

Hyper v1.15a

COLLABORATORS

	<i>TITLE :</i> Hyper v1.15a		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 5, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Hyper v1.15a	1
1.1	Hyper v1.15a - © Koessi 92	1
1.2	Getting started	2
1.3	Icons & Tooltypes	3
1.4	Compatibility with `Am*gaGu*de'	3
1.5	The Hyper window	4
1.6	Keyboard commands	5
1.7	The Hyper ARexx Port	5
1.8	The `help' tool	6
1.9	Author & Credits	6
1.10	test.rexx	7
1.11	Access hyper from other programs like this:	7

Chapter 1

Hyper v1.15a

1.1 Hyper v1.15a - © Koessi 92

```

*****
*
*           This program is SHAREWARE !
* To register and to support further development send
*
* ----->> DM 10,00/$ 10,00/£ 5,00 <<-----
*
* to: Koessi, Peterstr.60, W5609 Hueckeswagen, Germany
*
*****

```

Credits

Startup

First Aid

What is it good for ?

Hyper will show documents that are written to be used with the legendary

Am*gaGu*de

from Commo. Several authors do already use it, but I - as a more normal user - have no access to it. So I decided to write my own version.

Where is the problem ?

Hyper needs Am*gaDOS 2.x [U2 \8-(] & a lot of memory ...

This Demo may be copied and spreaded as long as all files are kept untouched with it and the price for the disk containing it is not higher than DM/\$ 5,00. If U paid more, U're fooled and DON'T BUY ANYMORE FROM THAT RAT.

1.2 Getting started

Put Hyper in the SYS:Utilities drawer and assign "HYPER:" to Ur ↔ docs-drawer.

Hyper can be called from the CLI by typing:

```
Hyper [<FILENAME>] [DOC/K <chapter>] [SCREEN/K <publicscreen>]
[X/N <n>] [Y/N <n>] [W=WIDTH/N <n>] [H=HEIGHT/N <n>] [G=GADS/T ON|OFF]
[F=FONT/K <name>.font<n>] [S=SLEEP/S] [Q=QUIT/S]
```

- FILENAME** If U don't supply a filename to read in, Hyper will open the asl-requester first. If the directory containing the hyper-docs is assigned as "HYPER:" this one is the default. Supplied filenames are searched first in the current directory, then in HYPER: and there also with the suffixes ".hyper" and ".guide" appended.
- DOC** If no chapter is given or if it is not found in the document, the "main" node is shown as the first page.
- SCREEN** If U know the name of a public screen, U want Hyper to appear on, supply that name (in strokes if it contains spaces). Hyper looks for the colours defined via the SA_Pens tag of the foreign screen. If Ur program opens that screen, try to arrange them in a good-looking 3D-style to make hyper look fine. By default the window will appear on the WorkBenchScreen.
- X, Y** Window positions preferred to start with (default 0, 0).
- WIDTH**
HEIGHT Window dimensions preferred to start with (default 640x200). -1 will use the screen's maximum values.
- GADS** Choose whether U want (ON) the gadgets (default is OFF). The shortcuts are always supported. With the shortcut <shift-g> U may toggle the gadget's state later also (try now).
- FONT** Specify the font and its size to render the text as a string formatted like this: "pearl.font8" (which is the default font) Proportional fonts are supported, but tabs are uncontrolled.
- SLEEP** This option will make Hyper stay in the background until it is called via its ARexx-Port. There is also an menuitem added to the workbench-Tools-menu. Use this in the startup-sequence if U have enough memory: "run nil:> Sys:utilities/Hyper S".
- QUIT** This will force a running instance of Hyper to quit immediately.
-

Of course the window is an AppWindow, so drag&drop icons of the files U want to read into the Hyper Window and on it goes.

On Workbench it behaves completely normal: just doubleclick its icon

U may shift-select the first project to work with.

