Applications which continue to rely on the BSD derived libraries of SunOS will require more conversion effort.

The *SunOS 4.1 to SunOS/SVR4 Compatibility and Migration Guide* is a Sun publication that can help with this process. It details the specific migration of commands, utilities, library routines and system calls. Furthermore, lint libraries on SunOS 4.1 corresponding to SVID Issue 2 and SVID Issue 3 can also help.

Dynamic Linking

Sun recommends that applications be dynamically linked. Dynamic linking benefits applications by reducing the overall working set of most executables. This improves overall runtime performance and insulating the application from changes in the underlying operating system.

Applications that are intended to run on SunOS/SVR4 using the SunOS 4.X Binary Compatibility Package (BCP) must be dynamically linked. The BCP is a transition tool that allows end users to migrate to SVR4 while running their existing SunOS 4.X software in the short-term, thereby providing a smooth progression and simplifying the creation of ready market for native SVR4 applications.

Looking ahead, developers who plan to release binary compatible SVR4 based SPARCware™ applications on the rapidly growing number of hardware platforms compatible to SPARC should know that applications are required to be dynamically linked by the SPARC Compliance Document (SCD2.0)—the SPARC binary compatibility guideline that enables shrink-wrapped applications.

ANSI C

The purpose of ANSI C is to promote application portability, which is also a major goal of System V Release 4. Consequently, SunOS/SVR4 will provide an ANSI C compiler. Does this mean that applications have to be modified for use with the ANSI C compiler? Not necessarily. Software developers have several strategies available to them, depending on the time they care to invest and the features of ANSI C in which they are interested.

Developers can start by choosing an appropriate ANSI C strategy and then beginning work with Sun C 1.1, an ANSI C compliant compiler that will be available early in 1991. Sun C 1.1 can also be used to supply performance boosts to SunOS 4.X-based applications. Using Sun C 1.1 on SunOS 4.1, software developers who wish to take advantage now of ANSI features, such as function prototypes and improved type qualifiers, can do so knowing that changes to their sources will migrate to the ANSI C compilation environment on SunOS/SVR4.

An alternative strategy involving less effort in the short run is to leave common C source files unchanged. The ANSI C compiler offers options that accept inputs of Sun common C sources. In this strategy, the compiler merely warns where common C and ANSI C semantics differ. This option will also be available on SunOS/SVR4.

Plan Ahead for Success

Applications developers who begin to plan their migration strategy today, and who complete the checklist of preparatory steps on SunOS 4.1 will be rewarded with a smooth and easy transition to SunOS/SVR4.

The preceding is a reprint from the February 1992 edition of *Catalyst PartyLine*.

Preparing Applications for SunOS/SVR4 Transition

Plan Ahead for the SunOS/SVR4 Migration

As application developers look ahead and begin to plan for SunOS/SVR4, there are steps which they can take today which will simplify the migration of their applications in the future. Developers can prepare for migration by going through this five step checklist:

- Move to SunOS 4.1
- Convert to OpenWindows™
- Compile Using System V Libraries
- Dynamically Link the Application
- Choose an ANSI C Strategy

SunOS 4.1

SunOS 4.1 is the bridge to SunOS/SVR4 for applications already running on Sun platforms. Using 4.1 as a development platform gives developers a head start in migrating to SunOS/SVR4. The more work done on the application on SunOS 4.1, the less work will remain to be done on SunOS/SVR4.

Software engineers who use SunOS 4.1 as their development platform have access to the richest set of software development tools in the industry—available from Sun and also from third-party sources. And SunOS 4.1 is not only a great application development environment—it is a great application delivery vehicle.

A release of the application based on SunOS 4.1 is an alternative that should be considered seriously due to the large and rapidly growing installed base of SunOS 4.1 systems. Sun estimates that slightly more than half of its existing base of systems are currently running releases based on SunOS 4.1, and that number is growing by more than 10,000 systems monthly. A large percentage of those systems run OpenWindows.

OpenWindows

The next step for applications headed for SunOS/SVR4 is to convert existing SunView applications to run in a native OpenWindows environment. Although existing dynamically linked SunView applications continue to run in binary compatibility mode on SunOS/SVR4, the native window system for SunOS/SVR4 is OpenWindows. Therefore, SunOS/SVR4 does not provide the facilities for creating new applications based on SunView.

For many application developers, the strategy that makes the most sense is to plan on running in SunView binary compatibility mode on SunOS/SVR4 in the short term, and to convert to OpenWindows in the medium term. SunOS 4.1 is the best development system to use as a base for that strategy.

System V Libraries

To minimize the conversion effort needed later on SunOS/SVR4, applications should be compiled in the SunOS 4.1 System V environment and use the System V libraries. Because SunOS 4.1 conforms to the System V Interface Definition (SVID) Issue 2, applications can be written in a manner that is source-compatible with SunOS 4.1 today, and with SunOS/SVR4 tomorrow.