

## **GetReflQ documentation**

**COLLABORATORS**

	<i>TITLE :</i> GetRefIQ documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>GetRefIQ documentation</b>	<b>1</b>
1.1	GetRefIQ documentation	1
1.2	Table of contents	1
1.3	Introduction	2
1.4	Distribution	2
1.5	Getting started	3
1.6	Starting from Shell	3
1.7	Starting from Workbench	4
1.8	Everyday usage	4
1.9	ARexx port and commands	5
1.10	GUI and operation	5
1.11	ARexx return codes	6
1.12	Contacting author	6
1.13	History	7
1.14	Thanks and Final words	7

---

## Chapter 1

# GetRefIQ documentation

### 1.1 GetRefIQ documentation

-----  
GetRefIQ 1.0 Release I  
-----

-----  
A commodity that allows Amiga C and assembler programmers quickly access  
needed information about system calls and structures.  
Even more. It allows something similar to filename completion on Unix or  
implemented in KingCON.  
-----

Introduction  
What is it.

Distribution and Copyrights  
How to spread it.

Getting started  
How to start it ...

Everyday usage  
And how to use it.

Contacting author  
How to contact me, author.

History  
If you like to read (me do :).

Thanks and Final words  
Why not to look here ?

### 1.2 Table of contents

---

Introduction

Distribution and Copyrights

Getting started

Workbench

Shell

Everyday usage

ARexx port and commands

GUI and operation

Contacting author

History

Thanks and Final words

## 1.3 Introduction

GetRefIQ is designed to help programmers in their everyday work. It is a commodity which on a command finds specified structure definition(s), macro(s), etc. You can use it with any editor that supports ARexx (which today doesn't ?) or even with some work, with others. Of course it is not limited to editors, it has general ARexx port so it easy to use as you like it. GetRefIQ uses index file produced by GenIndexIQ . To run it requires only rexxsyslib.library and about 15 KB of memory. ↔

<- Main

## 1.4 Distribution

### DISCLAIMER

WITH THIS DOCUMENT I MAKE NO WARRANTIES OR REPRESENTATIONS EITHER EXPRESSED OR IMPLIED WITH RESPECT TO THE PRODUCT DESCRIBED HEREIN. THE INFORMATION PRESENTED HEREIN IS BEING SUPPLIED ON AN 'AS IS' BASIS AND IS EXPRESSLY SUBJECT TO CHANGE WITHOUT NOTICE. THE ENTIRE RISK AS TO THE USE OF THIS INFORMATION IS ASSUMED BY THE USER. IN NO EVENT WILL I BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY CLAIM ARISING OUT OF THE INFORMATION PRESENTED HEREIN, EVEN IF I HAVE BEEN ADVISED OF THE POSSIBILITIES OF SUCH DAMAGES.

COPYRIGHT

---

GetRefIQ and its documentation is Copyright ©1996-1997 Radoslaw Chyra, GetRefIQ.ced ARexx script is ©1996 Cromax/Alchemy and is included in this package with his permission.

#### DISTRIBUTION

GetRefIQ is GIFTWARE. It means if you find it useful and use it I'll appreciate a gift you think will match my effort creating and future development of this program. It also means that you can freely spread this program provided that all the associated files are distributed with the program itself. Neither the program nor their associated files must be modified in any way. The complete distribution package can be compressed with one of easy available archiving programs (like Lha, Tar, etc.). No charge must be made for use or distribution of this package. You can upload it any BBS, include it in a magazine's coverdisk, in a Public Domain library or in recompilation CDs of PD/Freeware/Shareware (like Aminet) although I'll appreciate a copy of such a media (CD, Magazine, etc.). For contact addresses look

here

.

<- Main

## 1.5 Getting started

GetRefIQ is a commodity to be started from

Workbench

,

but it is possible to start it from

Shell

.

If started twice just both copies quit!

As a commodity, GetRefIQ supports Remove and Disable/Enable. When disabled, everything that comes to ARexx port will be returned with no error code.

