

# Installation Instructions

## HELIOS-ATW

This is Beta 1 version of Helios 1.1 for the ATW. It now also includes the latest version of X-windows for the Issue 3 machines. Please do not pass this system to anyone else and please keep your comments private.

This version of Helios is upwards compatible with V1.0. Any program which runs under 1.0 will run under 1.1. However, programs compiled under 1.1 will NOT run properly under 1.0, so if you are developing software please note that you will still need to create programs under 1.0 until 1.1 is released in February 89.

The Helios kernel has been completely rewritten in C for ease of maintainance and for portability. Despite this the new system is faster than the old. The network server and task force manager have also been rewritten. Note that the format of resource maps has changed slightly; it now matches the documentation. See example maps in /helios/etc. Note also that the startup entry in loginrc is different as the netserver now starts up the TFM (and the new batch server) for you. We have spent a great deal of time making the system more robust when using the TFM; if you have multiple transputer systems please set the cdl flag and test the multiprocessor support as much as possible. This release also supports multiple networked users; more details will be sent with Beta 2.

There is a new server /pipe which is used in preference to /lifo when pipes are used from the shell or from the CDL. /pipe is more efficient then /lifo as it does not buffer; if buffering is required then /lifo must be used explicitly.

### How to install Helios-ATW

1. For your own security, make one set of backup diskettes.
2. Make a directory called \HELIOS on the C partition of your hard disc.  
**Note:** it must be called C:\HELIOS unless you intend to edit the file 'host.con' which contains a reference to the directory C:\HELIOS.
3. Copy the entire contents of the supplied diskettes to \HELIOS.  
**Note:** The subdirectory names used within the \HELIOS directory MUST correspond to those provided on the diskette. The easiest way to do this is to use the GEM mouse copy to transfer all subdirectories.

Before you can start Helios you must first install the transputer driver traps. This is done by executing the command P3LINK.PRG, normally by double clicking on it. Note that this program used to be called INSTALL.PRG; we have changed the name and it now handles the SCSI drive. The program runs, installs the traps and returns. If you forget to do this you will get a screen covered in little bombs and the ST will crash when you try and run the server, so it is normally a good idea to place the P3LINK.PRG program in the \AUTO folder so that it is done each time you start.

The server itself is started by double clicking on the SERVER.PRG program. You should then see a banner from the server which proceeds to open the file HOST.CON in the current directory to determine various options.

Helios-ATW then starts, and it comes up with a window manager provided by the IO server and displayed on the ST screen. This is similar to the old NOWINDOW system except that it is implemented on the ST and it offers redisplay of entire windows when you use ALT/F1 to switch screens. This option is selected by entering the line SERVER\_WINDOWS in the HOST.CON file.

You can replace this with either the simple window manager which runs on on the ATW and displays on the transputer video screen, or by X windows. In either case you must remove the line SERVER\_WINDOWS from the file HOST.CON which ensures that Helios now reads the file \HELIOS\LIB\WINDOW. As supplied this

is the old NOWINDOW system using the ST screen.

In order to install the simple window manager save the file `\HELIOS\LIB\WINDOW` and rename `\HELIOS\LIB\WINDOW.ABQ` as `\HELIOS\LIB\WINDOW`. To install the X system, rename `\HELIOS\LIB\WINDOW.X` as `\HELIOS\LIB\WINDOW`; fuller instructions are given below.

### Using the system with Kuma interface boards

If you are using the Kuma interface boards to drive an ATW (or any transputer system) from an ordinary ST you must use a different interface program. Instead of running the program called `P3LINK.PRG` before attempting to use the Helios server, you must continue to use the program previously supplied called `KINSTALL.PRG`. As with `P3LINK.PRG`, this is best placed in the `\AUTO` folder so that it is executed automatically.

### Helios directory structure

The directories within the `\HELIOS` directory are used by Helios when it starts. It refers to them as `/helios/xxx` thus enabling you to alter the name of the actual directory mapped to `/helios` on the ST if required.

The `lib` subdirectory contains library stubs and C prologue, which are required for linking. It also contains dynamically loaded modules, such as the RAM disc handler, as well as the system image itself.

The `bin` subdirectory contains all the commands that are currently available under Helios.

The `include` subdirectory contains the C header files for the system which define all the system data structures.

The `example` subdirectory contains a simple makefile and a highly complex test program. The command `'make hello.z'` should cause an executable module called `hello.z` to be created. This can be tested by giving the command `hello.z` which should cause the program to run.

The `etc` directory contains some useful text files. It contains some example resource maps; this is explained below. It also contains a file called `nopasswd` which has some entries for possible users of Helios. Normally Helios boots and starts at the directory `/helios`, executing the files `loginrc` and then `eshrc` held there. If you have an ATW which is used by more than one person you can add their names to the file `nopasswd` following the format outlined there. This has a user name, a (possibly blank) password, user and group numbers, real name, home directory and shell to be called.

If you now rename or copy the file to `etc/passwd` then Helios will read this file and prompt you for a user name. Depending on the name you give, Helios will set up a private home directory instead of `/helios` and proceed to execute the `loginrc` and `eshrc` files held there. This is useful for customising individual user's systems. If you get it wrong you can always alter it again from GEM.

If you create a file called `etc/motd` then Helios prints the contents of this out before starting the system. This is useful for announcing messages of the day.

