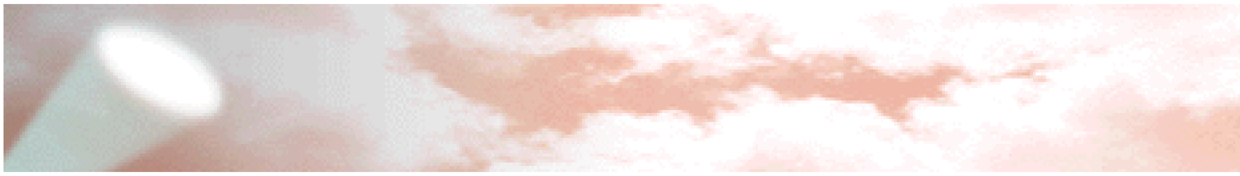

SPARCware Program/SCD Compliance

Introduction:
In November 1992, Sun handed off the SPARCware name to SPARC International with the express intent to have SPARCware be the binary compatibility banner for the entire SPARC community.

The objective of SPARC International's SPARCware program is to put in the marketplace thousands of SPARCware applications whose binary will execute unmodified on all SPARC compliant platforms. SPARC International accomplishes this through the combined efforts of SPARC system and software vendors.

When hardware and software vendors join the SPARCware program, they agree to adhere to the interfaces specified in the SPARC Compliance Definition (SCD). This enables software vendors to deliver a single binary version of their product that will run on all SPARCware compliant systems.

SPARCware Participation Benefits:
Participation in the SPARCware program gives software vendors and their customers assurance that a single SPARCware binary will run on any SPARCware branded system. In addition, SPARCware independent software developers can benefit directly. SPARCware represents:

1. A reduction in the number of variations of products to be maintained, since the single port will run on all SPARCware platforms.

2. An expansion of the independent software developer's market to include leading SPARC platforms.

3. An economical migration path to SVR4 and increased portability.

4. Long-term compatibility between the application, hardware, and operating system releases.

SPARC Compliance Definition (SCD):
The SPARCware compliance process ensures that an application developer adheres to the interfaces specified in the SCD. This document describes the SCD as well as the tools and techniques developed to simplify the porting process.

The current version of the SCD is SCD 2 and is based on the System V Release 4 Application Binary Interface (ABI) and the System V Interface Definition, Third Edition (SVID3). The SCD also includes facilities that all SPARC system vendors have agreed to supply with their system (e.g., MOTIF, Network Services, and X Window System™).

The basis for compatibility among SPARC systems and software is the SPARC Compliance Definition (SCD) which defines:

1. The minimum set of system functionality that must be provided by the hard-

ware and operating system (referred to collectively as the system).

2. The maximum set of operating system functionality that can be used by an application.

The SCD specifies the interface between the system and the applications software. The minimum set of system functionality is verified by requiring that each system vendor pass the SPARC System Compliance Test (SCT). The maximum set of system functionality required by an application is verified by requiring that an application pass the SPARC Application Verifier (SAV).

The SCD defines an application binary interface (ABI). This is the interface as it appears after the application has been compiled and linked. The SCD specifies the application and system interaction at the binary interface between application and system library, omitting implementation details of the system library and system kernel. This enables system manufacturers to implement their SPARC systems independently without regard to any other SPARC implementation.

The SCD has several important characteristics that establish it as a simple, stable, usable programming interface.

Specifically, the SCD describes the interfaces that all SPARC system vendors have agreed to make available on their systems. The SCD does not prevent system vendors from providing additional system functionality. This means that there may be some interfaces (beyond the scope of the SCD) that some SPARC vendors choose to supply.

The SCD attempts to define all needed functions without being excessively redundant. When multiple alternatives for providing functionality are available, only one or two choices have been selected. This avoids over-burdening systems with unnecessary requirements, potentially increasing system size, and complexity.

The SCD is a low-level definition of the interfaces which are supplied by system libraries. Higher level functions are expected to be supplied with the application.

The SCD defines complete interfaces (usually referencing other documents for the detailed definition of the interface). SCD 2 does not include interfaces for which detailed specifications do not exist.

The SCD defines interfaces for existing products. It does not define possible future products. This means that software vendors can use the interfaces immediately.

Vendors supplying SPARC systems have committed to supporting the interfaces in the SCD for at least three years. This assures independent software developers porting to the SCD that there will not be a sudden change in the functionality of SPARCware systems.

The SPARC Application Verifier (SAV):
As a part of the SPARCware program, all participating developers receive a software package called the SPARC Application Verifier (SAV). Independent software developer use SAV to verify that the software is SCD compliant.

In addition to testing for SCD conformance, SAV is intended to be used throughout development as part of the software vendor’s normal build and test environment. SAV was designed to fit easily into existing Quality Assurance processes with little additional effort. SAV can also be used as a development tool aiding the application writer in designing and producing portable and compatible code.

SAV uses both static and dynamic checks to determine if there are references to functions which are not contained in the SCD or supplied by the application. In addition, SAV checks all system functions called by the application invalid parameter inputs.

SAV runs on any SCD 2 compliant systems and consists of two parts: the Replacer and the Wrappers.

The Replacer statically verifies the file formats of the executable files of a program. While performing this static check, it examines all external references and all library references. The Replacer checks and verifies:

- That all shared objects needed by the application are either SCD-defined libraries or in the input file list.
- That all symbols needed by the application are either SCD-defined functions or defined in a file in the input file list.
- That symbols defined in the SCD are not provided by the application nor by the files in the input file list.
- That the application does not use privileged instructions and other instructions disallowed by the SCD.

The Replacer can be executed as a verifier at any time during the development process. The Replacer also prepares applications for the dynamic check by replacing calls to SCD-defined system interface functions with calls to Wrapper functions.

The Wrapper functions are shipped and installed as shared libraries which match the system libraries. Once system calls are replaced by their corresponding Wrapper functions, the dynamic test can be performed to verify application conformance.

Each Wrapper function verifies the arguments of its corresponding system function, records its findings (i.e., generates a status of error messages), then invokes the original system function. The Replacer and Wrappers output is a list of deviations or potential deviations from SCD 2.

Benefits for Independent Software Developers Developing Applications on Solaris 2.x:
In addition to all the benefits mentioned above, there is one more benefit to the independent software developer on Solaris 2.x. Since Solaris 2.x will be a SCD-compliant operating environment, if the developer passes SCD compliance through SAV, then the developer has bought “insurance” against future releases of the Solaris 2.x operating environment. “Insurance” ensures that binaries will not fail to execute on newer versions of Solaris 2.x. Insurance is based on the request of developers spending large portions of revenue and engineering cycles validating their products on permutations of complex matrices of hardware/ software releases. Therefore, SAV provides some assistance in validating an application’s correct use of the operating environment. Thus, if the application is correctly using the operating environment and the operating environment is correctly implemented, then the application should continue to run without the need for validating the application every time a new revision of the environment is released. This is one of the key visions behind Solaris 2.x.

For more information about SPARCware, contact:

SPARCware *Info*line:
1-800/453-6657

SPARC International:
1-415/321-8692 ext. 226