

HowDoYou.hyper

COLLABORATORS

	<i>TITLE :</i> HowDoYou.hyper		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 6, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	HowDoYou.hyper	1
1.1	How Do You ... ? (Tue Nov 3 15:18:24 1992)	1
1.2	How Do You ... ? : Set the PowerVisor screen to interlace	1
1.3	How Do You ... ? : Start debugging a program	1
1.4	How Do You ... ? : Interrupt PowerVisor output	2
1.5	How Do You ... ? : Restore PowerVisor after ARexx error	2

Chapter 1

HowDoYou.hyper

1.1 How Do You ... ? (Tue Nov 3 15:18:24 1992)

Contents:

- Set the PowerVisor screen to interlace
- Interrupt PowerVisor output
- Start debugging a program
- Restore PowerVisor after ARexx error

Various:

- Back to main contents

1.2 How Do You ... ? : Set the PowerVisor screen to interlace

The easiest way to do this is the following sequence of commands :

```
< mode lace <enter>
< fit main <enter>
< saveconfig <enter>
```

See the appropriate documentation for more information (mode , fit and saveconfig).

1.3 How Do You ... ? : Start debugging a program

The easiest way to start debugging is by using the standard fdebug alias (this alias is defined in the PowerVisor-startup file). This alias opens the source and debug windows and redefines some keys so that you can easily singlestep in your programs (the defined keys are like the keys used by the 'monam' (© HiSoft) debugger). The 'fdebug' alias will show all the keys that are defined. The 'fdebug' alias opens the source window. This window is only used for sourcelevel debugging. If you don't need this you can close this window with the swin command.

After opening the fullscreen debugger with 'fdebug' you also need to

load the program. For programs intended to be started from the Shell or CLI you can use :

```
< debug l <programname> <enter>
```

It's better not to use this command if you have AmigaDOS 1.3 since the 1.3 version of PowerVisor does not create a CLI structure for the loaded program.

The best way to load a program (in case you have AmigaDOS 1.3 or if you want to debug a WorkBench program) is with :

```
< debug n <enter>
```

PowerVisor will now wait for the next program. With this command it is possible to debug WorkBench programs. After typing 'debug n' you can double click on an icon and PowerVisor will intercept the program. Note that you can't debug resident programs!

Because 'debug n' does not load the symbols ('debug l' does) you have to load them yourselves :

```
< symbol l <programname> <enter>
```

If you want to load the source (for sourcelevel debugging) you must use the following command (also for 'debug l') :

```
< source l <programname> <enter>
```

Note that you can only use sourcelevel debugging if you have compiled or assembled your program with special options (these options depend on your compiler or assembler).

See `Debugging` for additional information about debugging.

1.4 How Do You ... ? : Interrupt PowerVisor output

Normally you can interrupt PowerVisor output with the <esc> key. You can redefine the interrupt key to another key but <esc> is the default. See the `prefs` command for more information about redefining standard keys.

1.5 How Do You ... ? : Restore PowerVisor after ARexx error

Normally ARexx scripts started from within PowerVisor (with the `rx` command) run asynchronously with PowerVisor. This means that you can still type commands on the PowerVisor commandline while the ARexx script is running. However, it is possible that an ARexx script wants complete control over PowerVisor and the commandline. In that case this script will probably use `sync` to synchronize PowerVisor to the script. After the script is done it will use `async` to restore PowerVisor. But, for one reason or another, it is possible that the ARexx script 'forgets' to use

the 'async' command (because of a bug or because the ARexx script is interrupted). PowerVisor has no way to know that the ARexx script is ready so the PowerVisor commandline will remain locked forever. You can solve this problem by sending an 'async' command from the Shell or CLI commandline like this :

```
Cli< rx "address rexx_powervisor async" <enter>
```