Sun Technical Bulletin

April 1993

Technical Information Services

Sun Microsystems Computer Corporation
A Sun Microsystems, Inc. Business

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No: 812-9304-01
Issue 1993 - 04

The *Sun Technical Bulletin* is distributed to customers with software/hardware or software-only support contracts. Send comments or corrections by electronic mail to *stb-editor@Sun.COM*, or by mail to:

Sun Technical Bulletin
Sun Microsystems Computer Corporation
2550 Garcia Avenue , M/S MPK02-04
Mountain View, CA 94043-1100 U.S.A.

Contents

STB Editor's Introduction 1

STB Overview

This month's STB contains articles of general interest in the topics outlined below.

Security

This month, we cover the newly released Solaris CMW 1.0 secure operating system for Compartmented Mode Workstations, with two lengthy reprints from the official documentation.

"Solaris CMW 1.0 Release Report" is an edited reprint of the offical technical white paper describing the CMW operating system in detail. The report describes how CMW 1.0 differs from the base Solaris 1.0; discusses some security concepts, such as the Trusted Computing Base, distributed systems, security evaluation and certification; defines and describes Trusted Facility Management, mandatory and discretionary access control, labels, etc.; and other features and benefits of the CMW product.

"Secure Software" is an edited reprint of an Appendix from the *Solaris CMW System Configuration Guide*. The appendix describes basic security vocabulary and history, including the origins of the Orange Book and the meanings of security levels as defined in the Orange Book, including B1 and CMW.

Network

This month's articles both address the difficult subject of converting a network's name service to NIS+. There is no clear-cut, definitive checklist of tasks to accomplish this transition, since the required steps are so dependent on site-specific details.

"NIS+: Planning for Setup" is a reprint of a chapter from the SunSoft Press book *All About Administering NIS+* by Rick Ramsey, available in retail bookstores worldwide. The chapter provides information to help you plan the setup and administration of your NIS+ service. It begins by recommending a setup procedure that works for most namespaces. Based on the steps of that procedure, it points out the decisions you need to make and the information you need to gather *before* you begin to set up NIS+. Details of the actual setup procedure will be found in the full book.

"NIS+ Transition Planning" is another in our occasional series of instructional articles assembled from Su n internal training materials. This article tackles the daunting task of describing everything a network administrator might have to do at any site to convert from NIS to NIS+. The reader is assumed to have a basic understanding of NIS and NIS+ concepts, and some familiarity with the NIS+ commands in the Solaris 2.*x* releases. Material in this article should enable the reader to understand the factors that influence the planning of conversions for network name service; follow a checklist to prepare a conversion plan; and prepare a Table Planning Worksheet to create site-specific name service maps.

Graphics

"XIL Imaging Library" describes SunSoft's new XIL product, the industry's first foundation imaging library with open software, hardware, and interface specifications. The library defines how imaging and video functions, such as display, manipulation, compression, and decompression, are carried out; it also specifies the interface between XIL and hardware. XIL strengthens SunSoft's strategy to encourage OEMs and thrird-party graphics vendors to develop through open hardware and software interfaces.

Software Release Versions

This section includes the current Sun software products and release version tables. We also include a revised updating of Sun's support matrix for Operating System release levels and platforms.

Notes & Comments

World Hotlines

For Sun customers world-wide served by your local service groups, use the customer service telephone numbers listed in this monthly item. Also, look to this section during the upcoming year for details on your local support call policies and procedures.

Reporting Bugs

The information in this monthly section continues to be expanded as Sun software service centers are added world-wide. The section includes a list of Sun service centers, addresses, e-mail hotlines, telephone hotlines, and FAX numbers.

Your STB Subscription

This notice is published monthly, giving customers useful information regarding ordering multiple STB hardcopy subscriptions, and duplicating additional STB copies from a single “master” copy. This duplication permission is limited, as detailed in the note.

We would like to encourage our readership to e-mail us any comments or suggestions regarding the content of the Sun Technical Bulletin, including requests for articles on specific subject matters. We are aiming at a broad spectrum of topics that are of interest to all sectors of our audience, both highly experienced technical people and relatively inexperienced users who are new to desktop workstations. We are not the appropriate channel for researching any and all technical problems that may come up in the field, but we will do our best to follow up on questions about articles that we have printed in the recent past.

Please let us know what you think, and we will do our best to accommodate you in future issues.

Thanks.

The STB Editor
sun!stb-editor or *stb-editor@Sun.COM*

Security 2

Solaris CMW 1.0 Release Report

This article is an edited reprint of the *Solaris CMW 1.0 Release Report* (part number FE369-0) from Sun Microsystems Federal, Inc. (a division of SMCC), describing their trusted version of the Solaris 1.0 operating system that meets the documented requirements for a Compartmented Mode Workstation.

Solaris CMW 1.0 was created to meet the needs of government and commercial users of Sun workstations and servers who need a more secure version of the operating system. Solaris CMW 1.0 is designed to keep users who are working at various security levels from accessing information they are not cleared to access. Solaris CMW 1.0 is also designed to keep the classification of the information being processed by the computing system from being accidentally compromised.

The end of this article includes an Evaluation Summary and chart from the *Solaris CMW Technical Brochure* (part number BE37-0). The following article is an Appendix from the *Sun CMW System Configuration Guide* (part number FE358-0), which describes the trusted computing system criteria and the evaluation process for trusted systems that have been developed by the federal government.

Introduction

Solaris CMW 1.0 is the trusted UNIX operating system developed by Sun Microsystems Federal, Inc. to provide security protections not available in standard, commercial versions of the UNIX operating system. Solaris CMW 1.0 is designed to be used in environments where classified material or other sensitive information must be protected. Users can run Solaris CMW 1.0 on a wide range of Sun SPARC hardware while sharing information, disk space, and server resources, and while continuing to run commercial databases and other popular applications. When configured and administered properly, Solaris CMW 1.0 is a trusted system that protects its own security mechanisms from being compromised and that protects the security of information being processed by its users.

This release report describes Solaris CMW 1.0 and how it differs from the base Solaris 1.0 (which includes SunOS 4.1.1 and OpenWindows 2.0).

Background Information

To understand the SunOS 4.1.1 operating system on which Solaris CMW 1.0 is based, you should be familiar with the features of SunOS 4.1 that are described in the *SunOS 4.1 Release Report* (part number FE257-0/20K). You can obtain the *SunOS 4.1 Release Report* and other supplemental documentation mentioned in this report from your Sun sales or marketing support representative.

The following documents provide more information about Solaris CMW 1.0:

- *Sun CMW System Configuration Guide,* #FE358-0
  This white paper includes information about the Sun hardware and software components required to configure a Solaris CMW 1.0 system. This guide also describes the memory requirements and supported software. Appendixes describe the trusted computing system criteria and the evaluation process that have been established by the federal government.
- *Introduction to Running Commercial Off-the-Shelf Applications on the UNIX-Based Compartmented Mode Workstation*, #FE359-0
  This technical white paper is a guide to installing commercial software applications and assessing their trustworthiness and their compatibility with Solaris CMW 1.0.
- *Solaris CMW Technical Brochure*, #BE37-0
  This brochure is an introduction to Solaris CMW 1.0 features for technical users, administrators, and security officers, who need to understand how these features protect information in a UNIX-based client/server computing environment.

Release Summary

Solaris CMW 1.0 has the following benefits:

- It enables users to work with information at various security levels while enforcing access controls, which protect information from users and programs that should not look at or modify it.
- It allows sharing of resources between clients and servers in a trusted distributed system.
- It provides compatibility with industry standards and government security standards.
- It creates a friendly environment for running many existing commercial applications.
- It permits trusted communications between Solaris CMW 1.0 machines and other machines.

What Is a Trusted System?

Government specifications talk about *trusted* systems rather than *secure* systems. No computing system is completely secure, because any system can be penetrated, given enough time. But systems can be trusted, some more than others. Government evaluation criteria have been developed based on the assumption that it is possible to measure this trust, and that it is possible to build, evaluate, and certify a computing system that conforms to a specific set of criteria and that therefore merits a certain security rating.

The National Computer Security Center (NCSC) and the Defense Intelligence Agency (DIA) define computer security standards and evaluate commercial products to assess their security rating. The role of the NCSC is described in the white paper *Sun CMW System Configuration Guide* along with the basic distinctions between the possible ratings (see the following article on "Secure Software"). The standard Solaris 1.0 provides security features that meet the criteria for a C2-rated system. Solaris CMW 1.0 provides additional security features that meet both B1 and CMW requirements.

Understanding the differences in Solaris CMW 1.0 that make it a trusted computing system requires understanding the following security concepts. The primary concept is that of the trusted computing base (TCB).

What Is a TCB?

The trusted computing base (TCB) is defined in government security criteria as being made up of any component of a computing system that could affect the computing system's security. The TCB includes these components:

- Hardware
- Software (the security-relevant portions)

  A software component must be part of the TCB whenever that component can influence the outcome of security decisions. Some system components, such as libraries and system calls, are in the TCB when they are invoked from a privileged process and are not in the TCB when invoked from unprivileged processes.
- Firmware
- Administrative databases, administrative programs, and the administrative procedures on which the other TCB components depend

TCB hardware is physically protected on a site-by-site basis by such means as keeping the hardware in locked rooms with access restricted to authorized personnel and by other site-specific policies and procedures. TCB software is protected by the Solaris CMW 1.0 system design and by administrative actions. The hardware protects the memory, and the software and administrative actions protect files.

Figure 2-1 shows the TCB from the point of view of the software. The figure shows how users interact with the kernel and through the kernel with the hardware by means of both untrusted and trusted programs. The two types of users shown in Figure 2-1 are *normal users* and *authorized users in roles*. Normal users are users who are not acting as administrators. Authorized users in roles are authorized users who have assumed one of the trusted facility management roles that replace the superuser in Solaris CMW 1.0 (see the section below on "Trusted Facility Management" for more about roles).

Untrusted programs are those that do not need to bypass Solaris CMW 1.0 security in order to run. *Trusted* programs are any programs that run with the use of one or more privileges. A *privilege* is a right granted to a TCB program to perform an action that the Solaris CMW 1.0 security policy would otherwise prohibit. (Privileges are more fully explained in the section below on "Principle of Least Privilege." See also Chapter 4 of the *SunOS CMW Trusted Facility Manual*, which is referred to as the TFM.)[1]

---

1. Solaris CMW 1.0 and SunOS CMW 1.0 are different names used for the same product.

The three kinds of trusted programs in the TCB are shown in Figure 2-1. *Forced privilege programs* are programs that run with privilege(s) no matter who invokes them. *Administrative program*s are commands or window tools that must always obtain privilege(s) from the administrative shell in order to function, and these programs are run by administrative roles. *Miscellaneous trusted Solaris CMW programs* are programs that are run both by normal users and by administrative roles. When a miscellaneous trusted program is run by a role, the program inherits privilege; when these types of programs are run by other users, the program runs without privilege.

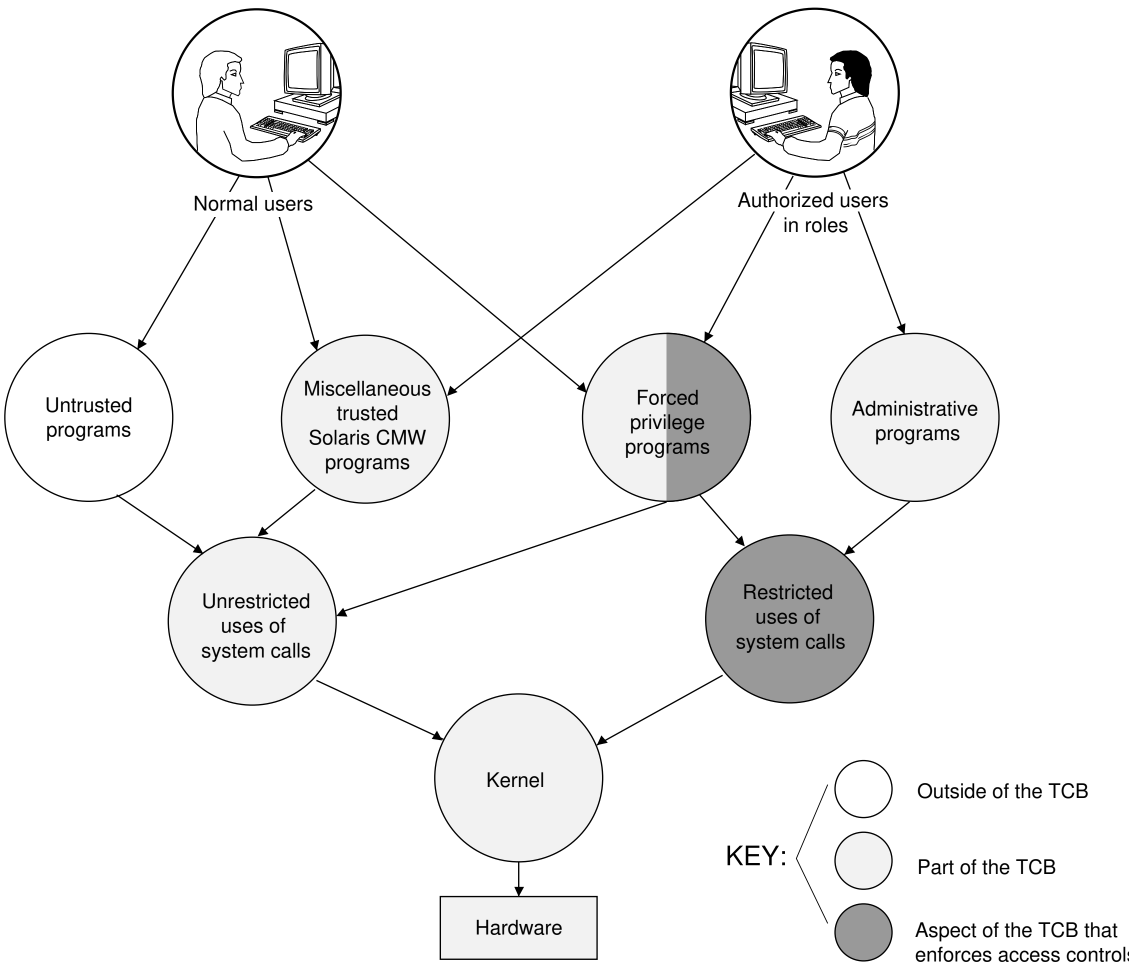


Figure 2-1 Elements within and outside of the trusted computing base

Customers may develop their own new programs or add commercial applications to the Solaris CMW 1.0 system. If they do not need privileges to run, programs or applications can easily be installed to run as untrusted programs outside of the TCB. If the programs require privileges to run, the programs can be added to the TCB as any of the three types of trusted programs. For more about the different kinds of programs in Solaris CMW 1.0, see Chapter 2 of the TFM.

Note – Even though the illustration shows many more items inside the TCB than are outside of it, in Solaris CMW 1.0 the greatest number of files and applications are untrusted and outside of the TCB.

What Is a Distributed System?

Sun is the first and currently the only vendor being evaluated by the NCSC at either B1 or CMW with a distributed work group model for a trusted computing base. Other vendors' B1 and CMW systems are being evaluated either as standalone systems or as a collection of individual systems that communicate by means of a separately evaluated networking applications, if they communicate at all. Solaris CMW 1.0 is currently being evaluated as a distributed computing system made up of the Solaris CMW 1.0 operating system running on one or more Sun SPARC workstations or servers that are administered as if they make up a single system.

When two or more Sun workstations are running Solaris CMW 1.0 and are connected to each other, all the machines must be administered centrally in order for trust to be maintained. Even though a set of workstations running Solaris CMW 1.0 and connected by an Ethernet cable appears to be a network of loosely coupled individual processors, when the appropriate security procedures are followed, the workstations become part of a single, virtual computing system. This virtual computing system is called the *distributed system*. From the user's point of view, he or she doesn't log into a single machine but logs into a system made up of multiple machines on which the user has equal access. From the administrator's point of view, each machine is configured not on its own but as part of a whole system with a unified set of user and machine configuration information that is centrally maintained using Sun's Network Information Service (NIS).

Figure 2-1 shows the concept of the distributed trusted computing base from the point of view of the hardware. All of the machines together are viewed as a single system, and all together make up the TCB.

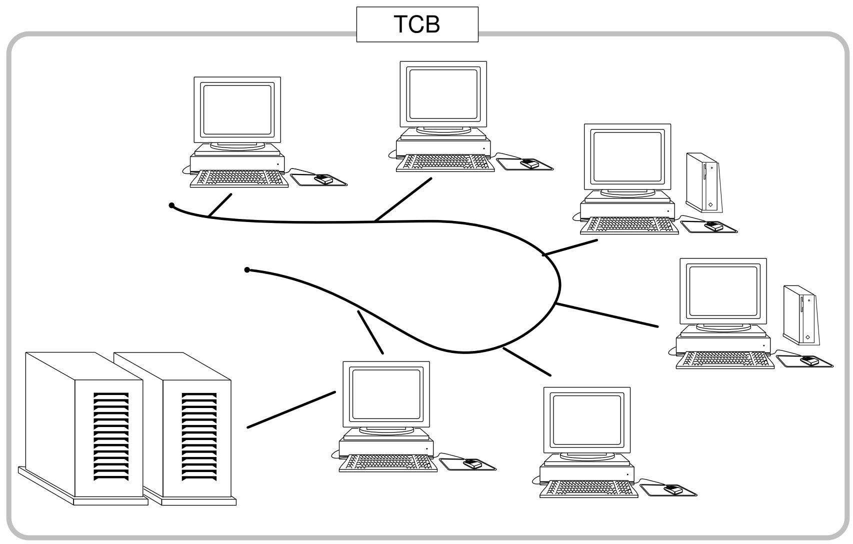


Figure 2-2 A distributed trusted computing base

NIS, Central Administration, and Accountability

NIS centrally manages user account information, machine name and address information, and other configuration information for all machines. Each distributed system is administered as a single NIS domain with a single NIS master server and one or more optional slave servers. Other configuration information is centrally administered by the information system security officer (ISSO) and the system administrator (see the section below on "Trusted Facility Management Implementation").