1.3 Icons & Tooltypes

Tooltypes supported since v1.11/2 are the same as the commandline parameters

They are recognized in the following order:

- a) the icon of the program Hyper itself sets the default values
- b) those are overridden by the commandline parameters
- c) and/or by the tooltypes of the project's icon, if that is picked on the workbench or if the project is specified in the commandline with its complete path. Easy, isn't it ?

e.g. set the tooltype "DOC=gadgets" in the icon of the project "hyper.hyper" and each time U select that text from the aslrequester or click that icon or drag&drop it or type in the cli "Hyper HYPER:hyper.hyper" or send that string as a message to Hypers

```
ARExport
  the chapter gadgets
of the document will be shown. That's it.
```

1.4 Compatibility with `Am*gaGu*de`

The only source I found were some documents published in the Fish- ↔ Library.

I don't know wether there are more keywords possible, than those guys used.

Hyper ignores the @height information.

If the @width is -1 or 255 Hyper will center each chapter corresponding to its real width.

If the nodes have an extra name, Hyper presents it as a headline on top of the page. The windowtitle is taken from the @DataBase-name or the filename.

If the chapter contains links, they will appear in inversed colors. Double-click on those keywords to jump into the associated chapters - that's the interactivity...

Hyper also supports links to nodes inside other docs and it reminds the

way back as the previous page of the new docs first shown page, accessible from that by <p>, <backspace> or <right mousebutton> (see shortcuts).

I've heard, that 'Am*gaGu*de' is a shared library, but I don't know any of it's calling conventions, so Hyper cannot emulate that calls. Instead I've implemented an

```
ARexx-Port
```

```
.
```

1.5 The Hyper window

The window will appear on the WorkBenchScreen (default size is 640 ← x200).

If U supply a SCREEN=xxx argument, the window will be guest of that public screen if that can be found. Hyper looks for the colours defined via the SA_Pens tag of the foreign screen. If Ur program opens that screen, try to arrange them in a goodlooking 3D-style to make hyper look fine.

Of course the window is an APPWINDOW, so drag&drop icons of the files U want to read over the Hyper Window and on it goes.

U may rearrange the window in the usual ways with the system-gadgets. If there are gadgets at the bottom of the window, they control these items:

```
Load Doc      - if U want to read in another document
First Page    - will show the '@Main' chapter
Prev Page     - will go back one chapter in the document
               the keywords '@Prev' and '@Toc' are recognized here
Next Page     - will move forward to the next page in the document
               (this may be confusing, because the chapters do not need to
               be arranged in a logical order)
               the keyword '@Next' is supported if found
Sleep         - will make Hyper close its window and wait for a message to
               its
               ARexx-Port
               . Also an AppMenuItem is
               added to the WorkBench's Tools-menu named "WakeUpHyper".
```

Scroller gadgets in the right- and bottomborder of the window allow to move inside the text if it is bigger than the window.

There are several shortcuts build in to access functions via keyboard:

If the document-file contains links, they will appear in the shown text in inversed colors. Doubleclick on those keywords to jump into the associated chapters - that's the interactivity...

1.6 Keyboard commands

There are several shortcuts build in to access functions via keyboard:

```
Return      - <q>, <Esc>
Load Doc    - <l>, <Ins>
First Page  - <f>, <Home>
Prev Page   - <p>, <PgUp>, <Backspace>, <Right mousebutton>
Next Page   - <n>, <PgDn>, <Space>,
Sleep       - <s>, <Enter>, <Return>
Gads on/off - <Shift-g>
Help        - <h>, <Help>   (try to load this text)

Scroll-up   - <cursor-up>, <kp8>
Scroll-down - <cursor-down>, <kp2>
Scroll-left - <cursor-left>, <kp4>
Scroll-right - <cursor-right>, <kp6>
```

1.7 The Hyper ARexx Port

To make Hyper accessible for your applications I've build in an ARexx port. ←

To keep things simple there's only one way to use it:

Send a REXXMsg with one string in the first REXXArg slot.
This string should have this format:

```
"[filename] [DOC chapter] [SCREEN publicscreen]
[X n] [Y n] [W n] [H n]
[G ON|OFF] [F=FONT <name>.font<n>] [S] [Q]"
```

The options are the same as those for calling Hyper from the CLI

The portname is "HYPER_RXPORT".

