

# **xref-system**

Stefan Ruppert and Julian Scheid

|                      |
|----------------------|
| <b>COLLABORATORS</b> |
|----------------------|

|               |                                     |                |                  |
|---------------|-------------------------------------|----------------|------------------|
|               | <i>TITLE :</i><br>xref-system       |                |                  |
| <i>ACTION</i> | <i>NAME</i>                         | <i>DATE</i>    | <i>SIGNATURE</i> |
| WRITTEN BY    | Stefan Ruppert and<br>Julian Scheid | August 9, 2024 |                  |

|                         |
|-------------------------|
| <b>REVISION HISTORY</b> |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
|        |      |             |      |

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>xref-system</b>                            | <b>1</b> |
| 1.1      | xref-system.guide . . . . .                   | 1        |
| 1.2      | XRef-System.guide . . . . .                   | 1        |
| 1.3      | XRef/Copyright and Registration . . . . .     | 1        |
| 1.4      | XRef/XRef-System and Feature List . . . . .   | 2        |
| 1.5      | XRef/Configuration-File . . . . .             | 4        |
| 1.6      | XRef/Installation . . . . .                   | 4        |
| 1.7      | XRef/XRef.Library . . . . .                   | 5        |
| 1.8      | XRef/Tools . . . . .                          | 5        |
| 1.9      | XRef/How to use it for own projects . . . . . | 6        |
| 1.10     | XRef/To Do . . . . .                          | 7        |
| 1.11     | XRef/Thanks . . . . .                         | 7        |
| 1.12     | XRef/Address . . . . .                        | 8        |
| 1.13     | XRef/Compiler and Software Tools . . . . .    | 8        |
| 1.14     | XRef/AmigaGuide Setup and Problems . . . . .  | 9        |

# Chapter 1

## xref-system

### 1.1 xref-system.guide

### 1.2 XRef-System.guide

```
Cross-Reference-System (XRef-System)
~~~~~
```

Welcome to the XRef-System Version 1.1.

The XRef-System is a cross-reference system using the AmigaGuide Format to display documents. It provides many more features than normal AmigaGuide XRefs. It is much faster, more flexible and allows you to easily add own projects.

Topics :

- Copyright and Registration
- XRef-System and Feature List
- Installation
- XRef.Library
- Tools
- How to use it for own projects
- To Do
- Thanks
- Address
- Compiler and Software Tools
- AmigaGuide Setup and Problems