Centralized administration is essential for auditing, and auditing is essential for ensuring user accountability. Unless all filenames and users are uniquely identified in the distributed system by means of central administration of user accounts and mounts, the trusted administrator who administers auditing cannot analyze the audit data and trace suspicious actions back to individuals.

Evaluation

Solaris CMW 1.0 meets the evaluation requirements for a Compartmented Mode Workstation system as defined in the *Security Requirements for System High and Compartmented Mode Workstations*, DDS-2600-5502-87 (MTR 9992, Revision 1, November 1987) (CMWREQS). These requirements have been clarified in the *Compartmented Mode Workstation Evaluation Criteria*, Version 1, DDS-2600-6243-91 (MTR 10935), November 1991. Solaris CMW 1.0 also meets the requirements for a class B1 operating system as specified by the Defense Intelligence Agency (DIA) in DoD 5200.28-STD, *Trusted Computer System Evaluation Criteria* (TCSEC). The security levels and evaluation criteria are more fully described in Appendixes A and B in the *Sun CMW System Configuration Guide* (see the following article on "Secure Software"). Because it also meets some security requirements from classes B2, B3, and A1, Solaris CMW 1.0 can also be referred to as a B1+ product (see the Evaluation Summary chart at the end of this article).

Certification and Accreditation

Where evaluation assesses the trustworthiness of an abstract system, the *certification* and *accreditation* process assesses the trustworthiness of a specific configuration of hardware, software, and procedures. Installations of trusted computing systems in certain branches of the government may be certified and accredited by a government approving authority, called a *Designated Approving Authority* (DAA).

Certification is the technical information-gathering portion of the accreditation process, which establishes that a particular system design and implementation meet the site's mission requirements. Accreditation is the authorization by the DAA that the level of risk is sufficiently low to allow operation for a specified use.

Hardware Being Evaluated

The following hardware models are included in the Solaris CMW 1.0 evaluation:

- SPARCstation ELC (4/25)
- SPARCstation IPX (4/50)
- SPARCstation 1 (4/60)
- SPARCstation 1+ (4/65)
- TEMPEST SPARCstation 1+ (4/65T)
- SPARCstation 2 (4/75)
- TEMPEST SPARCstation 2 (4/75T)
- SPARCsystem 470 (4/470)
- SPARCsystem 490 (4/490)

Other Hardware That Runs Solaris CMW 1.0

Solaris CMW 1.0 also runs on the following Sun machines, which can be accredited:

- SPARCstation SLC (4/20)
- SPARCstation IPC (4/40)
- SPARCstation/SPARCserver 330
- Sun 4/110
- Sun 4/260
- Sun 4/280

Safely Extending the Configuration

Solaris CMW 1.0 provides tools and databases that make it possible for customers to add new trusted software or connect hardware that has not been evaluated and create an extended system that can still be trusted. Adding unevaluated software or connecting unevaluated hardware is referred to as creating an *extended configuration*. Accreditation can be given to both evaluated and extended systems.

Adding Unevaluated Machines and Connecting to Other Networks

The *SunOS CMW Trusted Facility Manual* (TFM) tells the administrator how to specify security attributes in the trusted networking databases for machines that are not running Solaris CMW 1.0, so that the specified security attributes will be applied to communications with these machines. These machines may be on the same Ethernet or connected through a gateway. Figure 2-3 shows an extended configuration with the TCB connected through a gateway to other machines on another network. The other computers may be running other vendor's CMW software, or they may be running a version of standard SunOS or another UNIX operating system that supports TCP/IP.

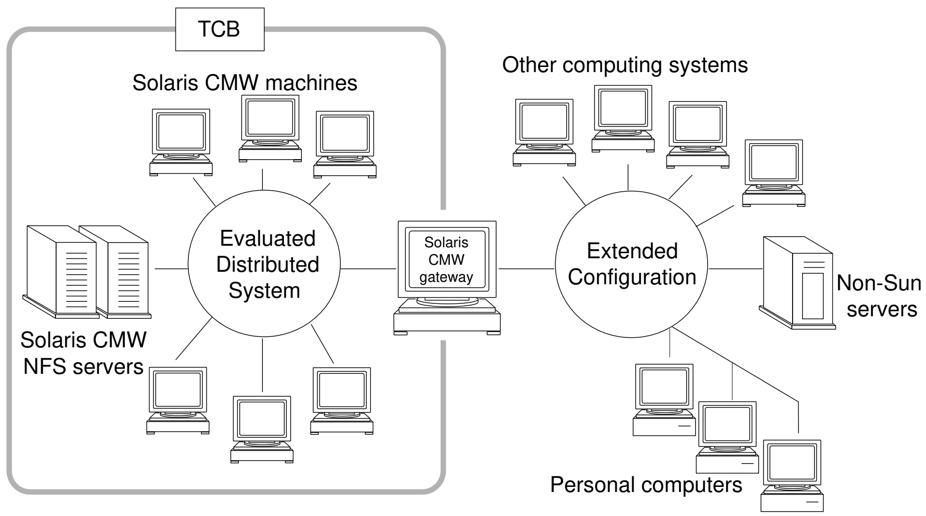


Figure 2-3 Extending the TCB

Adding Software

Solaris CMW 1.0 creates a friendly environment for running most existing commercial applications that don't need privileges in order to run. In addition, Solaris CMW 1.0 provides mechanisms for adding new trusted programs.

Adding applications to Solaris CMW 1.0 cannot be as straightforward as adding them to SunOS or other UNIX systems. However, unlike other trusted operating systems, Solaris CMW 1.0 allows a wide variety of applications to run without change, as long as the applications are well-behaved. (Well-behaved applications work within the Solaris CMW 1.0 security policy. See the white paper, *Introduction to Running Commercial Off-the-Shelf Applications on the UNIX-Based Compartmented Mode Workstation*, in which the criteria for well-behaved applications is described.)

Customers may develop new programs or add commercial applications that require privilege(s) as long as the administrator follows the procedures described in the TFM for assessing the trustworthiness of a new program. Before adding the software to the TCB, the trusted administrator must assess whether the program can use the privilege(s) in a trustworthy manner.

Trusted Facility Management

Trusted facility management is one of the assurance features required for trusted computing systems. An assurance feature provides the confidence that the security and accountability features have been implemented correctly, both during configuration and during operation of the system.

Trusted facility management is related to another security requirement called the principle of least privilege.

Principle of Least Privilege

The *principle of least privilege* requires that each subject be granted the most restricted set of privileges needed for the performance of the subject's authorized tasks. (Subjects are either users or processes; processes are programs executing on behalf of users.)

In thinking about how least privilege is applied in Solaris CMW 1.0, it is helpful to compare Solaris CMW 1.0 with traditional UNIX systems. Users of traditional UNIX systems, including SunOS, are familiar with the concept of the *superuser*. There is no superuser in Solaris CMW 1.0 and the powers associated with that user have been divided. Authorizations and privileges in the Solaris CMW 1.0 system are the means to distribute the power of the superuser. Each user and each program is given only the amount of power each needs, without giving away all power to one individual or one program.

For programs, the powers associated with the traditional superuser ID `0` have been divided into *privileges*. A privilege is a right granted to a TCB program to perform an action that the Solaris CMW 1.0 security policy would otherwise prohibit. For users, the tasks and capabilities of the traditional superuser are distributed among administrative roles and among certain other trusted users by means of *authorizations*. Authorizations can be thought of as electronic credentials. In order for a user to assume an administrative role or to perform an action that would be otherwise prohibited by the Solaris CMW 1.0 security policy, that user must have an authorization.

Trusted Facility Management Implementation

Solaris CMW 1.0 implements trusted facility management in these ways:

- Administrative duties are divided among three trusted facility management roles: information system security officer (ISSO), system administrator, and system operator. A fourth optional root role is available for installing certain applications.[2]

  The separation of duties among roles ensures that no one user has total control of the system. This division of tasks reduces the security risks of human error, deliberate wrongdoing, and system failure.

- Each administrative role requires that a user be given a specific authorization before that user can assume the role.
- The administrative commands and window tools that may be run by each role are strictly controlled. The privileges that a command or tool can use when running on behalf of a role are similarly restricted.
- Management of users and machines is under two-person control.

  Two-person control requires two administrators, the ISSO and the system administrator, to do certain tasks. The two roles work together to set up and unlock user accounts. They both configure and maintain the databases that identify machines and file systems on the distributed system and that regulate communications across the trusted network.

- Each command that has been determined to be security-sensitive may be executed by only one role.

  An example of a security-relevant command that may be run by only one role is `eeprom(8)`, which enables the ISSO role to perform such security-sensitive tasks as setting the PROM security mode.

- Roles are configurable.

  Sites can add new roles and reconfigure the existing roles, adding new administrative commands and window tools, and/or changing the privileges on administrative commands.

---

2. Do not confuse the Solaris CMW 1.0 root role with the UNIX superuser. The Solaris CMW 1.0 root role is used only by the ISSO and only when an application requires a real user ID of `0` in order for installation to succeed, and may run only a predefined set of commands.

Window Tools and Shelltools for Roles

Window tools and administrative command shells enable administrators to centrally control the working environment for all users, to configure all machines, to audit security-relevant actions, and to trace actions back to individual users — from any point in the system.

Administration is made easier because administrative window tools and shelltools can be invoked by an administrator not only on the machine where administrator is logged in, but also on the NIS master or on any other user's machine without having to do a remote login.

Trusted Facility Management Shell and Editors

Solaris CMW 1.0 provides a trusted facility management shell and trusted facility management editors for tasks that cannot be done using a point-and-click window tool.

The *trusted facility management shell* is a modified Bourne shell. The trusted facility management shell runs in a shelltool either at `SYSTEM_LOW`, which is the lowest label in the system, or at `SYSTEM_HIGH`, which is the highest label in the system (see the following section for a description of labels).

Each administrative role is assigned only the minimum number of commands that the role needs to do its assigned tasks. Likewise, the commands that roles use are assigned only the minimum number of privileges the commands need to do their work. Unlike regular shells, the trusted facility management shell has a built-in ability to make the specified privileges available to commands when the commands are executed by the various roles.

The *trusted facility management editors* are text editors with restrictions that force the user to stay within security policy. All of the standard Solaris 1.0 editors except for `ed(1)` have trusted facility management counterparts. For example, the `vi(1)` counterpart is `tfmvi(1T)`. The `tfmvi(1T)` command is the only editor configured for any role in the default Solaris CMW 1.0 system.

Security Concepts

The security concepts described in this section are necessary for understanding how Solaris CMW 1.0 protects information and protects itself.

Discretionary Access Control

Programs, data files, electronic mail, and devices are all treated as files in Solaris CMW 1.0, as in all other UNIX systems. Solaris CMW 1.0 uses discretionary access control (DAC), which is also a feature of other UNIX systems, to control access to files and directories based on the identity of the user and the group to which the user belongs (the ISSO creates site-specific groups, if desired, and the system administrator assigns users to groups).

Access to system files is partially controlled by DAC. Because trusted facility management roles work with an effective user ID of `0`, which is the traditional `root` user ID, roles have ownership of TCB files that are owned by `root`. As in other UNIX systems, users control the DAC permissions on any files that they own.

Labels

Labels represent levels of security. Labels are assigned to subjects and to objects. A *subject* is usually a process that is acting on behalf of a user. An *object* is an entity such as a file, directory, or device, that contains or receives information.

The labels used at a site are configured by the customer. Depending on site security requirements, the ISSO specifies the names of labels and the relationships between labels. The ISSO assigns clearances and minimum labels to users. A user can work and create new files at any valid label within the user's clearance and the user's minimum label.

Machines, printers, and devices may be given restricted label ranges by the ISSO, and then only users who are working within the restricted label range may obtain access.

Label Types

Two labels are associated with each subject and object in Solaris CMW: a *sensitivity label* (SL) and an *information label* (IL). The SL and the IL together make up a CMW label. A CMW label has this form when viewed by users:

```
INFORMATION LABEL[SENSITIVITY LABEL]
```

Figure 2-4 The syntax of a CMW label

Sensitivity Labels

Sensitivity labels (SLs) are an NCSC requirement for B1 systems. Solaris CMW 1.0 compares SLs when deciding which subjects can access which objects.

Information Labels

The requirement for information labels (ILs) was introduced by the DIA for CMW systems. ILs classify the actual sensitivity of the information itself, rather than the sensitivity of the object that contains the information. Words in ILs are also used to indicate how the information should be handled.

ILs can be very useful for customers that want to control how information is distributed because the IL may contain code words and warnings on how the information should be handled. Here are some examples:

- Every e-mail message sent from a Solaris CMW 1.0 machine is automatically labeled with the IL that applies to the message.

  Figure 2-5 shows an IL of `CONFIDENTIAL D/E ALL EYES`.

- Whenever an authorized user exports information to a tape or floppy, the software supplies an IL that the user is prompted to write on a physical label. The user is required to put the IL along with the SL of the data on the archive label. The IL provides a guide to the real sensitivity of the information, which may be used in deciding whether to manually downgrade the SL.

  Figure 2-6 shows the user being prompted to write the IL of `CONFIDENTIAL D/E ALL EYES` on a label on an archive tape.

- The IL can help users decide whether to downgrade the SL of a file (users need authorizations to upgrade or downgrade an SL).

  A user might choose to downgrade the SL of a file if the IL (which reflects the actual security label of the information in the file) is below the SL of the file. In Figure 2-7, a file called `1.14.93` has an IL of `UNCLASSIFIED` lower than its SL of `CONFIDENTIAL`, which is shown as `[C]`.

- The IL is displayed on banner and trailer pages for every print job.

  Figure 2-8 shows an IL of `CONFIDENTIAL PROJECT X LIMDIS` on a printer banner page.

```
From: roseanne@terse (Roseanne Sullivan—Manual Laborer)
To: griffon_notes@griffon
Subject: IL floating on label translations.
X-Sender-Information-Label: CONFIDENTIAL D/E ALL EYES
Content-Length: 1513
X-Lines: 23
```

Figure 2-5 The information label as part of a mail header

```
prolix% deallocate st0
Please remove the tape from the st0

Please label the media removed from st0 with information label
CONFIDENTIAL D/E ALL EYES
prolix%
```

Figure 2-6 The information label supplied for physically labeling exported information

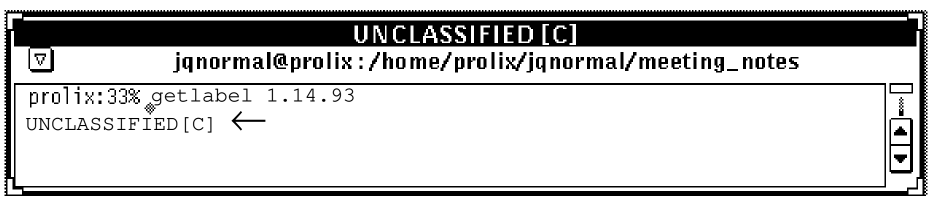


Figure 2-7 A file’s information label lower than its sensitivity label

SECRET

This output must be protected as:

SECRET AB

unless manually reviewed and downgraded.

The system has labeled this data:

CONFIDENTIAL PROJECT X LIMDIS ←

Sun OS CMW 1.0
NeWSprint Interpreter version 2.1

jqpublic@prolix
stdin

Tue Feb 11 18:28L25 1992

Printer queue: npcmw

HANDLE VIA (CH B)/(CH A) CHANNELS JOINTLY

BANNER
3-1646730499

SECRET

Figure 2-8 Information label on the printer banner page

Information Label Floating

The SL is a fixed label, while the IL is a label that floats to dynamically show how the security level of information changes as information from various sources is combined. For example, if a user is working at a high SL such as `TOP SECRET` and opens a new file for editing, the IL of the process starts at the lowest label in the system. If the user reads a file with an IL of `TOP SECRET` into the file, the IL floats up to `TOP SECRET`. When the user saves the file, the file will have both an SL and an IL of `TOP SECRET`.

Label Components

An SL consists of a *classification* and a set of *compartments*. An IL consists of a classification and a set of compartments, plus a set of *markings*.

Classifications

Classifications have a hierarchical order and represent a level of sensitivity. For example, if `TOP SECRET`, `SECRET`, and `UNCLASSIFIED` are valid classifications at a site, `TOP SECRET` is more sensitive than `SECRET`, and `SECRET` is more sensitive than `UNCLASSIFIED`. A label contains only one classification. In a Solaris CMW 1.0 system, a maximum of 256 classifications may be defined.

Compartments

Compartments represent topics or work groups. Each compartment has a name such as `NATO`, `NUCLEAR`, or `CRYPTO`. Compartments have no intrinsic ordering, but the ISSO can impose constraints on which compartments and classifications can be combined. For example, the use of the compartment `EXPORTS` might cancel the use of the compartment `IMPORTS`, while the combination of two or more compartments of `GUATEMALA`, `HONDURAS`, `NICARAGUA`, or `PANAMA` might be replaced with the compartment `CENTRAL AMERICA`. Labels may have any number of compartments up to the total of 128 compartments that may be defined.

Markings

Markings are the code words, dissemination and control markings, and handling caveats that apply to the labeled information. Markings have names such as `NOFORN` (No Foreign Distribution) or `LIMDIS` (Limited Distribution) or

EYES ONLY. As with compartments, the ISSO can impose constraints on which markings and classifications can be combined. Labels may have any number of markings up to the total of 128 markings that may be defined.

Figure 2-9 shows the components of a CMW label.