<- Main

## 1.6 Starting from Shell

Shell template is:

```
"FROM/M/A,PUBSCREEN=P,WINX/N,WINY/N,EXTENDED=E/S,KEEPINDEX=K/S,HANDLE_LVO=H/S"
```

```
FROM file ... : file ... is list of index files to scan references for,
                If file name consists space characters then enclose it
                in quotes. This parameter must be specified, if not
                GetRefIQ will exit.
```

```
PUBSCREEN=P name : specify public screen name to open window on, if there
```

is no such screen, GetRefIQ will open it's window on default public screen.

```
WINX x      :
WINY y      : specify window positions (default 0, 0).
EXTENDED=E : turn on extended key handling (read below) (default off).
KEEPINDEX=K : turn on index caching in memory (default off).
HANDLE_LVO=H : turn on handling of _LVO prefix for reference names
              (implemented for Cromax' request :) (default is off).
              This option works in such a way (if on): First when
              'GRIQ_FIND key' command is issued, it checks if there is
              _LVO prefix in a key. If it is, _LVO is cut, and search
              begins. If something was found and selected, GetRefIQ
              checks if key had _LVO prefix and if it had one,
              selected reference is prefixed with it.
              Simple, isn't it ?
```

<- Back

## 1.7 Starting from Workbench

Recognized Workbench Tool Types:

```
FROM=file_list : file_list is list of index files to scan references for,
                 file names are joined with plus ('+') sign. This Tool Type
                 must be specified, if not GetRefIQ will exit.
PUBSCREEN=name  : specify public screen name to open window on, if there is
                 no such screen, GetRefIQ will open it's window on default
                 public screen.
WINX=x         :
WINY=y         : specify window positions (default 0, 0).
EXTENDED      : turn on extended key handling (read below) (default off).
KEEPINDEX     : turn on index caching in memory (default off).
HANDLE_LVO    : turn on handling of _LVO prefix for reference names
                 (implemented for Cromax' request :) (default is off). This
                 option works in such a way (if on): First when GRIQ_FIND
                 key command is issued, it checks if there is _LVO prefix
                 in a key. If it is, _LVO is cut, and search begins. If
                 something was found and selected, GetRefIQ checks if key
                 had _LVO prefix and if it had one, selected reference is
                 prefixed with it. Simple, isn't it ?
```

<- Back

## 1.8 Everyday usage

ARexx port and commands

GUI and operation

<- Main

## 1.9 ARexx port and commands

ARexx port name is "GetRefIQ". You can send following ↔ commands and get following return codes:

```
GRIQ_QUIT           : tell GetRefIQ to quit
GRIQ_HANDLE_LVO YES|NO : turn on/off HANDLE_LVO option (but only
                        temporarily, until nearest search command is
                        completed)
GRIQ_WINX x         :
GRIQ_WINY y         : specify window positions (also temporarily, as
                        above)
GRIQ_PUBSCREEN name : specify public screen to open window on
                        (temporary)
GRIQ_FLUSH          : flush index memory (if there were any files kept)
GRIQ_FIND searchkey : find reference, all references beginning with
                        'searchkey' will be found (searching IS case
                        sensitive!)
```

If there is only one reference found GetRefIQ assumes user wanted both info and symbol and returns RET\_BOTH (about return codes read here). If there aren't any GetRefIQ will just flash the screen indicating that nothing was found and returns RET\_NONE. If there are more than one reference GetRefIQ will open a window allowing user select wanted reference and action to perform on it.

See GetRefIQ.ced for examples how to use GetRefIQ ARexx port.

<- Back

## 1.10 GUI and operation

GUI is completely font sensitive, I tried to do my best to make it ↔ work also under v2.04, but if something is wrong contact me please! There is one listview for selecting appropriate reference. You can select it with mouse or using cursor keys (up and down). Shifted cursors jump 15 references up and down. Below there is info gadget which displays which file selected reference comes from (it is handy when you have few the same references that come from different files, e.g. there are few OpenDevice references in different autodocs files!). And at bottom of the window there are three gadgets to select action:

```
Info : returns RET_INFO and saves information to file returned in ARexx
      variable RESULT2
Symbol: returns RET_SYMBOL and stores selected reference in ARexx variable
```



**RESULT**

Both : returns RET\_BOTH and acts as both RET\_INFO and RET\_SYMBOL

Supported keys:

I (and RETURN if option EXTENDED is on) : Info  
S (and SHIFT-RETURN if EXTENDED on) : Symbol  
B (and ALT-RETURN if EXTENDED on) : Both  
ESC : close window and return RET\_EXIT

About return code values read  
here

.