```
xref.library :
  Documentation
  History
```

### 1.3 XRef/Copyright and Registration

The XRef-System is Shareware, but has no restrictions.

Permission is hereby granted to distribute the program's archive

---

containing the executable's and documentation for non-commercial purposes as long as the archive and its contents are not modified in any way. You can modify the given tools for your own use, but don't distribute such changed files !!! You can send me these changes, so that I can coordinate the distribution of the XRef-System !!!

It is forbidden to include this archive in any kind of software collection except the Fish Amiga-Library, AmiNet-CD/FileServer or BBS fileareas !

No guarantee of any kind is given that the programs described in this document are 100% reliable. You are using this material at your own risk. The author takes no responsibility for any damage which is caused by using these programs.

Registration :

If you use the XRef-System, you have to send me a fee of DM 20 or US\$ 15. If you do so, send the fee to the following address:

Stefan Ruppert  
Windthorststraße 5  
65439 Flörsheim am Main  
GERMANY

EMail: ruppert@vs3.informatik.fh-wiesbaden.de

Copyright (C) 1994,1995 by Stefan Ruppert

## 1.4 XRef/XRef-System and Feature List

The main reason for writing the XRef-System was that the current implementation of the amigaguide.library is very slow. Also, I wanted to integrate some concepts ←  
from

HotHelp (such as searching using patterns), which has the disadvantage of not supporting the AmigaGuide document standard.

The result satisfies my demands, and I think it is a very useful utility ←  
especially  
for programmers.

All programs and the library are compiled with SAS-C V6.50! (see Compiler and ←  
Software  
Tools)

Note: All the links you see within this database are generated via the XRefConvert tool (except of three links, which I had to create manually).

These are the features the XRef-System offers you:

- o generates xref entries for almost any C expression (#defines, #includes, structures, functions, typedefs etc.), for Commodore's autodoc file format, amigaguide files, and unix man pages.
- o XRefConvert - A tool to convert all common files to AmigaGuide databases (autodoc files, header files, doc files and other AmigaGuide files).  
Now implements an option to generate AmigaGuide files for use with

- the amigaguide.datatype V40 and higher, using the smartwrap algorithm.
- o optional search with AmigaDOS patterns. In case of more than one entry is matching, XRef generates a temporary AmigaGuide database (AGuideXRefV37).
  - o a new tool replacing AmigaGuide, which has a direct interface to the xref.library and some additional features (AGuideXRefV39)
  - o xref files can be held resident in memory (xref.library). It is also possible to lock individual databases, so that they can't be freed when memory is flushed.
  - o xref.library can be configured via xref.prefs, with AUTOOPEN mechanism during xref.library startup and some global settings.
  - o xref database files can have different priorities assigned, so that your most-used files are searched first (see Enqueue()).
  - o memory allocation is fully implemented using V39 memory pool functions (including the V37 library version)!
  - o Dynamic AmigaGuide nodes for tables of:
    - xref files
    - all file references in an xref file
    - all generic entries
    - all function entries
    - all device command entries
    - all include file entries
    - all macro entries
    - all define entries
    - all structure entries
    - all typedef entries
  - o categorization (with patterns) of XRef files gives you more flexibility .
  - o binary search algorithm for normal strcmp()/strncmp()  
parse modes (XRefConvert uses only these modes, therefore being very fast! Better than AD2AG utility from Commodore.)
  - o GoldED API-Client XRefAPI for completing phrases.  
For example, "ow" would be completed to "OpenWindow(".
  - o Workbench support for all tools (some using a small status window) using FinalReadArgs() function (also included).
  - o configuration directory can be specified with an environment variable
  - o dynamically loadable xreffiles (see Configuration-File).
  - o rexxxref.library ARexx function FindXRef() to use the XRef-System from ARexx
-

If you missed a feature, please contact me (see Address).

## 1.5 XRef/Configuration-File

The configuration file "xref.prefs"

This file should be placed in your "Sys:Config/XRef" directory.  
If you don't want to put it there, you can specify a directory with the XREFCONFIGDIR environment variable.

The file must be an ASCII text and is parsed line-oriented. Each line is interpreted to match one of the following templates (using ReadArgs()) :

- AUTOOPEN/S/A, FILE/K/A, XREFPRI/N/K, LOCK/S, INDEX/S
- AUTOLOAD/S/A, FILE/K/A, XREFPRI/N/K, LOCK/S, INDEX/S
- MAINPAGE/S/A, LINELENGTH/N/K/A, COLUMNS/N/K/A
- CLEANUP/S/A, FORCE/S
- XREFDIR/S/A, DIR/A/K

### AUTOOPEN

This type of entry may exist several times in a prefs file.  
It specifies an XRef file to be opened when parsing this file. The specified arguments are passed to the XR\_LoadXRef() function. (see this document for further informations).

### AUTOLOAD

This entry specifies XRef files to be automatically loaded if an entry wasn't found in the XRef files in memory.

### MAINPAGE

This entry specifies the line length and the column number which will be used for all dynamic node layouts . Thus you can set these values appropriate to your screen/window resolution.

### CLEANUP

If this line appears in a prefs file, all XRef files are flushed (removed) from memory. If you specify the FORCE switch additionally, locked XRef files are also flushed.

### XREFDIR

If you specify this line, the given path is used as the default path for load XRef files.

Note: ReadArgs() doesn't support the "/S/A" combination, this is checked separately.

## 1.6 XRef/Installation

Installation  
=====

Simply run the Installer script in the Install directory. Make sure, that

you have the Installer program in the Sys:Utilities directory, or on another place, where the workbench can find it.

Here is a short description, what and where this script installs on your system :

It copies the xref.library to your LIBS: directory according to your kickstart version a V37 or a V39 library.

The configuration files for the library and the tools are copied to Sys:config/xref or to a specified directory. I decided this default location instead of ENV: because these files aren't changed often, so I think it isn't worth wasting RAM! The following files are copied:

- keywords - used by XRefConvert
- keywords.v40 - used by XRefConvert
- exceptions - used by XRefConvert
- xref.prefs - used by xref.library during library startup

The script sets the xref.prefs file correctly for you. You can add any XRef file that should be loaded directly after the library init. Simply add a line like