There are several ways to access a running instance of Hyper:

An example ARexx-script called
test.rexx
is part of this
distribution. It shows, how simple it is, to use Hyper from inside ARexx-
powered software.

Simply calling hyper again will detect the port, send the commandline as
a REXX-Message and terminate itself. This also works on the Workbench
with projects calling Hyper via its icons "
default tool
".

The provided tool "
help
" will do the same being much smaller.

If Hyper is sleeping, a wake-up-call can be send from the WorkBench's Toolmenu, the item is named "WakeUpHyper".

1.8 The 'help' tool

The easiest way to control Hyper from CLI or batchfiles is to use the provided tool "help".

```
help [<FILENAME>] [DOC/K <chapter>] [SCREEN/K <publicscreen>]
[X/N <n>] [Y/N <n>] [W=WIDTH/N <n>] [H=HEIGHT/N <n>] [G=GADS/T ON|OFF]
[F=FONT/K <name>.font<n>] [S=SLEEP/S] [Q=QUIT/S]
```

The options are the same as those for calling Hyper from the CLI

.

The

C-source

of "help" is part of this distribution, because it may give U an idea on how to use Hyper from inside your own programs.

1.9 Author & Credits

Koessi © 9-11.92

phone germany 02192 7630

```
*****
*
*          Bugreports & special wishes are welcome!
*
* A file called "strings.c" is included.  If U want a
* version in your own language, feel free to change that
* file and send it to me.  No problem, but keep the
* strings short (the doc ...).  Also I'm strongly intere-
* sted in all the developers supporting stuff from Commo
* I have no modem - perhaps somebody can help me -
*          thanx and have fun with your machine.
*
*****
```

thanx to ...

Matt Dillon - for DICE

Fred Fish - for the Library

... and all authors of nonorlesscommercial Am*ga software.

... and of course for Ur supply:

```

Rudolf Rauh          Gelsenkirchen
Thomas Kielbassa     Hamburg
Christian König      München
John Lehmkuhl        Denmark
APC&TPC ComputerClub Schonstett
Ekke Verheul (Asware) Rotterdam (greet Ur clients!)
Dietmar Eilert       Aachen      (still open wishes ?)
Michael Goedecke     Walle

```

1.10 test.rexx

```

/* argstr format:

   [FILENAME]
   [DOC/K chapter]
   [SCREEN/K publicscreen]
   [X n]
   [Y n]
   [W=WIDTH/N n]
   [H=HEIGHT/N n]
   [G=GADS/T ON|OFF]
   [S=SLEEP\S]
   [F=FONT/K <name>.font<n>]
   [Q=QUIT/S]
*/

PARSE ARG argstr

IF (SHOW('P', 'HYPER_RXPORT')) THEN DO

   ADDRESS 'HYPER_RXPORT'

   argstr

END

```

1.11 Access hyper from other programs like this:

```

/***** help.c *****/
*
*
*          v1.15
*
*          © by Koessi
*
*          Tuesday, 21 Nov 1992
*
*
* this is an help showing the usage of
* "hyper" from inside another program
*