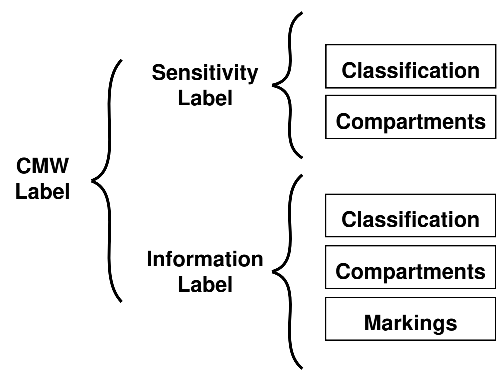


Figure 2-9 CMW label components

In the CMW label in Figure 2-10, the IL (on the left) has a lower classification than the SL (on the right in brackets). Besides the code word NUCLEAR that both the IL and SL have in common, the IL has NOFORN and EYES ONLY markings.

```
SECRET NUCLEAR NOFORN EYES ONLY [TOP SECRET NUCLEAR]
```

Figure 2-10 Example CMW label

Mandatory Access Control

Mandatory access control (MAC) is used to control access based on sensitivity labels (SLs). Whether a process acting on behalf of a normal user gains access to a file or, if authorized, to a tape drive or other media device, depends on the relationship between the SL of the subject and the SL of the object. MAC rules are explained more fully in Chapter 2, “System Security Overview,” in the *SunOS CMW Security Features User’s Guide*. The basic rule is that a subject’s SL must be higher than or equal to the SL of the object in order for the subject to access the object.

An SL is said to *dominate* another SL if the first SL is higher than or equal to the second SL. The SL of the subject *SL[S]* dominates the SL of the object *SL[O]* if these conditions are all true:

- The classification of *SL[S]* is greater than or equal to the classification of *SL[O].*
- The set of compartments in *SL[S]* includes all the compartments in *SL[O].*

Therefore, if a user is trying to access a file, the SL of the user's process must dominate the SL of the file.

Figure 2-11 gives some examples. The two SLs `TOP SECRET ALPHA2 BETA2` at the top dominate each other because they are equal. `TOP SECRET ALPHA2 BETA2` also dominates both the SLs `TOP SECRET ALPHA2` and `SECRET BETA2` at the bottom because the classification `TOP SECRET` dominates `TOP SECRET` and `SECRET` and the compartments `ALPHA2` and `BETA2` in the dominant label include the compartments in both the dominated labels. `TOP SECRET ALPHA2` in the lower left does not dominate `SECRET BETA2` on the right even though the first label's classification is greater because the first label does not contain the compartment `BETA2`.

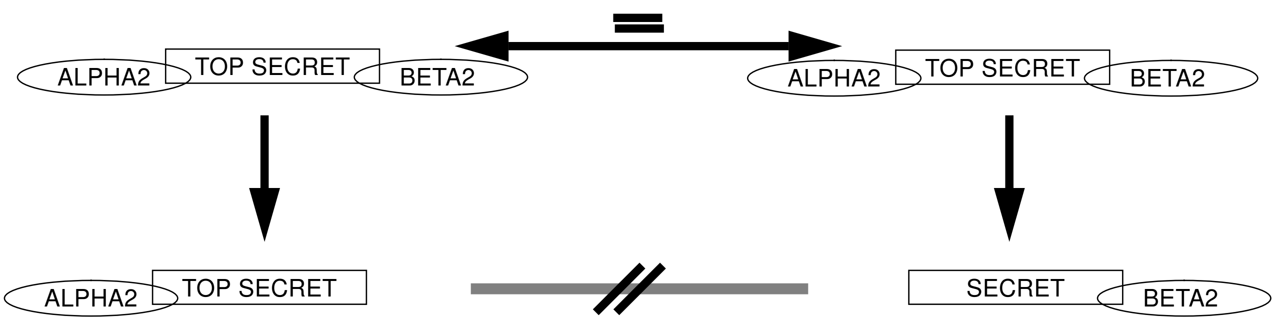


Figure 2-11 Label relationships

Features and Benefits

This section describes the main features and benefits of Solaris CMW 1.0.

Standards Compatibility

Solaris CMW 1.0 provides compatibility with government standards and other industry standards.

As stated earlier in this report, Solaris CMW 1.0 is being evaluated as a compartmented mode workstation (CMW) system and as a B1 system in a combined evaluation by the NCSC and DIA.

Solaris CMW 1.0 is compatible with industry standards to the same degree as the base system Solaris 1.0, with some modifications that enable Solaris CMW 1.0 to provide the necessary security protections. Solaris 1.0 meets the industry and government standards listed below. As far it is possible to determine without conformance testing, Solaris CMW 1.0 also conforms to these standards:

- X11 R4 Window System
- IEEE Standard 1003.1-1988 (POSIX)
- FIPS 151-1
- X/OPEN XPG2
- System V Interface Definition (SVID) Issue 2 (Vol. 1, 2, & 3)
- Application Binary Interface compatibility with SPARC Compliance Definition 1.0
- Various networking requests for comment (RFCs)

Like Solaris 1.0, Solaris CMW 1.0 provides SVR4 interfaces. Applications written in Solaris CMW 1.0 will be upward source-compatible with the Solaris 2.*x*-based future releases that are planned for Solaris CMW 1.0.

Trusted OpenWindows

Solaris CMW 1.0 allows users to work with information at various levels of sensitivity while the window system keeps the classification of the information from accidental compromise. The window system is compatible with the external interface of Solaris 1.0. Although what the user sees is somewhat different from the standard OpenWindows desktop because of the new labeling and trusted path features, the rules for using the modified window system layout are easy to learn.

The window system enforces MAC and DAC on windows just as the file system enforces MAC and DAC on files. Windows belonging to trusted applications cannot be interfered with by untrusted applications. The window system has been modified to provide a reliable means for a user to be sure that any password and other sensitive information he or she enters is being collected only by trusted programs.

Trusted OpenWindows is based on OpenWindows 2.0 and is a trusted implementation of the X Window System. Trusted OpenWindows lets users cut-and-paste and drag-and-drop between windows, while a selection confirmer automatically prevents unauthorized transfers of higher-level information into windows at a lower security level and of information at a lower security level into windows at a higher level. Selections may be relabeled during drag-and-drop by users who are properly authorized.

Users cannot bypass the window system. The only way into the Solaris CMW 1.0 system for both normal users and administrators is by logging in through the trusted path using the login tool. Only a trusted administrator who has the protected PROM password can bypass the window system, and only in emergency maintenance mode.

Trusted Path

All security-critical operations, such as login, entry of sensitivity labels, and change of user passwords, are performed through the window system's trusted path. The trusted path cannot be forged or tampered with by untrusted programs, to eliminate the risk that a user will provide sensitive information to a trojan horse program.[3]

3. A trojan horse is a program that is designed to maliciously spoof the user into providing information that may be used later in an attempt to penetrate the system.

Whenever the machine is in multiuser mode, if no one is logged in, the login tool always displays with an unspoofable background, as shown in Figure 2-12.

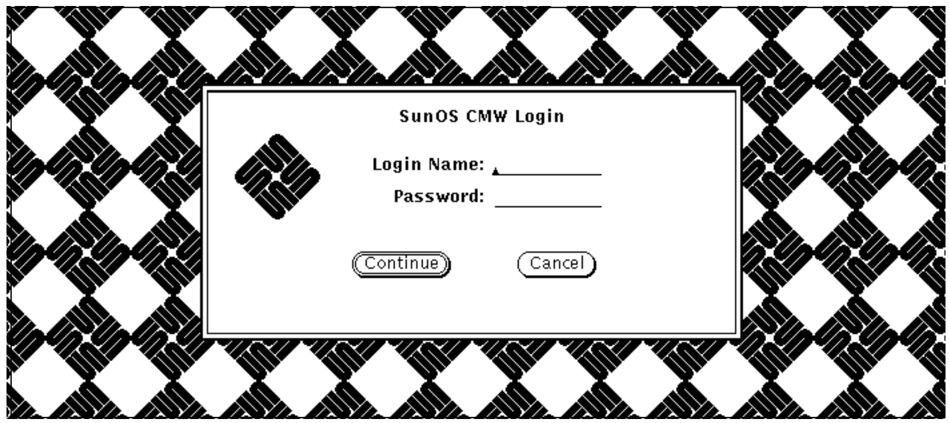


Figure 2-12 Logintool with unspoofable background

Users' first contact with the trusted path occurs when they use the login tool to identify and authenticate themselves with a password. After login, users are brought directly into the window system; a trusted screenstripe displays at the bottom of the screen.

Figure 2-13 shows the indicators in the trusted screenstripe during normal operation. Figure 2-14 shows the screenstripe when a trusted application is awaiting authentication from the user.

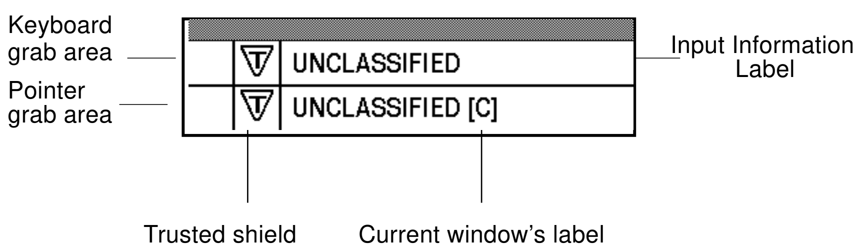


Figure 2-13 Screenstripe during normal operations

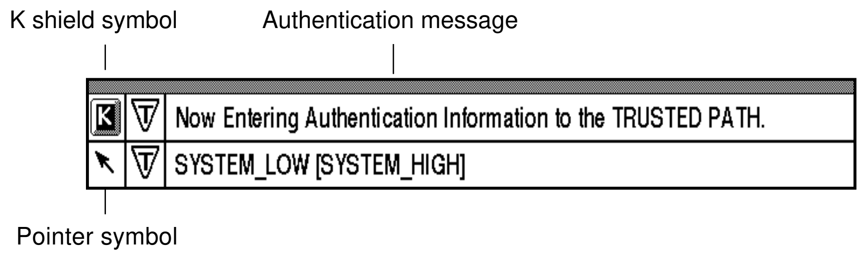


Figure 2-14 Screenstripe with authentication message and grab symbols

Users learn that the screenstripe is always displayed during a session. Users also learn that whenever they enter a username and password to either change their password or assume a role, if certain indicators do not appear, the user should not trust the authentication program.

Normal users do most of their work using standard command tools, shelltools, and other DeskSet tools, such as mailtool, calendar manager, and printtool. When normal users need to perform security-critical functions, such as changing the labels on files or changing their passwords, the users access the trusted path menu through the screenstripe. A user who is authorized to assume a trusted facility management role assumes the role through the trusted path menu.

Labeled Windows

Labeled windows identify the security level of the window itself and of the information contained in the window. As shown in Figure 2-15, the CMW label for each window is displayed in a frame at the top of the window. The background of the frame displays on color monitors in a color that is coded to match the label. The sensitivity label portion of the displayed CMW label is equal to the label of the process that created the window. The information label portion of the displayed CMW label is initialized to the lowest label in the system; and as work is performed in the window, the IL floats up to reflect the IL of any data that is displayed in the window.

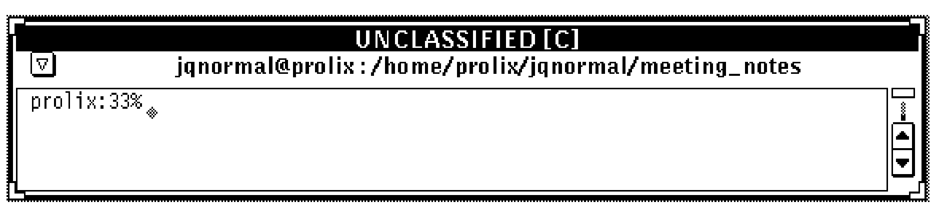


Figure 2-15 CMW label in the window's trusted frame

Window Applications Compatibility

Well-behaved applications run within the trusted window system without recompiling or relinking.[4] These include applications built with two standard OPEN LOOK toolkits — XView and OLIT — or with other toolkits that comply with the Inter-Client Communications Conventions Manual (ICCCM), including Motif applications.

Trusted NeWSprint

For printing, Sun's NeWSprint 1.0 product has been extended to meet CMW requirements for labeled laser printer output. Trusted NeWSprint 1.0 is an unbundled product that is needed for meeting CMW printing requirements. Trusted NeWSprint puts labels at the top and bottom of each page of printer output and puts labels and handling caveats on banner and trailer pages. These labels cannot be interfered with or spoofed.

The ISSO may set a restricted label range for a printer so that access to the printer will be denied to print jobs whose label is not within the printer label range.

Trusted Distributed Computing

Solaris CMW 1.0 provides trusted means for sharing information and for sharing resources among users on any number of client and server machines in a distributed computing system. Static routing is supported.

Trusted Networking Software

MaxSix is a set of networking protocols developed by SecureWare, Inc. to provide interoperability between CMW vendors. Trusted networking software that conforms to MaxSix 1.1 has been integrated into Solaris CMW 1.0, and has been extended to work in the Sun client/server environment. The trusted networking software passes user information and security attributes as IP options along with all communications between machines. As mentioned earlier in this report, in an extended configuration, the machines may be running other vendor's CMW software or other operating systems that support TCP/IP. Applications that use sockets will run without change, subject to MAC restrictions.

---

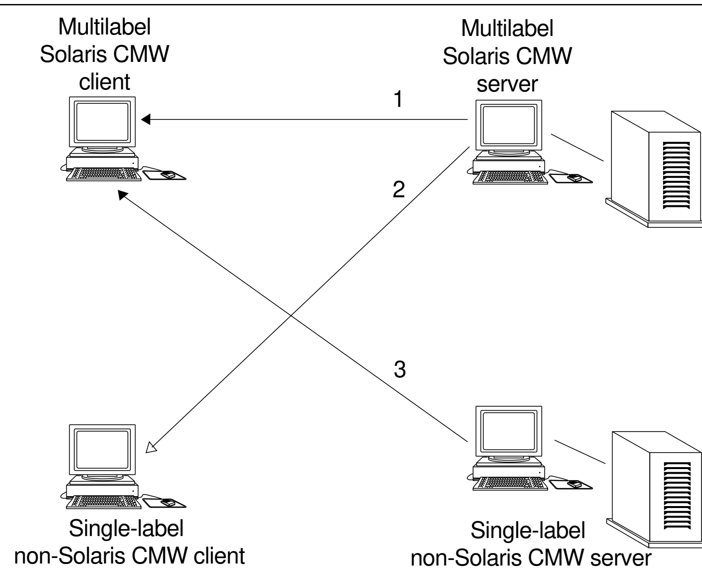
4. See earlier section on "Adding Software" for details.

Trusted NFS

Machines in the Solaris CMW 1.0 system can mount file systems from the following types of machines:

- Other multilabeled Solaris CMW 1.0 machines
- Unlabeled file systems from other machines running SunOS 4.*x*
- Unlabeled file systems from machines running other operating systems that support NFS

When setting up a mount for unlabeled file systems, the ISSO assigns the file system a single label as a mount label, and the mount label is applied to all files in the file system. Trusted NFS supports mounts between Solaris CMW 1.0 and non-Solaris CMW 1.0 servers and clients.[5] Figure 2-16 shows all the possible client/server relationships.



Key
1. Multilabel Solaris CMW 1.0 server exporting file systems to a Solaris CMW 1.0 client.
2. Multilabel Solaris CMW 1.0 server exporting file systems to a single-label non-Solaris CMW 1.0 client.
3. Single-label non-Solaris CMW 1.0 server exporting file systems to a multilabel Solaris CMW 1.0 client.

Figure 2-16 Possible client/server relationships

5. Non-Solaris CMW servers and clients may either be running a version of SunOS or another vendor's operating system that supports NFS.

Trusted NFS is based on NFS V2, Sun's distributed file-sharing product. Trusted NFS supports the multiple labels that Solaris CMW 1.0 applies to file system objects. Trusted NFS uses the trusted networking software to pass security attributes that the NFS server can then use as a basis for its access decisions. New file system types have been developed for locally mounted and remotely mounted labeled file systems, and new mount options help enforce security policy on any type of file system, whether labeled or unlabeled.

Multilevel Directories

Solaris CMW 1.0 provides multilevel directories (MLDs) to allow applications or users to put files at different labels into the same directory. For example, many UNIX applications keep temporary working files in the `/tmp` directory. When an ordinary directory has a particular sensitivity label, then unprivileged applications running at other sensitivity labels cannot create files in that directory because of MAC rules. With MLDs, the applications that use `/tmp` in other UNIX systems can continue to use `/tmp` in Solaris CMW 1.0.

An MLD is a special type of directory that can contain only special subdirectories called single-level directories (SLDs). Each SLD corresponds to an individual sensitivity label, and the name of the SLD is derived from the sensitivity label. When an application creates a file in an MLD, the file is actually created in the SLD that matches the application's sensitivity label.

With MLDs, applications that need to store files at multiple labels in the same directory can continue to work as before without reprogramming the applications or giving them privileges.

Identification, Authentication, and Accountability

As mentioned earlier in this report, in standard UNIX operating systems, knowledgeable intruders may be able to look at, copy, and manipulate information they should not be able to access, and to do these things without anyone knowing what they have done. Solaris CMW 1.0 makes all users individually accountable for their actions. The first feature that supports *accountability* is called *identification and authentication*. Login restrictions require all users to log in through the window system, and all users, even administrators, must log in and identify themselves with a username and authenticate themselves with a password. Identification and authentication also are done whenever a user assumes a role or resumes a session after the screen is locked.

Accountability is made possible by an audit ID assigned to each user at login. The audit ID is maintained even when the user's ID changes, as it does when a user assumes a role. The audit ID makes it possible to trace all user-level actions back to a specific user.

Centralized administration is the other feature that supports accountability. Because all machines are configured and maintained together like a single system, each user has the same identity on all machines, and administrators are able to identify the user and enforce accountability for user actions whenever they occur in the system.

Auditing

All security-relevant actions may be audited. The ISSO uses audit flags in audit control files to specify the types of actions to audit globally for all users. The ISSO may also fine-tune auditing for individual users by specifying audit flags in a password adjunct file, auditing some users more and some users less. The ISSO also configures the location of all the audit file systems and established procedures for archiving and interpreting audit data.

After audit data is collected by the auditing mechanisms in the centrally administered audit trail, auditing commands allow the ISSO to combine, reduce, process, and display audit trail files.

