

SSN/21.2

Starlink Project
Starlink System Note 21.2

M J Bly
9 April 1996

STARX — X Library linking

Contents

1	Introduction	1
2	Why have STARX?	1
2.1	Solaris 2	1
2.2	PCs with Linux	1
3	How it works	1
3.1	Link Script	2
3.2	Library Aliases	2
3.3	Linking	2

1 Introduction

The STARX Starlink Environment package provides a system independent method of linking with the X libraries, to relieve the other subroutines libraries and applications of having to worry about the various complications involved.

STARX allows the X Library linking strategy to be changed easily, without changing any of the other Starlink Environment components.

The system described has been tested for DEC AXP/Digital Unix, Sun Sparc/SunOS4, Sun Sparc/Solaris (version 2.3, 2.4 and 2.5), and Intel PC/Linux. It does not appear to hinder dynamic linking where available.

2 Why have STARX?

2.1 Solaris 2

When we came to build the USSC under Solaris 2.4, it quickly became apparent that the simple strategy for linking with the X libraries employed under Solaris 2.3 and on other platforms didn't work. Changes under Solaris 2.4 meant that various of the X libraries are interdependent on each other.

This is fine using the default dynamic linking — the linker is capable of resolving such dependencies. However, for static linking, the norm for Starlink applications, the linker cannot successfully resolve such dependencies.

One way round this problem is to specify the libraries concerned twice each in the chain of libraries for linking. However, because of the way the Starlink link scripts work, it is not possible to do this — multiple occurrences of library names are removed to shorten the number of linker arguments.

STARX resolves the problem by using alias names for the additional occurrences for the libraries concerned.

2.2 PCs with Linux

The port of the USSC to Linux on Intel PCs required another 'fix' to accommodate X Library linking.

Under Linux systems, the X Libraries are in a different place than on most Unix systems, and are not searched automatically by the linker.

The Linux version of STARX provides the path for the linker to find the X libraries, in addition to the name of the library required.

3 How it works

There are two components to STARX: the link script, and where necessary, library aliases.

3.1 Link Script

The link script `starX_link` simply takes the form of an `echo` command with a string of required libraries. The default version is this:

```
% echo -lX11
```

which is all that is necessary for DEC Alpha/Digital Unix and Sun Sparc/SunOS4 systems.

For Solaris 2.5, the script is this:

```
% echo -lXext -lX11 -lstarXext -lstarX11
```

The last two, `-lstarXext` and `-lstarX11` are the alias names for the two X libraries that need to appear twice, `-lXext` and `-lX11`.

For Linux on Intel ix86 systems, the script is this:

```
% echo -L/usr/X11/lib -lX11
```

3.2 Library Aliases

The alias names for the libraries are provided by soft links to the actual libraries. These are created in the `/star/lib` directory (static library) and `/star/share` directory (shared libraries).

For Solaris 2.4, the links created are:

```
/star/lib/libstarX11.a -> /usr/openwin/lib/libX11.a
/star/lib/libstarXext.a -> /usr/openwin/lib/libXext.a
/star/share/libstarX11.so -> /usr/openwin/lib/libX11.so
/star/share/libstarXext.so -> /usr/openwin/lib/libXext.so
```

The default is to soft link to the Openwindows libraries, though other X libraries may be used (if installed) by changing the definition of `XLIBS_PATH`. This should be done by setting the environment variable in the `mk` script and NOT by changing the default in the `makefile`.

There are no library alias requirements for the other USSC supported systems.

3.3 Linking

Linking with the X libraries is typically done in conjunction with linking with one or more of the Starlink Environment graphics packages. Where this is the case, users will not need to take any action to use the new linking system, since the graphics systems have been modified to call the new `starX_link` script where appropriate.

Where a program needs to link with just the X libraries, a command of this form should be used:

```
% cc myprog.c -o myprog -L/star/lib 'starX_link' ....
```

The `-L/star/lib 'starX_link'` section pulls in the X libraries in the correct manner for the system.