```

```

* Compile with DICE: *
* dcc help.c -ohelp -rr -2.0 *
* *
*****/

#include <exec/types.h>
#include <exec/execbase.h>
#include <exec/memory.h>
#include <dos/dos.h>
#include <dos/dostags.h>
#include <rexx/storage.h>

/* Prototypes */
#include <clib/exec_protos.h>
#include <clib/dos_protos.h>
#include <clib/alib_protos.h>
#include <clib/rexxsyslib_protos.h>

#define MSG struct Message
#define RMSG struct RexxMsg
#define MSGP struct MsgPort
#define SIZE 32

extern void SendRxMsg(char *);
extern __stkargs void _main(short, char *);
extern int main(int, char **);

#define NUMARGS 12

const char version[] = {'$', 'V', 'E', 'R', ':', ' ' };
const char taskname[] = "Help v1.15 @ Koessi 92 - Funware\n";
const char exthelpstr[] = "32mUsage:\tHelp [FILENAME] [DOC chapter]"
                        "[SCREEN publicscreen]\n"
                        "\t\t[X n] [Y n] [WIDTH n] [HEIGHT n] [GADS ON|OFF]\n"
                        "\t\t[FONT name.font<n>] [SLEEP] [QUIT]0m\n"
                        "\n\tcall Hyper (>= Ver1.15) via its ARexx-port:\n"
                        "\tFILENAME\tshould be a hyper-text-file\n"
                        "\tDOC/K\t\trequest a special chapter\n"
                        "\tSCREEN/K\t\tmake Hyper appear on that screen\n"
                        "\tX/N, Y/N\n"
                        "\tW=WIDTH/N\n"
                        "\tH=HEIGHT/N\t\tset position&size for the window\n"
                        "\tG=GADS/T\t\ttoggle gadgets ON/OFF (default is OFF)\n"
                        "\tF=FONT/K\t\tuse this font to render text (default is \" ←\n"
                        "pearl.font8\")\n"
                        "\tonly if Hyper is already running:\n"
                        "\tS=SLEEP/S\t\tstart Hyper into the background\n"
                        "\tQ=QUIT/S\t\tend Hyper and free memory\n"
                        "\t\t\t";

const char template[] = "FILENAME,DOC/K,SCREEN/K,X/N,Y/N,W=WIDTH/N,H=HEIGHT/N,"
                        "G=GADS/T,F=FONT/K,S=SLEEP/S,Q=QUIT/S";
const char portname[] = "HYPER_RXPORT";
const char command[] = "SYS:Utilities/hyper S";
const char error[] = "\n\n32m***ERROR0m";

const void *argarray[NUMARGS] = {0}; /* holds argptrs */

```



```

        break;
    }
}
PutStr("\n");
if (port)
    SendRxMsg(arg);
else
    PrintFault(ERROR_OBJECT_NOT_FOUND, portname);
}
FreeArgs(rda);
}
else
    errorcode = IoErr();

FreeVec(rdargs);
}
else
    errorcode = ERROR_NO_FREE_STORE;
}
if (errorcode)
{
    PutStr(exthelpstr);
    PrintFault(errorcode, error);
}
}

/*****
*
*
*   FUNCTION: SendRxMsg
*
*
*   INPUT:   char *msgtxt
*
*   OUTPUT:  void
*
*   NOTE:    like cmdline
*
*****/

void
SendRxMsg(char *msgtxt)
{
    MSGP *reply_port;
    if (reply_port = CreateMsgPort())
    {
        void *rx_msg; /* casted to parts of a REXXMSG struct */
        if (rx_msg = AllocVec(sizeof(RMSG), MEMF_PUBLIC|MEMF_CLEAR))
        {
            ((struct Node *)rx_msg)->ln_Type = NT_MESSAGE;
            ((MSG *)rx_msg)->mn_ReplyPort = reply_port;
            ((MSG *)rx_msg)->mn_Length = sizeof(RMSG);
            ((RMSG *)rx_msg)->rm_Args[0] = msgtxt;

            Forbid();
            MSGP *rx_port;

```

```
if (rx_port = (MSGP *)FindPort(portname))
{
    PutMsg(rx_port, (MSG *)rx_msg);
    Permit();
    WaitPort(reply_port);
    ReplyMsg(GetMsg(reply_port));
}
else
    Permit();

FreeVec(rx_msg);
}
DeleteMsgPort(reply_port);
}
}
```