Importing and Exporting Data

Importing information into or exporting information out of the Solaris CMW 1.0 system relies on device allocation and the modified `tar(1)` command. When a device is allocated to a user, no other user can access the information on that device. Normal users who have the authorization that is needed to allocate a media device can use the `tar` command to export or import information at a single label. Administrative roles can use the `tar` command with two restricted options that save file labels, privileges, and MLD pathname information. The modified `tar` command allows roles to archive and restore files while preserving Solaris CMW 1.0 security attributes.

Integrity Maintenance and Consistency Checking

Integrity maintenance and consistency checking are performed during the trusted recovery process after a reboot and on demand by the ISSO.

The purpose of *trusted recovery* is to ensure that the TCB is not corrupted after a machine failure. During boot, the consistency of data structures in labeled file systems is verified automatically. The attributes of files and directories in the TCB are also checked automatically to make sure that the objects have not been modified by unauthorized users. At other times, the ISSO can also manually do these checks:

- Check the validity of all file labels in the file system.
- Check for the existence of unauthorized forced privilege programs.
- Check for the existence of devices where they are not supposed to be.

Installation and Configuration Differences

The `suninstall(8)` utility has been modified to install the Solaris CMW 1.0 software with minimal changes to the user interface. Quick Install is not supported, and installation must be done from a directly connected CD device.

Installing Solaris CMW 1.0 is very different from installing Solaris 1.0 or any other UNIX operating system. The differences are mostly due to the need for centralized administration, strict access controls, and user accountability, which are all part of creating and maintaining a trusted distributed computing system. Other UNIX systems do not require the strict controls that a trusted system needs.

Installation is performed by an install team made up of the ISSO and the system administrator. Working together, they first plan the configuration of the system, and then install and configure the machines in a defined sequence, starting with the NIS master. The first sections of Part 3, "Installation and Configuration," in the TFM describe the planning and other preparatory work that must be done and the knowledge that the install team must have before taking the shrink wrap off the Solaris CMW 1.0 CD.

The boot PROM on a properly configured Solaris CMW 1.0 machine requires a password, which should be known only by the ISSO. The proper use of the PROM password prevents anyone except the ISSO from escaping to the PROM monitor and booting a different kernel, altering or examining memory, or other security-sensitive operations.

After `suninstall` runs on each machine, mandatory configuration steps must be performed to tie each machine into the distributed system.

Other Major Differences

The following table shows a side-by-side comparison of the other major differences between Solaris CMW 1.0 and other UNIX systems.

Table 2-1 Differences between Solaris CMW 1.0 and Other UNIX Systems *(1 of 2)*

Solaris 1.0 and Other UNIX	**Solaris CMW 1.0**
Traditional superuser (`root` user) has all power and access to every aspect of the system	The power of the superuser is broken into authorizations given to users and privileges given to programs. Programs that need to bypass security restrictions in order to work execute with only the limited set of privileges they need. One administrator no longer has all the power in the system because administrative tasks are divided among multiple trusted facility management roles: information system security officer (ISSO), system administrator, and system operator. Each administrative role is given only the commands needed to do the restricted set of tasks assigned to that role, and only an authorized user can assume a role.
One person controls user accounts and configures machines and mounts.	Administering users and machines is under two-person control. The system administrator maintains much of the same types of information that a standard UNIX system administrator maintains, setting up basic user account information, adding and deleting hostnames and Internet addresses for machines, and setting up file system mount table entries for mounts. The ISSO specifies users' security attributes, unlocks the user accounts, sets up auditing, establishes the security attributes that apply to communications between machines and networks, specifies the mount label for mounts, and protects the PROM password.
Anyone who gets the `root` password can log in and do anything anonymously.	An administrator must first log in as a normal user before assuming a role through the trusted path. Administrative actions are auditable and can be traced back to the user who assumed the role. Only the ISSO should know the PROM password, and the ISSO is the only user who should use the PROM password to bring a machine into single-user mode when emergency maintenance must be performed.
Anyone can install the software and configure one machine as an individual system.	An install team made up of the ISSO and the system administrator configures all machines together as a single system with one NIS master.
Users can have one user name and user ID on one machine and another user name and ID on another machine.	Each user must have a unique identity in the distributed system. Centralized administration of user account information and auditing makes it possible to trace security violations back to individual users.

Table 2-1 Differences between Solaris CMW 1.0 and Other UNIX Systems *(2 of 2)*

Solaris 1.0 and Other UNIX	Solaris CMW 1.0
User accounts are set up by the system administrator editing UNIX files.	The system administrator role uses one administrative window tool to set up the user account, and the ISSO role uses another window tool to specify the user's clearance, minimum label, and other security attributes, and to unlock the account.
Users can log in directly and bypass the window system.	Users log in through the logintool and, after setting the session clearance, do all their work in the window system. The ISSO bypasses window system only in maintenance mode, after using the PROM password to boot the machine in single-user mode.
If one of the optional PROM security modes is not set, anyone can bring the machine to monitor mode, and then do a number of things that can violate system security, such as boot from a nonstandard kernel or boot the machine in single-user mode and have complete access to all databases and program files.	PROM security mode is set as a required configuration step and the PROM password must be protected by the ISSO so normal users cannot reboot in single-user mode or boot from a nonstandard kernel. If full security mode is set, users cannot reboot at all and have to call the ISSO.
Discretionary access control (DAC) is used to restrict users' access to directories, programs, and data files based on the user's identity and group membership.	Users access directories, programs, and data files based both on DAC and on mandatory access control (MAC). MAC controls access by comparing the labels at which the user's programs are running to the labels of system objects (directories, data files, programs, or devices) that the user wishes to access. Users authorized to assume roles can use certain administrative programs that normal users cannot access or execute certain generally available programs with privileges that the program does not have when run on behalf of a normal user. An administrative program is assigned to be used by one or more roles with privileges that depend on what the role needs to do with the program. When the program is executed by a role, the necessary privileges are passed to the administrative program either automatically through a trusted facility management window tool or on the command line in a trusted facility management shell.
Integrity of system programs and files is not checked.	Integrity-checking mechanisms exist to ensure that labels are valid and that attributes of files and directories have not been changed by unauthorized users. Administrators must synchronize consistency tables after modifying configuration files. Verification is performed automatically during reboot. If discrepancies are found, administrative action must be taken to restore integrity before the reboot will succeed.
After a reboot, anyone can log in.	After a reboot, an authorized user (usually security officer) must enable logins before anyone else can log in.

Functionality Not Supported

Besides the differences noted in the previous two sections, Solaris CMW 1.0 is assumed to be generally compatible with Solaris 1.0 (except for the modifications necessary for security reasons). The following functions and hardware are not supported in Solaris CMW 1.0:

- Automatic mounting of file systems with the `automount(8)` command
- Remote File System (RFS)
- Translucent File System (TFS)
- Transport Layer Interface (TLI)
- Accounting
- Quotas
- Dynamic routing

  The `in.routed` daemon is not supported. [Static routing is supported with the `route`(8C) command.]
- Direct Graphic Access (DGA)
- NeWS-based toolkits
- SunView
- Loadable drivers
- Secure RPC
- Domain Name Service (DNS)
- NFS mounting with the secure option
- NFS file locking
- Quick Install
- Remote installation
- Some unbundled applications that are not well-behaved

  (See the *Introduction to Running Commercial Off-the-Shelf Applications on the UNIX-Based Compartmented Mode Workstation* white paper for what makes a well-behaved application.)
- Sun-2, Sun-3, and Sun-386i hardware

Evaluation Summary

Government security requirements address two factors: features and assurances. Features are simply the functions that impact the end user. Assurances are the components that provide confidence that the security features and architecture of the system accurately mediate and enforce the security policy.

Solaris CMW 1.0 merges the security features and assurances described in the two leading U.S. Government requirements documents for secure computing: the National Computer Security Center's *Department of Defense Trusted Computer System Evaluation Criteria* (DoD 5200.28-STD), commonly known as the Orange Book; and the Defense Intelligence Agency's *Security Requirements for System High and Compartmented Mode Workstations* (DDS-2600-5502-87).

Figure 2-17 shows the features and assurances that Solaris CMW 1.0 provides to meet both the B1 and CMW requirements and beyond. In the illustration, the columned boxes indicate features and assurances that are included in Solaris CMW 1.0; the vertical columns indicate which level of Orange Book (TCSEC) requirements are met by the product. The column labeled **CMW** shows CMW requirements (from the CMWREQS) that are distinct from the **C**-**B**-**A** levels of Orange Book specifications. Note that many of the features that show **A1** compliance really constitute no additional requirements beyond those in a lower level.

For further discussion of the levels of evaluation, see the following article on "Secure Software."

	C1	C2	B1	CMW	B2	B3	A1
SECURITY POLICY							
Discretionary Access Control							
Object Reuse							
Labels							
Label Integrity							
Exportation of Labeled Information							
Exportation to Multilevel Devices							
Exportation to Single-Level Devices							
Labeling Human-Readable Output							
Information Labels *							
Mandatory Access Control							
Subject Sensitivity Labels							
Device Labels							
Labeled Windows *							
ACCOUNTABILITY							
Identification and Authentication							
Audit							
Trusted Path							
ASSURANCE							
System Architecture							
System Integrity							
Security Testing							
Design Specification and Verification							
Covert Channel Analysis							
Trusted Facility Management							
Configuration Management							
Trusted Recovery							
Trusted Distribution							
DOCUMENTATION							
Security Features User's Guide							
Trusted Facility Manual							
Test Documentation							
Design Documentation							

* Unique CMW Requirement

Figure 2-17 Trusted Computer System Evaluation Summary — Solaris CMW 1.0

Secure Software

This article is an edited reprint of Appendix A, "Secure Software," of the *Sun CMW System Configuration Guide* (part number FE358-0).[6]

Trusted Computing Overview

Computer security is a hot issue today, but it's an issue that's been simmering for many years. The development of government security regulations and standards, research into security mechanisms, and debates over the threats to information and the cost of protecting against these threats — all of these activities are well into their third decade. Computer security itself isn't new. What's new is security's broader focus (now, security means more than just keeping outsiders out) and its wider appeal (now, security in important to business as well as government).

History

Throughout the decade of the 1970s, there was an ever-increasing awareness by the government of a need for computer security. In the early 1980s, there were a number of government agencies studying, and writing standards for, computer security. By the mid-1980s, responsibility for computer security within the federal government was centralized in the National Computer Security Center (NCSC).

Birth of the Orange Book

In August of 1983, the NCSC released the Department of Defense *Trusted Computer System Evaluation Criteria* (TCSEC), which is commonly known as the "Orange Book" because of the color of its cover. The Orange Book is the bible of secure system development. It describes the evaluation criteria used to assess the level of trust that can be placed in a particular computer system. It effectively makes security a measurable commodity so a buyer can identify the exact level of security required for a particular system, application, or environment.

6. Much of the information in this article was obtained directly from Deborah Russell and G.T. Gangemi Sr., *Computer Security Basics*, O'Reilly & Associates, Inc., Sebastopol, CA, 1991. See this book for more detailed basic information on security.

The Orange Book presents a graded classification of secure systems. It defines four broad hierarchical divisions, or levels, of protection: D, C, B, and A, in order of increasing security. Within each division, the Orange Book defines one or more classes, each defined by a specific set of criteria that a system must meet to achieve a rating in that class. Some divisions have only a single class, others two or three. The original Orange Book was revised slightly and was reissued in December of 1985.

Security Terms

In the security arena, there are very precise meanings for terms. Unfortunately, people frequently use the terms interchangeably. The following short glossary defines commonly used terms which have very different meanings and organizational implications.

Evaluation

Evaluation is the process of testing a product or system for a specific set of properties, requirements, or behaviors. Evaluation is usually performed by the National Computer Security Center (NCSC). Products passing the Evaluation process are listed on the *Evaluated Products List*, which is part of the National Security Agency's *Information Systems Security Products & Services Catalog*.

Accreditation

Accreditation is the authorization to operate in a certain manner and in a particular environment using a system which may or may not be comprised of evaluated products. Accreditation must be performed for every system deployed by the government which will process classified information. Accreditation is performed by the Designated Approving Authority (DAA).

Certification

Certification is the assertion that a product has successfully complied with some specification. Certification is part of the accreditation process. Frequently, this term is wrongly used as a synonym for Evaluation.

What is a Trusted System?

The Orange Book talks about "trusted" systems rather than "secure" systems. The words aren't really synonymous. No system is completely secure. Any system can be penetrated — given enough tools and enough time. But systems can be trusted, some more than others, to do what we want them to do, and what we expect them to continue to do over time. The central concept of the

Orange Book is that it's possible to measure this trust — to build, evaluate, and certify a system that conforms to a specific set of security criteria, and that therefore merits a certain overall security rating. The following sections summarize very briefly the high points of the Orange Book evaluation classes.

D Systems: Minimal Security

According to the Orange Book, division D "…is reserved for systems that have been evaluated but that fail to meet the requirements for a higher evaluation class." The Orange Book lists no requirements for division D. It's simply a catch-all category for systems that don't provide the security features offered by systems more highly rated.

In fact, there are no evaluated systems in this class; a vendor wouldn't go to the trouble of getting a system evaluated if it didn't offer a reasonable set of security features. It's likely that the most basic personal computer systems (e.g., IBM PC running MS-DOS, Apple Macintosh) would fall into the division D category if they were evaluated.

C1 Systems: Discretionary Security Protection

C1 systems provide rather limited security features. The Orange Book describes C1 systems as an environment of "…cooperating users processing data at the same level(s) of security." At this level, security features are intended primarily to keep users from making honest mistakes that could damage the systems (e.g., by writing over system memory or critical software) or could interfere with other users' work (by deleting or modifying their programs or data). The security features of a C1 system aren't sufficient to keep a determined intruder out.

The two main user-visible features required in the C1 class are:

- Passwords (or some other mechanism) — These identify and authenticate a user before letting him or her use the system.
- Discretionary protection of files and other objects — With systems of this kind, you can protect our own files by deciding who will be allowed to access them.

C1 systems must also provide a system architecture that's capable of protecting system code from tampering by user programs. The system must be tested to ensure that it works properly and that the security features can't be bypassed in any obvious way. There are also specific documentation requirements.

Few vendors consider having their systems evaluated at this low a level of security. Many people feel that if an ordinary UNIX systems (with no enhanced security features) were submitted for evaluation, it would be given a C1 rating (others claim that it would get a C2 rating).

C2 Systems: Controlled Access Protection

C2 systems provide much more stringent security than C1 systems. In addition to providing all the features required at the C1 level, C2 systems offer the following additional user-visible features:

- Accountability of individual users — Through individual password controls and auditing (maintaining a record of every security-related action every user performs), the system is able to keep track of who's doing what in the system.
- More detailed discretionary controls — The Orange Book describes C2 requirements as follows: "These access controls shall be capable of including or excluding to the granularity of a single user." Through access control lists or some other mechanism, you must be able to specify, for example, that only Mary and Joe can read a file and that only Sam can change it.
- Object reuse — This feature makes sure that any data left in memory, on disk, or anywhere else in the system doesn't accidentally become accessible to a user.

The C2 system architecture must allow system resources to be protected via access control features. C2 systems also require more rigorous testing and documentation.

B1 Systems: Labeled Security Protection

B1 (and higher) systems support mandatory access controls (MAC). In a MAC system, every file (and other major object) in the system must be labeled. The system uses these sensitivity labels, and the security levels of the system's users, to enforce the system's security policy. In MAC systems, protection of files is no longer discretionary. You can't give another user access to one of your files unless that user has the necessary clearance to the file. The Orange Book imposes many specific labeling requirements on B1 systems, including labeling of all information that's exported from the system.

B1 systems must have a system architecture that more rigorously separates the security-related portions of the system from those that are not security-related. B1 systems also require more stringent testing and documentation. In the B1 class, documentation must include a model of the security policy supported by the system. This policy need not be a mathematical one (as is required in the higher classes), but it must be a clear statement of the rules enforced by the system's security features.

B2 Systems: Structured Protection

B2 and higher systems don't actually add many more specific user-visible security features to the set already required of B1 systems. Instead, they extend these features, and they require additional assurance that the features were designed and work properly. The Orange Book says that B2 systems must be "…relatively resistant to penetration."

In the B2 class, labeling features are extended to include all objects in the system, including devices. B2 systems also add a trusted path feature, allowing the user to communicate with the system in a direct and unmistakable way — for example, by pressing a certain key or interacting with a certain menu. The system design must offer proof that an intruder can't interfere with (or "spoof") this direct channel.

B2 systems support *least privilege*, the concept that users and programs should possess the least number of privileges they need, and for the shortest time necessary, to perform system functions. This concept is implemented via a combination of design, implementation, and assurance requirements. One example is the requirement that there be a separate system administrator and operator.

From a system design point of view, B2 systems require substantially more modularity and use of hardware features to isolate security-related functions from those that are not security-related. B2 systems also require a formal mathematical statement of the system's security policy, as well as more stringent testing and documentation. They also require that a configuration management system manage all changes to the system code and documentation and that system designers conduct a search for *covert channels* — secret ways that a developer could use the system to reveal information.

B3 Systems: Security Domains

There are no requirements for new user-visible features at the B3 level, but the system design and assurance features are substantially more rigorous. Trusted facility management (assignment of a specific individual as security administrator) is required, as is trusted recovery (procedures ensuring that security doesn't fail if the system does) and the ability to signal the administrator immediately if the system detects an "imminent violation of security policy."

The most notable difference between B3 and lower systems is the requirement that the system satisfy the reference monitor requirement by being simple, tamper-proof, and impossible to bypass. The Orange Book says that B3 systems must be "…highly resistant to penetration." The TCB is very tightly structured to exclude code not required for the enforcement of system security.

A1 Systems: Verified Design

At present, A1 systems are at the top of the security heap, although the Orange Book does discuss the possibility of defining requirements for systems that exceed current A1 requirements in areas of system architecture, testing, and formal verification.

A1 systems are pretty much functionally equivalent to B3 systems. The only actual feature provided by A1 systems beyond B3 requirements is trusted distribution, which enforces security while a trusted system is being shipped to a customer. What makes a system an A1 system is the additional assurance that's offered by the formal analysis and mathematical proof that the system design matches the system's security policy and its design specifications.