<- Back

## 1.11 ARexx return codes

0 = RET\_NONE : nothing happened (or port is disabled)  
1 = RET\_SYMBOL: user selected Symbol  
2 = RET\_INFO : user selected Info  
3 = RET\_BOTH : user selected Both  
4 = RET\_EXIT : user took no action  
10 = RET\_ERROR : something went wrong  
(couldn't open window, no memory, etc.)

## 1.12 Contacting author

All stuff contained here in this package was brought to you by

Radoslaw Chyra (Chyras/NegativeIQ)

I'm a 23 year old student at Wroclaw University. I study Computer Science and now I'm trying to finish my third and fourth year (together :) At home I use Amiga1200 self made Tower with 540 MB harddisk. I also own Pentium PC box, but the only thing I use there is Linux and appropriate programming, TeXing, etc. stuff. MS-DOS, Win\*\*\*s and similar suck. Definitely suck. Although it's my opinion :)

I started with my first computer Amiga 500 back in 1987. I coded some demos in assembler under the name of PEPSI Drinkers Inc. Then I moved to coding OS compliant software (still in assembler) and together with Grzegorz Calkowski (UncleMat) we formed Beet Development Group. Now I programm under the name of NegativeIQ which is two-person programming team. Second member is Szeryf, who is PC owner.

For bug reports, enhancements, flames etc contact me at:

e-mail:

chyras@tempus.ii.uni.wroc.pl

(Should be valid for at least two years)

irc:

---

```
chyras @ #amigapl (not so often!)
www:
http://tempus.ii.uni.wroc.pl/~chyras/
```

I can also be reached by snail-mail, you can send gifts here:

```
s-mail:
Radoslaw Chyra
Kosnego 36/2
45-056 Opole
Poland
```

<- Main

## 1.13 History

```
-----
Release I

V1.0      - first public release
```

<- Main

## 1.14 Thanks and Final words

Here I would thank Anders Melchiorsen for his FetchRefs, which didn't fit my requirements but forced me to write my own program to fetch references. ↔

Also thanks to Cromax/Alchemy who submitted with some useful suggestions and comments in development stage. He also supplied GetRefIQ.ced ARex script.

The last one but not least whom I would thank is Robert George Wal (Reptile), who pointed my attention at BGUI library and convinced me to use it, instead of MUI (although programming in MUI was easier I think, for such a small program like GenIndexIQ MUI is too big and too slow).

Yeah! Almost forgot to thank Jan van der Baard and Ian Einman for BGUI, and Nico Francois and Magnus Homgren for ReqTools!

As I'm still waiting to get my hands on Blizzard 1240T/ERC accelerator, I'm stuck with 2MB Amiga. It's really hard to program on it, especially when I use SAS/C 6.56 compiler, but even with that troubles, I must say that development on Amiga is much easier for me than on PC or Unix (I use them both, so I can compare). The worst is PC, no multitasking, that's it! Although I would like to have such environment and system capabilities like on Unix, hope PPC Amiga will be out soon!

As a next project I'll probably pick up my old unfinished project of Module

---

Manager (as I collect modules) and concurrently I'll try to develop GUI layout engine under XWindows and Windows'95 (together with Szeryf). It's strange, but I haven't seen any layout engines on both platforms which would be similar to MUI, BGUI, Triton, EAGUI etc. engines on Amiga ... Strange!

GetRefIQ was tested on:

Amiga 500 with 0.5MB chip and 2 MB Slow equipped with KS2.04

Amiga 1200 with 2MB chip and KS3.0

Amiga 1200/Blizzard 030/50 2MB chip and 8MB fast with KS3.1

<- Main