```
AUTOOPEN myxreffile.xref LOCK
```

or see section "The configuration file" for a more detailed descriptions of xref.prefs.

If you have installed the GoldEd Editor, the Installer script copies the xrefaguide.ged ARexx macro to the GoldEd:ARexx directory.

All other files are copied to the place you specify via the Installer.

## 1.7 XRef/XRef.Library

The xref.library is the fundament of the XRef-System. It provides functions to load, expunge and parse XRef database files. The library can be configured using a file named xref.prefs. See section "The configuration file" or the LoadXRefPrefs() doc for further details.

In low memory situations, all libraries that aren't used currently are flushed. For this case, the xref.library has a locking mechanism for xref files. If you loaded an xref file with the LOCK switch turned on, this xref file isn't flushed, and the library neither. Thus you can decide if the xref.library should removed during a memory flush or not.

SEE ALSO

- Library-Documentation
- Library-History

## 1.8 XRef/Tools

---

|                   |  |
|-------------------|--|
| AGuideXRefV37     | parses the xref files searching for a given pattern/string and opens an AmigaGuide window using the amigaguide.library.<br>If more than one entry matched, a temporary AmigaGuide database is created to allow you to select the desired entry |
| AGuideXRefV39     | same as AGuideXRefV37, except that the window is opened using the amigaguide.datatype, providing you some new features according to the xref.library   |
| ExpungeXRef       | removes xref files from memory   |
| LoadXRef          | loads xref files into the memory   |
| MakeAGuideIndex   | generates an index file for given AmigaGuide files   |
| MakeXRef<br>files | creates xref files from C includes, AutoDoc files, AmigaGuide and unix manualpages files   |
| ParseXRef         | parses the xref files searching a given pattern/string and outputs to stdout   |
| StatXRef          | outputs the current settings of the xref.library   |
| XRefAttrs         | changes attributes of one or several xref file(s)  |
| XRefConvert       | converts AutoDoc or C Include files to AmigaGuide files  |
| XRefAPI           | This is not a tool but a GoldED API-Client for completing phrases like "of" to "OpenFont("   |

## 1.9 XRef/How to use it for own projects

If you want to generate xref files for your own projects or to build all AutoDocs, you have to follow these steps:

1. Generate an xref file from your C-Header or AutoDoc files.  
The category option string is put into the generated file, therefore do not specify the category explicitly ! (Note that if you set a global XRefPath, you can only use the filenames as shown here!)

Example: Generate two xref file for the directories include:myincludes and docs:mydocs :

```
MakeXRef FROM include:myincludes TO myinclude.xref \
CATEGORY Include VERBOSE
```

```
MakeXRef FROM docs:mydocs TO mydocs.xref CATEGORY AutoDoc \
VERBOSE
```

2. Load the generated files into memory:

```
LoadXRef myinclude.xref mydocs.xref PRI 5
```

3. Convert the include's and autodoc's to amigaguide databases:

```
XRefConvert FROM include:myincludes TO AMIGAGUIDE:Includes VERBOSE
```

```
XRefConvert FROM docs:mydocs TO AMIGAGUIDE:AutoDocs VERBOSE
```

4. If everything went alright, the result is one or more AmigaGuide files in the AMIGAGUIDE:Includes and/or AMIGAGUIDE:AutoDocs directories.

[Click here](#) to perform these steps for the System Includes & Autodocs with the Installer, if you have the installer in your C: directory:

## 1.10 XRef/To Do

Work I'd like to do :

- a new gadgetclass for V39 and above, which integrates the amigaguide datatype and a direct interface to the xref.library (this is currently the AGuideXRefV39 program)
- a complete ARexx interface to the xref.library, at the moment only the FindXRef(), LoadXRef() and ExpungeXRef() functions are available from ARexx using the rexxxref.library
- perhaps a Preference editor
- localization of all tools (only the AGuideXRefV39 uses a catalog) (?)
- Maybe in future a new amigaguide class, which supports graphics and/or other datatype formats. (But for this, I need a full documentation of the amigaguide.datatype. With a system monitor, I found out that the amigaguide datatype has 13 LVO function entry points. Where are the docs? Thanks to Commodore for these not accessible docs... Arghhh!!!)
- More flexible MakeXRef and XRefConvert, that allows to add new scanners for different languages.
- perhaps support of pipes for MakeXRef and XRefConvert

## 1.11 XRef/Thanks

I'd like to thank some people for ideas and beta testing of the XRef-System :

Marius Gröger      - for his ideas, beta-testing, the small shared library  
                              init code and for correcting my english !

Alexander Barthel - for his patience of beta-testing my tools and for his  
                              GoldED ARexx-macros and PatchScreen tool.

---

Julian Scheid        - for his beta-testing and for correcting my english !

Manuel Nuñez        - for his ideas.

And some other people :

Dietmar Eiltert     - for GoldED!

Markus Wild         - for his work in the unix area for the Amiga (GNU-C,  
ixemul.library and NetBSD) !

Micheal Sinz        - for the great Enforcer!

Fred Fish            - for his PD-Fish series!

Urban Mueller       - for his Aminet work !!!

And any other people, who helped to make the Amiga the only possibility!

## 1.12 XRef/Address

If you have any comments, suggestions, bug reports or ideas please contact me :

Mail:

Stefan Ruppert  
Windthorststraße 5  
65439 Flörsheim am Main  
GERMANY

EMail: ruppert@vs3.informatik.fh-wiesbaden.de

## 1.13 XRef/Compiler and Software Tools

Compilers used:

DICE-C              This isn't a tool but a great PD/SW C-Compiler I am using for  
V2.07.54            two years. Meanwhile I switched to SAS-C, but I look out for the  
                     commercial version of DICE. I am still using the DMake make  
                     utility for my projects.

SAS-C                The best C-Compiler for the Amiga. Think about the optimizer.  
V6.51                He reduces 2K from the xref.library .

GNU-C                I used this compiler for porting some unix tools and programs.  
V2.6.0               The Amiga port of the GNU-C/C++ compiler makes it easy to port  
                     unix sources to the Amiga. Thanks to Markus Wild (ixemul.library)  
                     for his work in this area, also to Fred Fish, Philippe Brand and  
                     Lars Hecking for porting the latest version of GNU-C.

Software-Tools used:

---

|                      |  |
|----------------------|--|
| Enforcer             | A tool to detect illegal memory access. Great!   |
| findhit,<br>findline | Tools to find enforcer/mungwall hits in the source code. This works only if you have turned line debugging on  |
| DOSTrace             | A Program that protocols dos.library function calls like Open() Read(), Write(), Lock(), UnLock(), ...   |
| makedoc              | tool to extract documentation from source code or any other Textfile. It's just like autodoc from Commodore DevCon disks, but has a more flexible extraction and language support. |
| pragma_gen           | tool to generate SAS-C pragma files. This program generates also tagcall pragmas.  |

## 1.14 XRef/AmigaGuide Setup and Problems

There are some rules you have to follow, if the XRef-System should work correctly with AmigaGuide:

Setup:

- You must setup the ENV:AmigaGuide/Path file correctly. In this file, you have to setup all directories with full pathnames, where all your AmigaGuide databases reside (on one line, separated by spaces).

Problems :

- For include files and all other files that have only a main node or links with a line number, the ENV:AmigaGuide/Path directories doesn't work due to a bug of AmigaGuide (I suppose). To fix this, generate xref entries for such files with full pathnames like this:

```
MakeXRef FROM include: TO include.xref CATEGORY Include PATH AG:SysInc
```

In this release, I included the include-xref files with the "AG:SysInc" global path for the system includes and "AG:OtherInc" global path for the xref-includes. Note that this global path is only stored one time in a xreffile ←

,  
so it doesn't take up much disk or memory space; yet the full path is used in ←  
all  
dynamic nodes and in generated AmigaGuide files. Therefore, make sure that the specified path name is not too long.

- The amigaguide.datatype and DOSTrace 2.12 don't work together:  
If DOSTrace is running, the amigaguide.datatype cannot open any database.  
I have no idea why this happens.

You can test this with MultiView or any other program that uses this datatype.

These problems appeared in amigaguide.library V40.4 and amigaguide.datatype V40.11 (my actual system version).

Compatibility:

- the column support works only with V40 and above. I used new features from the amigaguide.datatype V40 (settabs)!