### Resource maps

The `etc` subdirectory also contains suitable resource maps for simple transputer configurations. Note that the format of resource maps has changed between V1.0 and V1.1 of Helios, you must use the new version of `rmgen` to construct the maps, and maps will need a specification of the master node and the name of the device driver used to reset nodes. The driver `ab_ra_r3.d` is used for issue 3 ATW boards; the driver `in_ra_b8.d` is used for B004/B008 style boards. The system is supplied with a resource map called `etc/default.map` which is the binary version of the text resource map `etc/default.rm`. This defines the smallest possible configuration consisting of a single processor connected to the I/O server.

Once you are happy with Helios and have read the chapter in the User Manual concerning resource maps and networks then you will want to create a new resource map which reflects the layout of your own transputer network. When you have done this you should compile the text map with `rmgen` and place the result in

etc/default.map, or alter the loginrc file which contains a reference to this filename. Then reboot Helios and you should get a log of the nodes which have been booted. Error messages will be given if the physical configuration of the network does not match that specified in the map.

Once an additional processor has been booted you should be able to access it via /<name>; for example

```
ls /01/tasks
```

should list the tasks in processor named 01. Note that processor names do not appear in the name table until they have been accessed.

You can run commands in remote processors with the **remote** command - see the User's Manual for details. You can also start up remote programs yourself and an example of this is given in the Developer's Manual.

### X-windows

Included in this package is the next beta release of X-windows. This version of X is much, much faster, and we believe now quite usable.

Do not attempt to use or experiment with X if you are a new Helios user. Please spend some time getting used to the standard Helios environment, and only then see what X-windows adds to this. If you try everything at once you are liable to be overwhelmed with it all.

To install X, you must save the file \HELIOS\LIB\WINDOW and replace it with the file \HELIOS\LIB\WINDOW.X which will load the X system on startup. You must also ensure that the line SERVER\_WINDOWS has been removed from the HOST.CON file and that you have installed all the diskettes provided in this release. You must also edit your loginrc file to include the line

```
setenv DISPLAY abaq:0.0
```

The X system for V1.1 attempts to read a resource file before it starts which tells the system how to configure itself. The resource file is called XRC and should be kept in the \HELIOS directory. The file consists of entries interspersed with comments. Entries all have the same format, consisting of an upper case keyword, an equals sign and then one or more parameters. The parameter list is terminated by a character which cannot be a parameter - normally a newline. Spaces around the equals sign are optional, but the keyword must be in upper case. If a keyword is specified more than once in a file, the first entry is used. If no parameters follow the equals sign then default values are used. Entries can appear in any order.

Only two entries are currently defined. The keyword MODE is followed by the number 0 (default), 1 or 2 to specify mode 0 1 or 2. Mode 0 is in grey scale; you can also specify mode 4 if you are very rich and have a mode 0 colour monitor. Mode 3 is not supported.

The keyword SCREEN is used to specify the monitor setup information. It takes between one and seven parameters. The parameters are

- Horizontal total
- Horizontal displayed
- Horizontal sync
- HSYNC width
- Vertical total
- Vertical displayed
- Vertical sync

The actual numbers used are somewhat cryptic, but they are used to control the size and positioning of the picture on the monitor screen. The XRC file provided has suitable example values.

Now try rebooting Helios. If everything works OK the screen will be painted with a crosshatch pattern and an X shaped cursor will appear. Eventually a window will appear with the normal Helios startup messages in it. If you move the cursor over the window it will change shape, indicating that input from the keyboard will now be sent to the Helios terminal emulator. Start the X window manager by typing the command

```
/helios/clients/uwm &
```

which should print "PING" on the ST screen when it is ready.

You should now be able to use the X system. Look in the file `/helios/uwmrc` to see what actions the window manager will perform when the mouse and modifier keys are pressed. `uwm` intercepts both mouse buttons when pressed on their own, or when used in conjunction with the SHIFT or CTRL keys. In order to pass mouse clicks on to client programs you should either hold down the ALT key and click the mouse button, or engage the CAPSLOCK key. To move a window or icon press and hold down the left mouse button whilst over a window. To resize a window press and hold down SHIFT and the left mouse button. To iconify (or deiconify) use CTRL and the left mouse button.

The `/helios/clients` directory contains several other X clients. Most of these will print a usage message if executed as

```
<clientname> -help
```

Other examples and demos are held in the directories with those names.

If you want to experiment compiling programs under X you may need to fix the search paths for the compiler; note that all the header files for X are placed in `/helios/include/X11`. Programs that use the X library must be linked with `/helios/lib/Xlib.def`. Programs that use the X toolkit and Widget libraries must be linked with the files `Xtlib.def` and `Xawlib.def` also found in `/helios/lib`. Note that if you use the Widget library you will also need to link with the floating point libraries. The order of linking these files is important - see the makefile for `xclock.c` in the `examples` directory.

Included in this release are some utility programs. `fc` is the font compiler. It turns text `.bdf` files into binary `.snf` files. `showsnf` is the X font displayer. You can use it with the `-v` (verbose) and `-v -v` (very verbose) options. `txcp` is a version of `cp` is a version of `cp` used for copying text files from, say, the ST filing system to the RAM disc. `ucc` is a front end processor for the C compiler, the assembler and object editor. It acts very much like `cc` under U\*\*x. Try using the `-help` option for more details.

### Bugs

Bug reports for Beta 1 V1.1 should be sent in writing to us, either by post or by fax to 0749 4977, marked for the attention of the technical support group. We are very interested in getting bug reports as soon as possible.