most people have a few hundred kilobytes to burn these days.

1.15 FetchRefs does not understand AmigaGuide AutoDocs!

There seems to be a few people that have never seen classic AutoDocs but only those hot AmigaGuide documents.

The original AutoDocs are plain text files with a quite simple format. It is, however, possible to convert them into AmigaGuides with quite a bit of links (all function, structure, and constant names become a link).

This does not work very well together with FetchRefs as GenerateIndex does not understand how to scan AmigaGuides for keywords. Therefore you need to have the original AutoDocs installed if you want to use FetchRefs - however, with FetchRefs running, regular AutoDocs are even more useful than those fancy guides (and take up a little less space).

If you have some documentation which is only available in AmigaGuide format, you may try to convert it with the supplied Shell tool Guide2AutoDoc. It is a small program that I threw together one night to convert the guide of stdio C functions that SAS supplies with their compiler. It has not been tested heavily but it is definitely worth a try.

Usage is:

```
Shell> Guide2AutoDoc <guide >autodoc
```

That is, you must use redirection as Guide2AutoDoc reads from stdin and writes to stdout.

1.16 Do not scan E modules. Use the ShowModule output!

The E modules are binary files. That is bad when it comes to studying them. Therefore you cannot scan these files with GenerateIndex. Instead you must use ShowModule on all your modules and save the output somewhere. Then use GenerateIndex to scan *these* files.

1.17 RexxMast is not running

You cannot use FetchRefs if your ARexx system is not up and running. A few people like to save a few bytes of memory by not running RexxMast as part of their startup. That is a very bad idea if you intend to use FetchRefs. All FetchRefs does is based on the reception of ARexx commands so you need to have ARexx active, period.

1.18 GenerateIndex cannot scan <insert any file name here>

If you scan a file but GenerateIndex does not recognize any or not all references in that file it is most likely *not* a bug in GenerateIndex! (Yeah, right! :-).

First of all, check your scanning options. Be sure that you have activated both the file type and all the kind of references you want (for AutoDocs you only activate the file type itself as there is only one kind of references). If you are in doubt, read the section on scanning options in 'FetchRefs_GI.guide'.

If you have set all options correct the problem is probably that GenerateIndex either cannot figure the file type out or that it cannot understand the format.

The file type is selected by looking through the file for certain keywords. The complete list is in 'FetchRefs_GI.guide', section 3.1 ('Deciding what files to scan'). You can check if the file is recognized by looking at whether it gets added to the list if the KEEPEMPTY option is turned on. Unrecognized files are skipped totally and are thus not kept even if KEEPEMPTY is turned on to keep files without references!

If the file lacks all the recognized keywords you can add any one of them in a comment inside the file (while most scanners ignore comments, the file type detection does not). For AutoDocs you can add the text 'TABLE OF CONTENTS' at the top. Naturally, this is just a workaround. You really should contact the author of the file or perhaps me to get the problem

solved permanently.

If GenerateIndex recognizes the file but does not find all references in it, the problem is getting bigger.

Make sure that the format is correct if the file is an AutoDoc. Section headlines must contain just the function name twice on the same line and sections must end with a form feed (ASCII 12). If the AutoDoc does not follow this convention, make the author corrects it!

If it is an include type file, the format is probably too complex for GenerateIndex at present. Send me the file and I will see what I can do to improve the scanner. However, except for the rarely used complicated C typedef statements (which I cannot do much to scan better) this should happen very seldom.

1.19 FetchRefs cannot pick the current word from the editor

Some of the scripts just pick the word the cursor is at when you call them. Other scripts take the rest of the word, starting from the cursor position. Make sure the cursor is at the start of the word if you use a script of the second kind!