Compartmented Mode Workstation

Compartmented Mode Workstations (CMWs) are a new type of trusted system that has recently gained a lot of attention, particularly in military and intelligence applications. A CMW is a secure workstation providing a high-resolution monitor, a window manager, and a detailed set of security functions. Unlike workstations that act simply as terminals, CMWs have enough built-in security to operate as trusted computers. Although CMWs can operate in standalone mode, they are more typically attached to a local area network (LAN), communicating with other workstations and often with a central computer.

The idea of a CMW was initiated by the Defense Intelligence Agency (DIA), based on research done by Mitre. From a security point of view, CMWs are very useful because they allow workstation users to process multiple compartments of data simultaneously. Each compartment contains a certain type of data. A user who is cleared to access `TOP SECRET [VENUS]` data, for example, may not necessarily be cleared to access `TOP SECRET [ALPHA]` data. The workstation keeps the compartments internally separated and typically uses windowing capabilities to control access to data.

The DIA saw CMWs as being effective tools, particularly for intelligence analysts. The idea was that an analyst could open several windows simultaneously, examine a variety of intelligence sources (possibly those at different levels of security and in different compartments) and compose reports by "cutting" from source documents and "pasting" into a final document. The system would control the cutting and pasting by ensuring that the analyst couldn't read documents at a higher security level than his own (or in a compartment he's not cleared to access). The system would also keep the analyst from creating a document at a lower security level than that of the source documents pasted into the final document.

Actual CMW evaluations are performed using the security requirements stated in the DIA's *Security Requirements for System High and Compartmented Mode Workstations* (called the CMWREQS). System high workstations differ from CMWs in a few important way; for example, system high workstations process information at only one sensitivity level.

The CMWREQS are similar to the Orange Book requirements for B3 system; for example, they require mandatory access controls, access control lists, a trusted path, trusted facility management, and trusted recovery.

Information labels are an important addition to the standard Orange Book requirements imposed on CMWs. Information labels must be associated with all objects (e.g., files, windows, processes) that are accessible to the user. An information label is similar to a sensitivity label. However, in addition to the hierarchical classification (e.g., `TOP SECRET`) and the non-hierarchical compartments or categories (e.g., `VENUS`, `TANK`), an information label contains required markings — including code words and handling caveats — to be used for labeling data.

In the following example, information can be released only to U.S. sources and those in the United Kingdom:

```
TOP SECRET [VENUS, TANK REL UK]
```

In the next example, information cannot be released to foreign nationals:

```
SECRET [ALPHA, BETA, DELTA NO FORN]
```

There are two other important differences between sensitivity labels and information labels.

- Sensitivity labels permit access decisions to be made between subjects and objects; information labels simply represent the sensitivity of the information contained in a subject or an object.
- Sensitivity labels typically remain static for the life of the label; information labels are automatically adjusted as the information content of a subject or object changes. For example, in an X Window System-based CMW, the information label of a window would change as new information moves into the window. This adjustment of labels is called *floating*.

There are elaborate rules for how sensitivity labels and levels interact with information labels and levels.

Other Security Considerations

In addition to the secrecy and confidentially aspect of computer security addressed by the Orange Book, there are other important aspects of security to consider. These other aspects of computer security relate to system accuracy (sometimes called integrity) and availability.

Accuracy, Integrity, and Authenticity

A secure computer system must maintain the continuing integrity of the information stored in it. *Accuracy* or *integrity* means that the system must not corrupt the information or allow any unauthorized malicious or accidental changes to it.

In network communications, a related variant to accuracy known as *authenticity* provides a way to verify the origin of data by determining who entered or sent it, and by recording when it was sent and received.

In some environments, accuracy is the most important aspect of security; confidentiality of transactions is less important than the verifiable accuracy of the transactions. Authentication methods (such as checksums and parity checks) are used to ensure the accuracy of a transmission. Authentication techniques calculate a code and appended it to a message when the message is sent across a network. At the receiving end, the code is calculated again. If the two codes are identical, the message received is the same as the message sent — proof that it wasn't forged or modified during transmission.

Availability

A secure computer system must keep information available to its users. *Availability* means that the computer system's hardware and software keeps working efficiently and that the system is able to recover quickly and completely if a disaster occurs.

The opposite of availability is denial of service. Denial of service means system users are unable to get the resources they need. The computer may have crashed. There may not be enough memory or processes to run a program. Needed disks, tapes, or printers may not be available. Denial of service can be every bit as disruptive as actual information theft. When the Internet worm attacked systems, it didn't actually destroy or modify any data. Instead, it attacked system availability by spreading through networks, creating new processes, and effectively blocking all other work on the infected computers.

In some ways, availability is the baseline security needed by everyone. If you can't use your computer, you won't be able to tell whether your secrecy and accuracy goals are being met. Even users who abhor "security" agree that their computer systems have to keep working. Many of them don't realize that keeping systems running is also a type of security.

Network 3

NIS+: Planning for Setup

By – Rick Ramsey – SunSoft, Inc.

Note – This article is a reprint of Chapter 6, "Planning for Setup," of *All About Administering NIS+*, a retail book available in bookstores worldwide.[1]

This article provides information to help you plan the setup and administration of your NIS+ service. It begins by recommending a setup procedure that works for most namespaces. Based on the steps of that procedure, it points out the decisions you need to make and the information you need to gather *before* you begin to set up NIS+.

Because many sites already use NIS, this article also provides guidelines for managing the transition from the NIS service to NIS+.

The chapter has the following sections:

- Recommended Setup Procedure
- Planning Guidelines
- Transition Guidelines

1. Rick Ramsey, *All About Administering NIS+*, SunSoft Press / Prentice Hall, Englewood Cliffs, NJ (paperback; ISBN 013-068800-2).

Recommended Setup Procedure

Figure 3-1, on the following page, summarizes the recommended setup procedure. The blocks on the left represent the major setup activities, such as setting up the root domain or setting up a client. The text in the middle describes the blocks. The chapter numbers (referring to chapters in the complete *All About Administering NIS+* book) and tasks on the right point to the instructions that describe how to carry out each block.

Planning Guidelines

The order of activities shown in Figure 3-1 is not mandatory. You could set up your namespace differently. For instance, you could set up all your servers first, then create their domains. However, to keep this planning discussion clear, we will discuss only the scenario in Figure 3-1. Based on that scenario, we recommend that before you begin to set up your namespace, you go through the following planning phases:

- Sketching a Domain Hierarchy
- Selecting Servers for the Namespace
- Determining Credential Needs of the Namespace
- How Many Administrative Groups?
- Determining Access Rights to the Namespace

Each phase is described in its own section below. In addition, each section includes a summary of the information you should have gathered for that phase.

A Summary Planning Sheet is provided at the end of this section. If you fill it out, you will have much of the information you need to begin setting up your namespace. Additional copies are provided in Appendix A of *All About Administering NIS+*.

	Activity	Description	Chapter of Full Book	Task
1	**Set Up Root Domain**	Create the root domain, and set up and initialize the root master server. Create root domain admin group and administrator credentials.	7	▼ How to Set Up the Root Domain
2	**Populate Tables**	Populate the NIS+ tables of the root domain from text files.	8	▼ How to Populate NIS+ Tables from Files
3	**Set Up Root Domain Clients**	Create credentials for clients of the root domain and set up the clients. Some of them will subsequently be converted into servers.	9	▼ How to Set Up an NIS+ Client
4	**Set Up Servers**	Convert some clients of the root domain into servers. Some will later become root replicas, others will support lower level domains.	10	▼ How to Set Up an NIS+ Server
5	**Designate Root Replicas**	Designate one or more of the servers just set up as a replica of the root domain.	10	▼ How to Add a Replica to an Existing Domain
6	**Set up Non-Root Domain**	Set up a new domain. Designate previously set up servers as its its master and replica(s). Create its admin group and admin credentials.	11	▼ How to Set Up a Non-Root Domain
7	**Populate Tables**	Populate the NIS+ tables of the new domain from text files.	8	▼ How to Populate NIS+ Tables from Files
8	**Set Up Domain Clients**	Create credentials for clients of the new domain and set them up. Some of them will subsequently be converted into servers for lower-level domains.	9	▼ How to Set Up an NIS+ Client
9	**Set Up Servers for Lower Level**	If the domain will have lower-level domains, convert some if its clients into the servers that will later become masters and replicas of the lower levels.	10	▼ How to Set Up an NIS+ Server
10	**Set Up Other Domains**	Set up other domains, whether parallel to the new domain or beneath it, by repeating blocks 6-9.		

Figure 3-1 Recommended NIS+ Setup Procedure

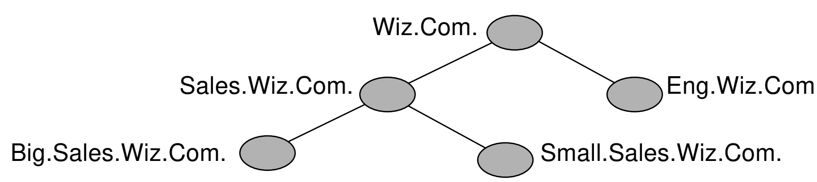
Sketching a Domain Hierarchy

The first planning step we recommend is to sketch a domain hierarchy. Make a map of how you expect the hierarchy to look when you are finished. This will be a useful reference when you are in the midst of the setup procedure. As a minimum, you will need to consider the following questions:

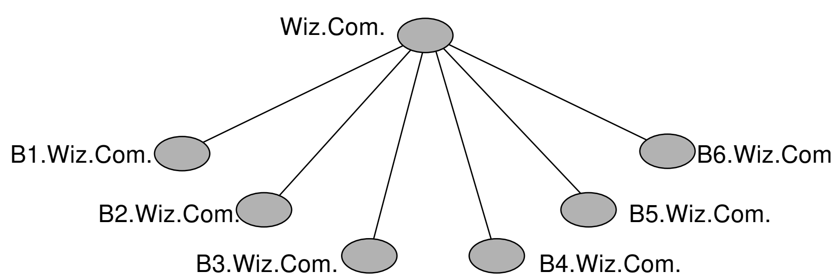
- What Kind of Hierarchy?
- Shoudl the Root Domain Support Clients?
- Which Domain Names?
- What Security Levels?
- Which Directories Will Support the Domains?

What Kind of Hierarchy?

One of the major benefits of NIS+ is its capability to divide the namespace into smaller, manageable parts. You could create a hierarchy of organizations, such as those of the imaginary Wizard, Inc.:



You could also organize the hierarchy by buildings instead of organizations:



The scheme you select depends primarily on how you prefer to administer the namespace and how clients will tend to use the namespace. For example, if clients of "Eng.Wiz.Com." will be distributed throughout the buildings of the Wizard Corporation, you shouldn't organize the namespace by building. Since the clients would need to constantly access other domains, you would need to add their credentials to the other domains and you would increase traffic flow through the root master server. A better scheme would be to arrange them by organization. On the other hand, building-sized domains are immune to the reorganizations that require organization-based domains to be restructured.

Don't be limited by the physical layout of the network, since an NIS+ namespace does not have to be congruent with the physical network.

The number of domains your namespace needs depends on the kind of hierarchy you select, but keep in mind that smaller domains process NIS+ requests faster than large domains.

Should the Root Domain Support Clients?

In the two domain hierachies illustrated above, are all the clients placed in domains beneath the root domain? Or do some belong to the root domain? Is the purpose of the root domain only to act as the root for its subdomains or will it support its own group of clients? You could place all clients in the lowest layer of domains, and only those used for administration in the intermediate domains. For example, in the first illustration, all clients would belong to the "Big.Sales.Wiz.Com.", "Small.Sales.Wiz.Com.", and "Eng.Wiz.Com." domains, and only clients used for administration would belong to the "Wiz.Com." and "Sales.Wiz.Com." domains.

Or, you could place the clients of general-purpose deparments in higher level domains. For example, in the second illustration, you could put the clients of the Facilities Department in the "Wiz.Com." domain.

Which Domain Names?

See "*Naming Conventions (To Dot or Not)*" in *All About Administering NIS+* for details about domain naming conventions. Within the constraints described in that section, you can choose any names you like, but here are two suggestions. First, pick names that are descriptive. "Sales" is considerably more descriptive than "BW23A". Second, pick short names. When you adminster the namespace, you want to avoid appending to object names something as odious as "EmployeeAdministrationServices.WizardCorporation".

Remember that all root domains must have at least two labels. One of those can be the name of an Internet domain.

What Security Levels?

In most cases, you'll run the namespace at Security Level 2. However, if you plan to use different security levels for different domains, you should identify them now. Chapter 4, "Understanding NIS+ Security," of *All About Administering NIS+* describes all three security levels offered by NIS+.

Which Directories Will Support the Domains?

Remember, a domain is simply a particular grouping of directories. You may create additional directories between domains for administrative purposes, although you don't have to. If you do, sketch a map of your namespace directories, complete with `org_dir` and `groups_dir` directories, so you don't get confused during setup.

Summary

As a result of these steps, you should have:

For the Domain Hierarchy

- A map of the namespace with the names of all the domains
- A brief description of the purpose of each domain (e.g., whether for clients or administrators, or to support a certain part of the organization, etc.)
- The security level of each domain
- Optionally, a map of the directories that form the framework of your namespace

Selecting Servers for the Namespace

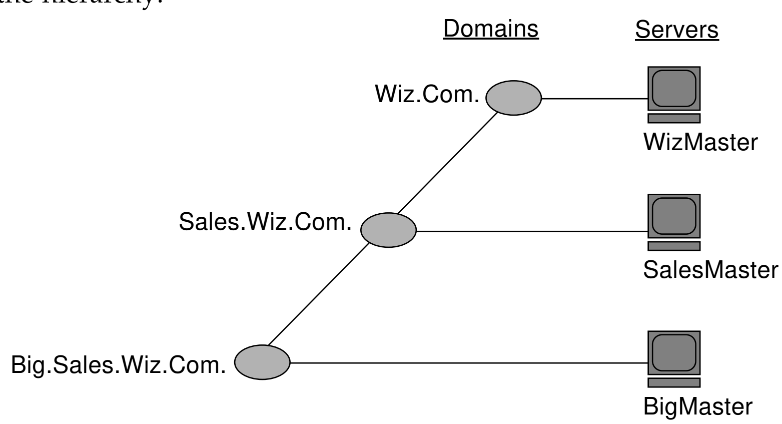
Once you know the domain structure of the namespace, you can select the servers that will support them. When selecting servers, you need to differentiate between the requirements imposed by the NIS+ service and those imposed by the traffic load of your namespace.

The NIS+ service requires you to assign at least one server — the master — to each NIS+ domain. You can assign as many replicas as required.[2] How many servers a domain requires is determined by the traffic load and the configuration of its servers.

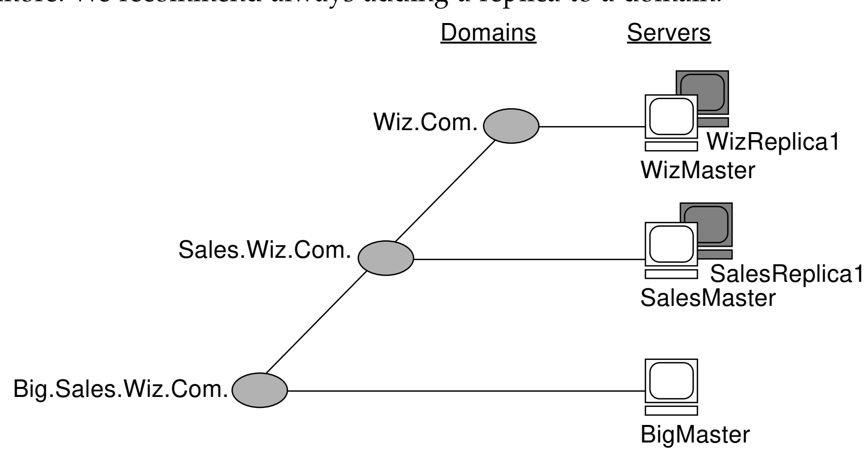
2. An NIS+ server is capable of supporting more than one domain, but we don't advise this except in small namespaces or testing situations.

The traffic loads you anticipate will determine the total number of servers used to support the namespace, how much storage and processing speed each will require, and whether a domain needs replicas to ensure its availability. How can you determine how many servers you need? Here's a recommended sequence of steps:

1. A good way to begin is by assigning one master server to each domain in the hierarchy:



2. If certain domains must always be available, add replicas to them; one or more. We recommend always adding a replica to a domain.



3. Determine the disk space requirements of each server (see "*Disk Space and Memory Requirements*" below). Based on the required disk space, estimate whether one server will be enough for each domain.

Disk Space and Memory Recommendations

How much disk space you need depends on four factors:

- Disk space consumed by Solaris 2.*x*
- Disk space for `/var/nis` (and `/var/yp`)
- Amount of memory
- Swap space required for NIS+ processes

Solaris 2.*x* can consume between 90 Mbytes and 130 Mbytes of disk space, including OpenWindows, depending on how much of it you install (this is an estimate — for exact numbers, see the Solaris 2.*x* installation manual). You should also count the disk space consumed by other software the server may use. NIS+ is part of the Solaris 2.*x* distribution, so it does not consume additional disk space.

NIS+ directory objects, groups, tables, and client information are stored in `/var/nis`. As a rule of thumb, `/var/nis` uses about 1K of disk space per client. So if a namespace has 10,000 clients, `/var/nis` requires about 10 megabytes of disk space. However, because transaction logs (also kept in `/var/nis`) can grow large, you may want additional space per client — we recommend an additional 60-70 percent. In other words, for 10,000 clients, allocate 16 to 17 megabytes for `/var/nis`. You can reduce this if you update transaction logs regularly. If you will use NIS+ concurrently with NIS, allocate an equal amount of space for `/var/yp` to hold NIS maps that you may transfer from NIS. By the way, if you can, keep `/var/nis` on a separate partition.

Although 32 megabytes is the minimum memory requirement for servers, we recommend that you equip servers of medium-to-large domains with at least 64 megabytes.

You also need swap space equal to three times the server's memory — in addition to the server's normal swap space requirements. Most of this space is used when directories are checkpointed (with `nisping -C`), since during a checkpoint, an entire NIS+ server process is forked.

Summary

As a result of these steps, you should have:

For Servers

- A list of all the servers required by the namespace, their disk, swap space, and memory requirements, and the domain(s) each server will support. For example, for a namespace with 10,000 clients:
  - 130 Mbytes for Solaris 2.*x*
  - 17 Mbytes for `/var/nis` (plus 17 Mbytes more for `/var/yp`)
  - 32 to 64 Mbytes memory
  - swap space equal to triple the memory (in addition to normal swap space)
- A map of all the domains in the namespace with the names of the servers that will support them.

Determining Credential Needs of the Namespace

Once you know the domain hierarchy of the namespace and your server specifications, you should determine what kind of credentials will be required in the namespace. When doing so, you should consider the factors discussed below.

For the Client User or for the Client Workstation?

As described in Chapter 4, "Understanding NIS+ Security," of *All About Administering NIS+*, NIS+ principals can be client users or the `root` identity on the client workstation. When you determine the credentials you need to create, make sure you know which type of principal the credential is for.

For instance, when you set up an NIS+ client according to the instructions in Chapter 9 of *All About Administering NIS+*, you will create credentials for the client workstation, not for the client user. Unless credentials for the client user are created, the client user would only have the access rights granted to the Nobody category. This can work perfectly well if that's how you choose to set up your namespace. But if you don't give any access rights to the Nobody category, the namespace won't be available to the client users.

Which Type of Credential?

The type of credentials required in a namespace is determined primarily by the security level at which each domain is running. If a domain is going to be operated in NIS-compatibility mode (security level 0), no one needs credentials of any kind. If the domain is going to operate at security level 1, any NIS+ principal that will have access rights, in addition to those granted to the Nobody category, needs LOCAL credentials. At security level 2, they need DES credentials.

For Which Domains?

For security level 2, all NIS+ principals must have DES credentials in their home domain. A client workstation cannot have credentials in any other domain. However, a client user can have LOCAL credentials in other domains, in addition to the DES credentials in its home domain. So, if you anticipate that users will request information from other domains in the namespace, you can give them the proper access in one of two ways. You can create LOCAL credentials for them in additional domains, or you can assign those domains Read rights to the World category, which includes all authenticated NIS+ principals. Which you select depends on how secure you want to make your namespace.

Summary

As a result of these steps, you should have:

For Credentials

- The security level of each domain
- A list of the type of credential required for servers, clients, administrators, and users, per domain

How Many Administrative Groups?

After identifying the type of credentials you'll need, you would normally try to select the access rights that are required in the namespace. To make that task easier, you should first figure out how many administrative groups you'll need. Will you create one group for the entire namespace or one group per domain? Using separate groups is useful if you want to assign them different rights.

To determine the number of groups you need, determine the classes of principals you will have in your namespace. Then create a group for each class. You can also create groups of groups. For example, principals who are simple users of the namespace form one class of principal. They only need Read access to the tables in the namespace. On the other hand, principals who administer the namespace need much greater access rights. Some of them may only perform basic administration tasks, like adding user accounts, so they simply require Read and Modify rights to the namespace tables. Others, however, may actually set up or modify the namespace, so they will need Read, Modify, Create, and Delete rights to the entire namespace — or perhaps only to a single domain in the namespace. The instructions in Part II of *All About Administering NIS+* include steps for setting up each domain's main administrative group. You can set up additional groups after setting up the namespace.

Here's an example of groups for the namespace of the Wizard Corporation:

Group Name	Description
clients.Wiz.Com.	All clients in the Wiz. domain
clients.Sales.Wiz.Com.	All clients in the Sales.Wiz. domain
clients.Eng.Wiz.Com.	All clients in the Eng.Wiz. domain
admin.Wiz.Com.	All administrators in the Wiz. domain only
admin.Sales.Wiz.Com.	All administrators in the Sales.Wiz. domain
admin.Eng.Wiz.Com.	All administrators in the Eng.Wiz. domain
adminall.WizCom.	All administrators in the Wizard Inc. namespace

You don't need a group for all the principals in the namespace, something like "all.Wiz.Com.", because the "World" category already accomplishes that.

Summary

As a result of these steps, you should have:

For Groups

- A list of the groups in the namespace, with a description of their membership

Determining Access Rights to the Namespace

After arranging your principals into groups, you should determine the kind of access rights granted by the objects in the namespace to those groups, as well as to the other categories of principal (Owner, World, and Nobody). Planning these assignments ahead of time will help you establish a coherent security policy.

NIS+ provides preset defaults that apply to every object created in the namespace. They are:

Nobody	Owner	Group	World
	Read	Read	Read
	Modify		
	Create		
	Destroy		

You can use these defaults for all your objects or assign different rights to different types of objects. You'll need to consider two main factors:

- The type of object
- The domain

Here's a useful way to proceed:

1. Read the discussion about "Types of Principals" below. Use it to help you determine the kinds of access rights different types of principals will require.
2. Read the discussion about "Types of Objects" below. It divides the objects normally found in a namespace into several classes. This is an informal division; you can divide the objects in your namespace into any classes you prefer.
3. For each domain, determine the access rights each class of object should assign to each category of principal: Owner, Group, World, and Nobody. For the Group category, don't just specify the access rights — also specify one of the groups you already selected for the namespace.
4. During setup, as you create an object, classify it by type and and assign it the rights specified in your table.

Types of Principals

NIS+ divides principals into four categories: Owner, Group, World, and Nobody. You can determine most of the access rights you'll need by using these categories. For instance, if you don't want any unauthenticated requests to be granted, don't assign any rights to the Nobody category. On the other hand, if you expect to have some clients make unauthenticated requests, you can assign Read rights to the Nobody category.

Depending on the type of table information, you should assign the World category at least Read rights, since it comprises all authenticated principals in the namespace. That allows a Read request from an authenticated client anywhere in the namespace to be granted.

For each type of object (see "Types of Objects" below), you can pick an "owning" group. Depending on how you plan to administer the namespace, you can assign all the available administrative rights to the group, or only some of them. For instance, if only one administrator will create and modify the root domain, you should not assign the owning group Create and Destroy rights to the objects in the root domain. Instead, assign those rights only to the objects' Owner. However, if several administrators may be involved in the setup process, put them all in a group and assign full rights to it. That is easier than switching ownership back and forth.

Finally, the owner should have full rights, although this is not as important if the group does. A namespace is more secure if you give only the owner full rights, but easier to administer if you give the administrative group full rights.

Types of Objects

Objects in an NIS+ namespace can be informally classified into two categories: administrative and informative. Directories and groups are strictly administrative. There is no need to give the World category, for instance, Modify rights to a directory or group. It would compromise security and the structure of the namespace.

Tables and their entries are informative. The World category, and perhaps the Nobody category, should have Read rights to tables and entries. Depending on how secure you want the namespace, you may be willing to grant Modify rights for certain table entries to the groups that comprise all clients in a domain.

Summary

As a result of these steps, you should have:

For Access Rights

- A chart that specifies the types of objects in each domain and, for each category of NIS+ principal, the access rights granted by each type of object

Where Can I Get Information About the Namespace?

NIS+ provides several commands that provide information about the data and objects in the namespace. Here is a list:

Command	Lists	In Full Book
`niscat -o`	Object properties	Chapter 16
`niscat`	Table contents	Chapter 17
`nisls`	The contents and other particulars of a directory	Chapter 16
`nislog`	The contents of a transaction log	Chapter 17
`nisshowcache`	The contents of the directory cache	Chapter 16

By the way, to exit from an NIS+ command, simply use Control-c.

Summary Table

When you have designed the namespace and selected its servers, types of credentials, groups, and access rights, you have gathered most of the information you'll need to set up the namespace. You might want to summarize all this information in a worksheet that you can keep handy while you set up the namespace. Table 3-1, below, provides a sample worksheet that uses fictitious information for the Wizard, Inc. namespace. Appendix A of *All About Administering NIS+* provides additional blank worksheets in case you would like to use them.

Table 3-1 Sample Pre-Setup Worksheet

Domain: Wiz.Com.

Servers	**Type**	**Name**				**Specifications**
	Master	rootmaster				
	First Replica	rootreplica				
	Second Replica	rootreplica2				
Credentials	**Type of Principal**	**Type of Credential**				
	Servers	DES				
	Clients	DES				
	Administrators	LOCAL and DES				
	Users	LOCAL and DES				
Rights	**Types of Objects**	**Category & Rights**				
	Directories	**N**	**O**	**G**	**W**	**Use Defaults?**
	Wiz.Com.	`r`	`rmcd`	`rmcd`	`r`	no
	org_dir.Wiz.Com.	`r`	`rmcd`	`rmcd`	`r`	no
	groups_dir.Wiz.Com.	`r`	`rmcd`	`rmcd`	`r`	no
	Sales.Wiz.Com.	`r`	`rmcd`	`rmcd`	`r`	no
	Groups	**N**	**O**	**G**	**W**	**Description**
	admin.Wiz.Com.	`r`	`rmcd`	`r`	`r`	Admins who manage structure & security
	admin2.Wiz.Com.	`r`	`rmcd`	`r`	`r`	Admins who manage information
	Tables	**N**	**O**	**G**	**W**	**Notes**
	`hosts`	-	`rmcd`	`rmcd`	`r`	
	`bootparams`	-	`rmcd`	`rmcd`	`r`	
	`passwd`	-	`rmcd`	`rmcd`	`r`	
	`cred`	-	`rmcd`	`rmcd`	`r`	

Table 3-1 Sample Pre-Setup Worksheet (*Continued*)

Domain: Wiz.Com.

Rights	**Types of Objects**	**Category & Rights**				
	Tables	**N**	**O**	**G**	**W**	**Notes**
	group	-	rmcd	rmcd	r	
	netgroup	-	rmcd	rmcd	r	
	aliases	-	rmcd	rmcd	r	
	timezone	-	rmcd	rmcd	r	
	networks	-	rmcd	rmcd	r	
	netmasks	-	rmcd	rmcd	r	
	ethers	-	rmcd	rmcd	r	
	services	-	rmcd	rmcd	r	
	protocols	-	rmcd	rmcd	r	
	rpc	-	rmcd	rmcd	r	
	auto_home	-	rmcd	rmcd	r	
	auto_,master	-	rmcd	rmcd	r	

Transition Guidelines

This section provides information and guidelines about the transition from NIS to NIS+. The transition from NIS to NIS+ can be handled in several different ways. Following are some guidelines:

1. Have the administrators who will convert your site from NIS to NIS+ build a small NIS+ namespace so they can become familiar with NIS+ before they have to deal with the added complications of a transition from NIS. When you set up the test domain, make small, manageable domains. Set up the namespace first as a pure NIS+ domain, then convert it to NIS-compatibility mode. Then import real NIS information to test.

   You may also want to consider how to train the rest of your administration staff, as well as how to communicate the changeover to your NIS clients.

2. Design the final NIS+ namespace, following the guidelines in the first part of this chapter. Consider whether you need to maintain existing e-mail addresses of clients as well as existing login and hostnames.
3. Consider how NIS+ will interact with DNS. If the DNS domains are repartitioned, you'll need to redefine new DNS zone files; otherwise, no changes are required in the DNS environment. Clients, however, do require some adjustments (see Task 7 below).
4. Determine whether you'll need to change your e-mail environment.

   Because NIS+ offers a domain hierarchy but NIS has a flat domainspace, changing to NIS+ can have effects on your mail environment. With NIS, only one mailhost is required. If you use a domain hierarchy for NIS+, you may need one mailhost for each domain in the namespace.

   In addition, the clients who are not in the root domain will have their e-mail addresses changed. You'll have to find a way to compensate, such as using aliases files or correcting the "`From:`" fields.
5. Select the domains that will operate in NIS-compatibility mode. These domains will be less secure than NIS+ domains. In addition, changing passwords in NIS requires the `yppasswd` command. To make this change take effect in the NIS+ domain, you must repeat it with the `nispasswd` command.

   You'll need to figure out a way to keep the NIS+ domains in synchronization with the NIS domains. At first, you may want to update the NIS+ tables nightly with the latest NIS information. Be sure to also update the NIS maps from the NIS+ tables. Instructions for both types of transfers are provided in Chapter 8, "Setting Up NIS+ Tables," of *All About Administering NIS+*.

   At first, NIS will no doubt be the primary service. As you become familiar with the intricacies of sharing information, you'll be able to plan a transition into making NIS+ the primary service.

   Some NIS+ clients may want the capability to switch back and forth between the main NIS domain and the new NIS+ domain. Consider whether you can provide a script that helps them do this easily.

6. Take stock of your NIS servers, keeping in mind the requirements for your NIS+ servers. If you plan to eventually use them for the NIS+ service, upgrade them to the NIS+ recommendations. Identify which NIS servers will eventually be used to support which NIS+ domains, and in what capacity (master or replica).

   Since NIS+ is provided in a Solaris 1.*x* distribution, determine whether the servers will need to be upgraded to Solaris 2.*x* or whether they will use the Solaris 1.*x* distribution.

   If possible, plan to use your NIS+ servers only for NIS+. This may require you to transfer other network services, such as DNS name services, home directories, NFS, etc., to servers not scheduled for NIS+ use.

7. Make a list of your NIS clients and map them to their eventual NIS+ domains. Determine whether they will need to be upgraded to Solaris 2.*x* or whether they will use the Solaris 1.*x* distribution of NIS+.

   NIS servers handle DNS request made from Solaris 1.*x* clients. However, NIS+ servers don't provide this service. As a result, all NIS clients must have an `/etc/resolv.conf` file and the DNS resolver libraries installed locally before they are disconnected from their NIS servers.

8. Resolve conflicts between login names and hostnames. Although it gives login names precedence, NIS+ does not distinguish hostnames from login names, so if the same name is used for both, you'll run into problems with permissions and the automounter tables. You may need to alias the duplicate hostnames.

9. Change the dots in your automounter files to underscores. NIS+ uses dots to separate directories, so the dots in the automounter files will confuse it. Replace them with underscores. This involves several steps.

   First, change the file names. Verify that the automounter still works in your NIS environment. Then add the new automounter targets in each NIS domain master's `/var/yp/Makefile`, and build the maps. Also doublecheck the contents of all the automounter files and change any automounter references that appear in them. Finally, install the new, just-built maps into each NIS server in each domain, using `ypxfr`.

   Getting the automounter maps working properly before the transition will eliminate them as a possible cause of post-transition problems, if any occur.

NIS+ Transition Planning

This article, which addresses the subject of NIS-to-NIS+ transitions, is assembled from Sun internal training materials.

Introduction

The purpose of this article is to enable readers to plan and implement the conversion of networks from their current name services to NIS+.

The reader is assumed to have a basic understanding of NIS and NIS+ concepts, and some familiarity with the NIS+ commands in the Solaris 2.*x* releases.

Material in this article should enable the reader to:

- Understand the factors that influence the planning of conversions for network name service.
- Follow a checklist to prepare a conversion plan.
- Prepare a Table Planning Worksheet to create site-specific name service maps.

Prerequisites

This article covers advanced topics in name service administration. Basic principles of name services, NIS, and NIS+ are not covered in this document. Readers are assumed to be familiar with the basic concepts of the NIS+ name service, and should be able to perform some basic tasks:

- Define basic NIS+ terms
- Explain NIS+ concepts
- Configure NIS and NIS+ clients
- Configure NIS+ servers using the documentation

Other References

In addition to the SunSoft Press book *All About Administering NIS+* (see previous article in this chapter), there are other documents from past issues of the STB that are of interest to administrators wanting to learn more about NIS+:

- “NIS+” and “NIS+ Technical White Paper” (May 1992)
- “NIS+: An Enterprise Naming Service” (October 1992)
- “Name Services” (November 1992)

Planning Checklist

A sample checklist is included on the following pages. The checklist is all-inclusive and can be used for all networks. One result of its completeness is that it is long, that it may cover steps that are not relevant for any one particular network, and that it may be more difficult to follow for administrators of simple networks. *Not all administrators have to plan for every step to successfully convert a network!* Simple cases are the most typical cases. Figure 3-2 illustrates that the transition is not necessarily as complex at it might appear.

In general, keep the transition *simple* at first. If possible, follow these tips:

- Don’t change domainnames.
- Don’t add hierarchies.
- Use the NIS-compatibility mode feature.
- Use default tables and directory structures.
- Defer maintaining user credentials for the short term.

	Complexity	
Size	1 Site	2+ Sites
0-1000 Hosts	1 Domain *Keep It Simple*	Split Domains
1000-3000 Hosts	2-3 Domains	
3000+ Hosts	5-10 Domains	

Figure 3-2 Domain Planning

The planning checklist has two types of items:

- *Planning* tasks — to be done prior to starting a conversion
- *Conversion* tasks — done at the actual time of conversion

The standard NIS+ tables are created by default with the correct attributes. If you are planning to use site-specific tables, then table worksheets are very helpful in avoiding mistakes during the planning process, and are also useful during the actual conversion because they help avoid simple mistakes. (See “Table Planning Worksheet” at the end of this article.)

Decisions

1. Decide Whether to Upgrade

♦ **Decide whether your site should upgrade to NIS+ at this time, or wait until a better time to upgrade the name service.**

Planning and converting is a significant administrative task — less work in small networks, more in larger ones. Deferring the upgrade to NIS+ may simplify your life in the short run. The benefit would be more time for other Solaris-related tasks; but the work would need to be done later.

Planning Tasks

2. Educate Yourself on NIS+

♦ **Educate yourself so that you understand the basics of the NIS+ service.**

It is difficult for an administrator to simultaneously learn both the basics about the NIS+ service and plan for a transition in name service. Before starting the plan, you should be able to perform basic tasks such as the following in a classroom or laboratory environment:

- Installing and configuring a client-side system to use the name service
- Configuring a master server
- Configuring replica servers
- Using NIS+ commands to copy data from existing NIS maps into NIS+ tables

Sun recommends that prior to deploying the NIS+ service in production fashion on a network, the administrators working with the NIS+ service learn the basic operations with a small, multi-level test domain.

The name service administration manual in the Solaris documentation, *SunOS 5.1 Administering NIS+ and DNS*, is useful for reference while you are learning. Among other things, it graphically describes the use of the Database Manager for maintenance of data in the NIS+ tables. *All About Administering NIS+* from SunSoft Press / Prentice Hall is another valuable source of introductory explanation.

3. *Document Your Current Configuration*

♦ **Document your current name service configuration.**
Documenting the current configuration ensures that you clearly understand the starting point for the planning effort. This step need not be overly complicated. Your documentation should include:

- Names of all current domains, and the physical locations
- Host names of all current name servers and slave servers, and their locations

4. *List System Configurations*

♦ **List the system configurations for the existing name servers.**
The NIS+ service consumes more system resources than does NIS. Therefore, the minimal recommended system configuration for an NIS+ server is somewhat larger than for an NIS server.

In step 12, you will verify that all the designated NIS+ servers meet the expected system requirements. Step 12 can be completed more quickly if you have already prepared a list of the existing name servers. For large networks, compiling the list of current name servers can be a significant task. The list should include:

- CPU type
- Hostname
- Main memory size
- Disk space available
- Who has `root` access

5. *Design Name Space*

♦ **Design the new name space and list the designated NIS+ servers.**
Factors to be considered in designing the name space include the items listed below:

Notes – If you leave the current domainname unchanged, the transition will involve the least amount of short-term administrative work.

NIS+ currently requires domainnames to contain two or more labels, and will not accept a domainname containing only one label. This will change in a future Solaris 2.*x* release.

---

i. **Internet**
If your network is connected to the Internet, or if you plan to connect to the Internet, you should request a network name and number from the Network Information Center.

If the root domain is not connected to the Internet, you must still assign it a name with two labels.

If your NIS+ domain is a sub-domain of another NIS+ domain, that will also determine the name of the highest level.

ii. **Natural Administrative Unit**
Administration will be greatly simplified if the name structure reflects the manner in which administration is normally conducted. The natural boundaries of administrative support are often in terms of:

- Building, Site, Floor, or Geography
- Department or Organization

iii. **Sizes of Domains**
The NIS+ service was optimized to run most efficiently with around 1000 users per domain.

Reasons for Larger Domains
One large domain requires less administration than multiple smaller domains. Fewer skilled NIS+ administrators are needed in larger domains. As a result, administrative training expense is also lower in large domains.

Smaller domains are often structured with large number of tables joined by concatenation path. This will increase the dependence on other domains.

Smaller domains result in more domains, more boundaries, and more complex domainnames, all of which tend to confuse users.

Reasons for Smaller Domains
In smaller domains, it is easier to customize tables and table entries. Permissions and access rights may appear to be more easily customized in small domains, but can handled just as efficiently by defining groups within a larger domain.

Number of Servers Available
Every domain requires a server. The budget allocation for systems to be used as servers is usually limited. Any design that results in significantly more domains than systems available to be servers is probably flawed.

iv. Levels of Hierarchy
The NIS+ service was designed to handle multiple-tier hierarchies. There are no hard-and-fast restrictions on the numbers of levels permitted. In the case of a proposed name space that has many levels — say, more than six — it is more likely that this design would prove administratively unworkable, rather than that the NIS+ software could not accommodate it.

There are no hard-and-fast restrictions on the number of sub-domains within a domain, but more than 20 or so is likely to prove unworkable in practice.

v. Redundancy Level
Configuring one or more replicated name servers for each domain provides two benefits:

- Name service continues even if one name server is unavailable.
- Lookup performance is improved because the load is spread across multiple servers.

NIS+ master servers require fewer replicas than did NIS servers. Because NIS+ does not depend on broadcasts as NIS did, the typical model of two NIS slave servers per sub-net is not needed for client binding.

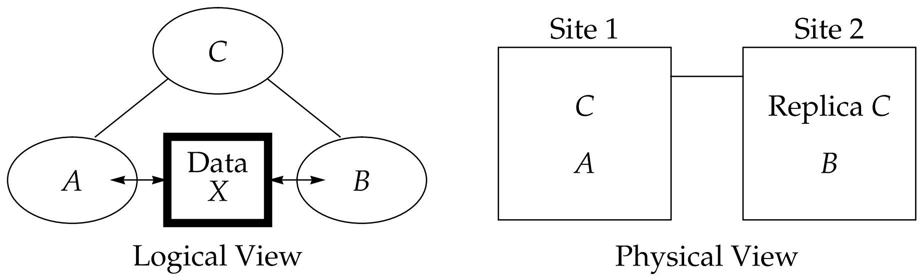
Putting replicas on both sides of weak network links is a good idea. When the link breaks and the networks are decoupled, name service continues on both sides. Examples of such weak links are optical links, some wide-area links, or interconnections between buildings.

Furthermore, there is a runtime lookup cost that varies with the number of NIS+ replicas. For all these reasons, in small- to medium-sized domains, typical configurations of 2-4 NIS+ replicas per domain are appropriate. In larger or widely distributed domains, a reasonable strategy would be to have a replica serving each distinct physical location, such as a building.

vi. Keep Transitory Labels Out of Names
Once established, domainnames are configured onto every system. Changing domainnames after the fact is very expensive. Therefore, irrelevant or transitory labels should be kept out of the name space. This can be a counterargument to the use of organizational names for domainnames, because organizational names are often transient.

vii. Decide Where Tables Will Reside

The diagram below illustrates a general principle useful in deciding where to place tables. Given a Table *X*, if clients in both domains *A* and *B* want to access all of *X*, then *X* should be moved higher into the parent domain *C*.



In the physical view, *C* is replicated in the home sites for both *A* and *B*. This ensures that the data in *C* is highly available to clients in both domains.

- Every domain must have access to every standard table.
- Volatile, frequently accessed data should be kept low in the hierarchy.
- Administrative autonomy argues for keeping data lower in the hierarchy.

viii. Point-to-Point Links

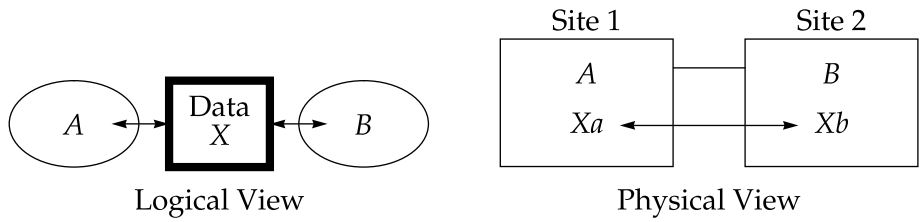
Some organizations have many sites — such as sales offices — joined by point-to-point links, each site of which needs to be its own sub-domain. Every sub-domain master is also required to be in a higher domain. This might at first seem to imply that all lookups on the servers have to go over point-to-point links.

The recommended solution is that the master server for each also be a replica of the master of the domain above. Presumably, the domain above is not generating as many changes as the sites. Lookups are then handled locally, and only changes to the higher domain propagate over the point-to-point link.

The general recommendation is to put a replica of the greatest common ancestor domain at each site.

ix. Concatenation Paths

In the illustration below, if clients in domains *A* and *B* both want to access a subset of the data in table *X*, then a concatenation path would be appropriate. The table *Xa* would be defined to concatenate to table *Xb*, and vice versa. Lookups performed by clients on one table would see the data in the other table.



You may think at first that you want to path tables, but you probably don't. The value of concatenation paths is that they allow an administrator to make changes in a table that appears to clients in other domains.

For security reasons, Sun does not recommend that you attempt to path either the `passwd` or `cred` tables.

x. Links

Links are useful only in a small number of cases in which you want to access the information in a table, but you don't want to administer it (e.g., the `services` table).

6. *Verify Servers Meet System Requirements*

♦ **Verify that all designated name servers meet the expected system requirements, including the name server you will use as a master for the initial shakedown period.**

Using the list of existing name servers prepared in Step 5 and the system requirements (listed in the previous article, or from the manual), confirm that all systems designated as new name servers will meet the requirements. Verifying this early will simplify equipment planning and budgeting, and will prevent surprises during the actual upgrade operation. For each name server, you should analyze CPU power, main memory, swap space, disk space and partitioning.

The requirements for replica servers are essentially the same as for master servers.

Older systems such as Sun-4/280s that may have been previously used as NIS servers are not recommended for use as NIS+ servers.

7. *Divide Labor of Network and Name Services*

♦ **Consider how to divide the labor for network services and applications between designated NIS+ servers and other network servers.**

NIS servers often provide additional services, such as DNS name services, NFS file services for `/home` directories, software distribution or mailboxes, routing, mail forwarding, modem or remote session control, database service, xhost service, etc.

It is important to all users on a network that name service lookups are rapid. Because the NIS+ service uses more system resources than did NIS, and because it is difficult for administrators to estimate the expected performance, Sun recommends that systems used as NIS+ servers be subject only to minimal loading from other services.

Troubleshooting, maintenance, and performance monitoring will be easier if the NIS+ server is the only significant activity on the system. With NIS+, you need fewer replicas than you did with NIS, which should allow you to remove older NIS slave servers and put them to work as routers, NFS servers, etc. Balance the benefits (fewer replicas) against the potential costs (more servers) and rethink the division of labor among your servers.

8. *Plan for E-mail Changes*

♦ **If the e-mail environment changes as a result of a change in the name space hierarchy, plan to maintain the new `sendmailvars` table.**

You can skip this step if you don't change the domainname and you don't add hierarchies to a domain.

The e-mail addresses of clients in lower domains will change when domainnames change, or when an extra domain level is introduced into an existing hierarchy. Traditional solutions to this problem involve extra work on the part of the administrator to add new mailhosts in the sub-domains, or to change the `sendmail.cf` file on the client.

Solaris 2.1 and subsequent releases are configured so that by default you do not have to reconfigure the `sendmail.cf` file on the client. This was accomplished by enhancing `sendmail` to substitute mail domains; creating new `L` macros in `sendmail.cf`; creating a new standard `sendmailvars` NIS+ table and the file `/etc/mail/sendmailvars` (which should be maintained if you change lower domains); and having `sendmail` look first for an NIS+ table, then examine local files.

Note that if you change a domainname, you may want to change the embedded domainname in the `netgroup` table.

It is a bad idea for mailhosts to access host tables via pathed host tables: performance will suffer because mistyped hostnames will cause sequential searches of all tables in the path.

9. *Decide on NIS-Compatibility Mode*

♦ **Decide whether to use the NIS-compatibility mode feature.**
The advantage of running the NIS+ server with the optional NIS-compatibility mode is short-term administrative flexibility. By running the NIS-compatibility mode feature, the name server can be upgraded to NIS+ without having to upgrade existing NIS clients, such as SunOS 4.*x* systems. Using the NIS-compatibility features, the conversion to NIS+ and Solaris 2.*x* can be done independently on the servers and clients. Also, many sites have NIS clients that are non-Sun systems; such heterogeneous sites can simplify their transition by using the NIS-compatibility feature.

The advantage is so straightforward that the *default* choice should be to use the NIS-compatibility feature. It is assumed that the domainname does not change.

If the `hosts` line in the Name Service Switch file (`/etc/nsswitch.conf`) does not already contain the entry `dns`, you will have to add it for NIS-compatibility mode. You will also have to add an `/etc/resolv.conf` file with appropriate data in it.

For organizations maintaining network passwords in NIS maps, one short-term consequence of using this features is that the `nispasswd` command update will update the NIS+ `passwd` table, and the `yppasswd` on the client will attempt to write to a non-existent NIS map.

Less probable circumstances in which a planner might choose not to use the NIS-compatibility mode include:

- If there are no existing NIS client systems on the network
- If any existing NIS client systems are planned to upgrade to Solaris 2.*x* and NIS+ at the same time as the name servers
- If the clients need special access to data in NIS+ tables that would not be visible in the restricted view of NIS+ data that is readable by a client

Sun does not recommend that you configure NIS+ and only use it to provide NIS service to NIS clients. Although this approach is possible, it provides none of the benefits of NIS+, and involves work to install.

10. *Keep External NIS Domains in Synchronization*

♦ **If there are external NIS domains to consider, determine how to keep the NIS+ domain in synchronization with them.**

You can skip this step if you don't have to share co-existing data with external NIS domains that are not transitioning on the same schedule.

In the case of very large organizations with multiple name service domains that are migrating to NIS+ separately, you should consider how to keep the name service data in synchronization. Organizations of this size have probably already encountered the need for this synchronization, and have created site-specific solutions.

Commands

The actual work of writing or changing such a script is simplified with the `nisaddent` utility. The `nisaddent` utility transfers data directly from `/etc` files and NIS `dbm` files into NIS+ tables, and vice versa.

The commands to create an NIS+ table from either an NIS `dbm` file or an ASCII source file are:

```
/usr/lib/nis/nisaddent –t table –y ypdomain [–Y map] [nisdomain]
/usr/lib/nis/nisaddent –f file type [nisdomain]
```

The command to dump formatted NIS+ data to standard output is:

```
/usr/lib/nis/nisaddent –d [–t table] type [nisdomain]
```

Approaches

Two approaches are possible. One option is to continue with a site-specific synchronization plan. Deciding what to synchronize can only be done by looking at the specific maps. Logins, passwords, and hostnames may need to be synchronized. Non-volatile data such as netmasks probably do not.

The cost of this option is that the synchronization scripts must be re-written, and also there is likely to be significant nightly clock-time involved if existing name data has to be downloaded and then updated in the two domains.

A second option would be to maintain both NIS and NIS+ services for every domain. Disadvantages of maintaining both services include the need for additional systems to run both SunOS 4.*x* and Solaris 2.*x* NIS+ servers in native-mode; extra work for the administrative staff; and the difficulty of troubleshooting in such an environment.

11. *Resolve Potential Hostname/Login Conflicts*

♦ **Identify and correct any potential conflicts between hostnames and logins.**

Every site needs to perform this step.

In the past, it was possible for both a host and a login to share an identical name (e.g., `mary@mary`). With the NIS+ service, this is not permitted. The planner should check to see if there are any names used as both host names and logins, and then correct one or the other so there is no duplication. This can most easily be done by renaming the host and entering the old host name as an alias in the `hosts` table. Add the host name to the `aliases.org_dir` table, using `nisaddent(1M)`.

In practice, even large organizations usually have only a very small percentage of names that conflict.

The reason for this restriction is that the NIS+ service tracks the network identify of principals using a fully qualified name for each principal. Principals can be either a user name or a host, which represents the `root` identity on that host. The NIS+ service has no way of distinguishing a host from a user if they both share the same name.

There is the potential for such conflicts with group names as well.

12. *Plan the Groups That Will Administer NIS+*

♦ **Decide how to delegate administrative control over NIS+. If you decide to create new administrative groups, list the names and members.**

One of the benefits of the NIS+ service is that it provides more fine-grained administrative control over tables and entries than was previously possible.

In general, Sun recommends that a new group, such as `nisadmin`, be created with the names of the administrators who have control over name service objects. The name (e.g., `nisadmin`) is not significant.

In small networks, one group may suffice. In large networks, the appropriate model may be multiple groups with permissions in separate domains — consider this model if a remote administrator may need to be included in the membership in case of emergencies.

If you have created this list, you will save time during conversion because you can consult your list.

13. *Review Domain's Directory Structure*

♦ **For each domain, review the structure of its standard directories, especially ownership and permission, to determine if the default values are appropriate for your site.**

Consider the appropriate administrative support boundaries, and plan to check the directory ownership. By default, directories are owned by the identity which creates them. This is usually `root` on the master server, which is appropriate. An extra step is required to set group access rights to tables for any new administrative groups you create.

The default directory structure is fine for most sites, which won't have a need for additional directories beyond `org_dir` and `groups_dir` until NIS+ applications — both internal and from third parties — become more widely available. The Distributed Objects Everywhere (DOE) environment also uses its own NIS+ directories.

Keep in mind the access requirements for support personnel associated with each domain's parent domain and all its child domains. In emergencies, they may require the ability to make changes, too — be sure they are as knowledgable about NIS+ as are the primary personnel.

You may wish to write up a separate worksheet if you need to for multiple or highly customized domains. An optional worksheet for defining domains is included in the appendix of *All About Administering NIS+*.

14. *Review the Standard Tables*

♦ **Review the standard tables to determine if the default definitions are appropriate — for most networks, they will be.**

The default definitions are appropriate for most sites: 17 standard tables, with Read permission to Owner, Group, and World; and Modify, Create, and Destroy permissions to Owner and Group. If you use NIS-compatibility mode for NIS clients, use the `nissetup -Y` option: tables created then are also readable by the unauthenticated principal Nobody.

Even if it turns out that the defaults are not appropriate for your site, you can easily go back later and change the owner, group, and permissions with the `nischown(1)` and `nischmod(1)` commands.

15. *Plan for Site-Specific NIS Maps or Applications*

♦ **If you maintain any site-specific NIS maps, list them and plan to convert them to NIS+ tables — use the Table Planning Worksheet to plan the site-specific tables. If you have site-specific applications that rely on NIS, list them and plan for their transition.**

If your site has site-specific NIS maps, you should fill out an NIS Table Planning Form for them (shown at the end of this article). This will help you plan out the table structure and permissions, and will save you time during the conversion.

Using the Table Planning Worksheet and the list of members in the `nisadmin` group, list each table's columns and attributes (searchable, case-insensitive, encrypted, binary, XDR, and time-to-live).

List any table paths or line/column separators and the source and frequency of external updates or synchronization for tables where that applies. After the structure of the table and their columns have been defined, add the permissions: owner, NIS+ group, and owner and group access rights to each table and column object.

During conversion, new site-specific tables must be created in the domain, and the contents loaded into the NIS+ map using a utility such as `nistbladm(1M)`.

When building tables for site-specific applications, remember that every searchable field you add to the table results in overhead. Adding a searchable field also results in the creation of an associated hash table. By the time you have three or four searchable fields, the hash tables will be larger than the size of the data itself, which is inefficient.

16. *Plan for DNS Clients*

♦ **If your site uses DNS, determine the DNS clients and decide how best to continue service to them.**

Organizations running DNS may need to alter the configuration of existing DNS clients. This is because current NIS servers and NIS+ differ in the manner in which they forward DNS requests. NIS servers will forward these requests, but NIS+ servers do not do so currently.

Therefore, systems running SunOS 4.*x* expect DNS requests to be handled by the NIS server. Because the NIS+ server does not do so currently, these client systems must have a resolver configuration file, `/etc/resolv.conf`, and the DNS resolver libraries installed on them (the "`Shlib_Custom`" software category) prior to disabling the network's NIS server and converting them to NIS+ servers running in NIS-compatibility mode. In the case of many clients, this can potentially be a lot of client-side configuration work, because the client conversion scripts need to run simultaneously with the conversion from an NIS master to an NIS+ master.

This client-side configuration is not necessary for DNS client systems running Solaris 2.*x*, because they process their own DNS requests directly. Solaris 2.*x* clients can be configured once initially as they are installed or converted.

17. Change '`.`' to '_' in Existing NIS Maps

♦ **If any existing NIS maps contain the character '`.`', plan to rename them at conversion time to use the character '`_`'. For example, convert automount maps of the form "`auto.*`" to "`auto_*`" names. Also check the content of the maps.**

Skip this step if your site does not currently maintain NIS automount maps, or maps containing the character '`.`' .

New Solaris 2.*x* clients should be initially configured using the new `auto_*` names.

18. Create Client Conversion Scripts

♦ **Create and test any necessary client conversion scripts.**

Creating and testing a conversion script will save time during the conversion. As a result of the previous steps, you may have identified configuration changes needed on client systems. These changes could include modifications to the domainname, editing the `sendmail` configuration, changing the DNS configuration, altering the Name Service Switch file (`/etc/nsswitch.conf`), adding any site-specific NIS+ environment variables, and automatically rebooting the system.

There is a way to reduce the amount of work associated with this task at conversion time, if your site has a shared, mounted directory of executables similar to `/usr/local`. If you do, put the client conversion script in that directory, and on the day of conversion, send mail to the clients telling them that just prior to going home for the day, they should run the script as `root` and then power off their systems.

19. Convert Site-Specific NIS Administrative Scripts

♦ **Create or convert any site-specific scripts you use to administer name servers.**

This step doesn't apply if you don't have special scripts to administer NIS maps. For example, some sites have a special script to add a user or an account within NIS. If so, then that script needs to be looked at. Remember that the `nistbladm` utility can convert back and forth between NIS+ tables and NIS ASCII files and `dbm` databases. Your modifications are likely to include a few new lines of NIS+ commands.

Consider how to administer large, volatile tables such as `hosts` and `passwd`. The `nistbladm` utility alone may be inappropriate. The Administration Tool's Database Manager has some limitations: it won't handle site-specific tables, and can be a performance bottleneck for very large tables. Examine any existing NIS scripts and consider rewriting them for NIS+.

20. Communicate Your Plan

♦ **Communicate your plan and schedule to users and other administrators who will be affected.**

Communicate a detailed plan well in advance of any changes to the name service environment, including a description, rationale, and schedule of changes. Explain the benefits of the new NIS+ environment, and also the need for improved and possibly more restrictive security measures.

Determine how much training system administrators and users will need, and develop a training plan.

A seminar for end users should include: new security features and their limitations; how to change passwords; how to change credentials; the use of common NIS+ commands; how to maneuver in the name service hierarchy; and if appropriate, how to access hosts and name services in other domains.

Conversion Tasks

21. Verify Pre-Conversion Tasks

♦ **Verify that all pre-conversion tasks are complete.**

Initial Shakedown

22. Small Initial Shakedown Phase

♦ **Start with a small shakedown test — identify 10-15 clients, and a system to be the master.**

This step is highly recommended. You will catch errors you've missed, and you will gain proficiency at NIS+ administration. The NIS+ service is sufficiently complex that everyone makes a mistake sooner or later in the first domain they build. Explaining mistakes to a small test group is easier than to the large audience of all your users.

You really want all the system administrators to be knowledgable and trained on NIS+ before you convert.

When planning the shakedown and conversion, if the plan results in your having to run NIS+ and NIS in parallel within a domain for a long time, you should consider the alternative of restructuring the domain in two or more pieces.

23. Convert Test Domain

♦ **Follow the conversion steps:**

i. **Build a root domain, or establish a sub-domain if a parent domain exists.**

ii. **Create a new directory, unless you are creating a root domain.**

iii. **Create your tables — if clients will be using NIS-compatibility mode, use `nissetup -Y`. Transfer existing name data into them.**

iv. **Create any site-specific tables and transfer the data.**

v. **Add credentials for all replicas in the domain.**

vi. **If you are starting a group to administer NIS+, then create the group and populate its membership. Remember, members must have previously defined NIS+ principal names and credentials.**

vii. **Start replica servers and force the initial replications.**

viii. **If you choose to initially maintain complete credentials for all principals, then add DES credentials for the `root` identity for each client, and LOCAL and DES credentials for the identity of the principals.**

ix. **Convert the test clients to NIS+ by editing the Name Service Switch files and possibly the domainname.**

x. **Verify from the client side that the name service is working correctly.**

xi. **If the e-mail environment has changed, populate the `sendmailvars` table with entries.**

24. Communicate Status

♦ **Communicate the status to users and monitor operations.**

Production

25. Convert Remaining Name Servers

♦ **Convert the remaining designated name servers to NIS+.**

The amount of time associated with the actual conversion effort can vary a great deal, depending on the extent to which these factors are applicable at your site:

- Ordering and installing new systems
- Moving network services from old servers to new ones
- Recommissioning old NIS servers
- Installing NIS+ on new servers
- Installing SunOS 4.1.*x* binaries
- Converting other NIS maps into tables
- If there are other NIS domains, either finalizing the synchronization system or turning it off

Don’t assume that you can do it all yourself in one night or over one weekend. Perhaps you can, but if you underestimate the task, you will have very unhappy users if you only get half the job completed.

Most domains are small and can do a conversion in off-hours. Large sites need to plan the actual conversion more carefully. Spreading the conversion over several days may help; for example, some large domains with multiple buildings may plan to convert from NIS to NIS+ one building per weekend.

26. *Convert Remaining Clients*

♦ **Convert the remaining clients in the old domain to work properly in the new domain.**

The steps involved are site-dependent.

<u>For Solaris 2.*x* clients:</u>

- If the domainname changes, edit `/etc/defaultdomain`
- If switching from NIS to NIS+, edit `/etc/nsswitch.conf`
- If switching to NIS+, initialize client with `nisinit -c`
- If a DNS client, edit `/etc/resolv.conf` and `/etc/nsswitch.conf`
- If your policy is to maintain full credentials, add entries for client and user credentials
- Reboot client

<u>For Solaris 1.*x* clients:</u>

- If the domainname changes, edit `/etc/defaultdomain`
- If a DNS client, add `/etc/resolv.conf` and the resolver library
- Reboot client

Frequently Asked Questions and Answers

Design of the NIS+ Service

Q: How long will the NIS-compatibility mode feature remain in the Solaris distribution?

A: A long time: protocols live forever. Once they have been installed, the systems that use a particular protocol revision such as NIS comprise a significant investment.

Q: Why does a NIS+ server have to be a client in its parent's domain or a domain above?

A: The NIS+ service is based on two fundamental assumptions that ensure all its dynamic service location algorithms will work correctly:

a. Every client knows how to get to its server.

b. A server must serve a domain lower than the one of which it is a client.

Together, these ensure that any client can locate any server in a secure fashion, knowing initially only its own server. As a by-product of these assumptions, a server must be in its parent's domain.

The assumptions in DNS are somewhat different in that clients also know the root server.

The one set of servers that can be in the domain that they serve are the root master and its replicas.

Q: Can one system be a server for a higher domain and a replica for a lower one?

A: Yes. All domains can be server at any level below.

Q: Can root servers be replicated?

A: Yes.

Q: Is it possible for administrators to restrict the servers that will receive an NIS+ `ping` operation?

A: Not at this time. In the future, perhaps multicast addressing would allow administrators to set up groups that would define a "radius" for a `ping` operation.

Troubleshooting and Diagnosis

Q: Can a replica become a master server in the event there is a problem with the master server?

A: Yes. The administrator can change a replica to a master server with one command: `nismkdir(1)`.

Q: How can an administrator tell which server responded to a request?

A: Currently this is not easy. Future software from SunSoft may make this easier by supplying a trace facility. For the time being, if you use the `-M` option of a command, such as:

```
niscat -M passwd.org_dir
```

or

```
nisls -M passwd.org_dir
```

then your request will be sent to the master server only. This guarantees the most up-to-date information.

Another method is to use the `nisshowcache(1M)` utility to examine the servers in the client-side directory object cache. Every entry is a candidate for a request; however, once a process uses the NIS+ client library to begin sending and receiving requests for a directory to a server, it will (as an optimization) continue to correspond with that server.

Yet another method is to use the `snoop` utility on the client — see the man pages for `snoop(1M)`.

Q: In an emergency, how do I tell clients not to use a bad server?

A: Simply use the `nisrmdir -s` command to disassociate the bad replica server from the list of valid replicas for that directory. The client should also get the new directory object from the master — you can either issue the command:

```
% nisinit -c -H master_server
```

or simply wait for the Time-To-Live of the directory object to expire.

Q: How do I diagnose the NIS+ service, particularly if the `niscat` command just hangs?

A: Currently, if the `niscat` utility just hangs, the administrator doesn't have a good way of debugging the fault. Future software in the Solaris distribution will improve the capability to diagnose such problems.

Q: How should an administrator fix a corrupt server?

A: Fixing a corrupt server involves, among other steps, cleaning out the affected directory in `/var/nis` on the replica. If it becomes necessary to do this, be sure that the replica itself is not a master for another server. Currently, there is no easy way of determining which directories a particular server is master of; future NIS+ releases should make this information more readily available. For now, you should carefully document your current configuration, and use this documentation as reference for future problems.

If a replica is corrupt, disassociate as a valid replica using the `nisrmdir -s` and `nisinit` commands (as described above); kill the replica's NIS+ daemon; carefully clean out the affected directory in `/var/nis` on the replica; restart the NIS+ daemon and add it back as a valid replica using `nismkdir -s` on the master server.

If a master server is corrupt, then make one of the replicas the master using the `nismkdir -m` command; kill the former master's NIS+ daemon; carefully clean out the affected directory in `/var/nis`; restart the NIS+ daemon; associate it with the new master as a replica using `nismkdir -s`; and when it is in synchronization again, reinstate it as the master using `nismkdir -m`.

If a root master server becomes corrupt, it is a lot of work for the administrator. There is not currently much choice, other than shutting the service down and recreating the directory from backup tapes.

Q: How can I change the system which is the root master?

A: This is currently fairly difficult. The steps that will work are:

i. **Use `nismkdir -m` to make a replica the new master.**
ii. **Kill `rpc.nisd` on the old master and the new master.**
iii. **Copy `root.object` from `/var/nis/`*hostname* of the old master to `/var/nis/`*hostname* of the new master.**
iv. **Restart the new master with the `rpc.nisd -r` flag.**
v. **Delete the `root.object` from the old master and restart `rpc.nisd` without the `-r` flag on the old master.**

Security

Q: Does the Solaris software support RSA authentication?

A: Not currently. In late 1992, Sun and RSA Data Security, Inc. made a technology announcement that future releases of the Solaris environment would include support for RSA authentication.

Q: Does the Solaris password aging feature age public keys?

A: No. The password aging feature in Solaris 2.1 applies only to local passwords located in the `/etc/passwd` file. Theoretically, keys have stronger security guarantees and do not need to be changed as often as passwords.

Other

Q: Will an Internet Request For Comments (RFC) be submitted for the NIS+ service?

A: Yes. SunSoft plans to submit an RFC regarding the NIS+ service.

Q: What is the difference between a concatenation path and a search path?

A: A search path indicates in which domains the client library should request an entry. Search path can be controlled using the `NIS_PATH` environment variable.

A concatenation path is an entry within a table. When a table is being searched, if a concatenation path is set, it will cause additional tables to be searched automatically.

Q: Was the NIS+ service designed to eliminate DNS?

A: The NIS+ service was not designed to eliminate DNS, but it was designed so as not to preclude doing so in the future.

Table Planning Worksheet

Table name: ______________________________

(Optional) Type ______________________________

Owner ______________________________

Group ______________________________

Permissions rmcd rmcd rmcd rmcd Nobody Owner Group World

Time-to-live ______________________________

Search Path ______________________________

Source of data/
Updated from ______________________________

	Column 1	Column 2	Column 3	Column 4
Column name:				
Owner				
Group				
Permissions	rmcd rmcd rmcd rmcd	rmcd rmcd rmcd rmcd	rmcd rmcd rmcd rmcd	rmcd rmcd rmcd rmcd
Time-to-live				
Searchable?	S	S	S	S
Case-insensitive?	I	I	I	I
Encrypted?	C	C	C	C
Binary?	B	B	B	B
XDR-encoded?	X	X	X	X

(Optional) Linked to: ______________________________

Graphics 4

XIL Imaging Library

By – Bill Fleming and Daniel Wexler, SMCC

XIL is the industry's first foundation imaging library with open software, hardware, and interface specifications. XIL strengthens SunSoft's strategy to encourage OEMs and third-party graphics vendors to develop through open hardware and software interfaces. The library provides the core functionality needed by commercial as well as technical imaging applications. It defines how imaging and video functions, such as display, manipulation, compression, and decompression, are carried out. It also specifies the interface between XIL and hardware.

By standardizing on the XIL API, software application developers can avoid the time-consuming and costly porting process that was previously required. XIL makes this task unnecessary, enabling software developers to design products that will work easily with all Sun systems. As the foundation imaging library, XIL can provide the base for other APIs such as PIKS, SunIPLib, and XIE. By supporting these APIs, Sun provides a direct route for porting existing applications.

XIL's standardized hardware interface will simplify development of frame buffers and accelerators by OEMs and third-party graphics hardware vendors, and the standard programming interface will encourage applications vendors to develop products. For end users, XIL standardization means that a wider range of both hardware and software products for imaging applications will become available.

The industry is quickly entering the era of collaborative computing, and nearly every workgroup application now has or soon will have a key imaging or video component. XIL makes it easier for OEMs and third-party partners to produce the software and hardware needed to make the interactive enterprise a reality.

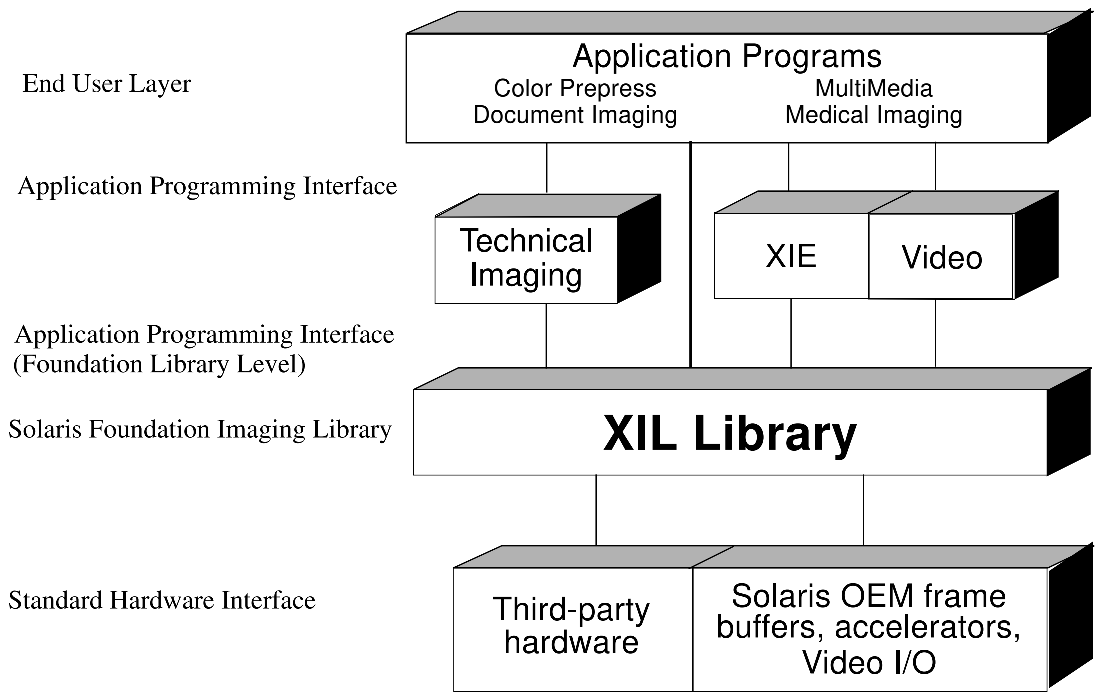


Figure 4-1 XIL Imaging Library

Product Highlights

XIL is the software layer for imaging applications, fitting between higher-level programming interfaces and system hardware. It contains three primary components: a programming interface specification for basic imaging functionality; a high-performance implementation of the specification, particularly tuned for standard workstation systems; and the hardware interface specification, which enables SunSoft and third-party imaging hardware developers to readily support XIL and applications built on top of it.

Additionally, XIL provides full 30-frames-per-second (FPS), 640 x 480 video windows on standard SPARCstation 2 workstations using advanced software video playback techniques developed by Sun. This is a significant achievement, since full-motion video normally requires a vast amount of file storage. Another advantage of XIL is its fast compression/decompression technology, including support for the JPEG standard. Reducing storage requirements by up to 98%, XIL eliminates the tremendous burden of storage for uncompressed image files.

The functionalities needed for commercial image processing are provided by XIL. Commercial imaging includes document imaging, graphics arts, and color prepress. These markets need arithmetic operations, geometric operations, sharping and deblurring, and color manipulations or translation.

The high performance of XIL saves time and money for software developers, since they no longer need to optimize their imaging functions for different workstation configurations. Software applications built on XIL will run on systems with the appropriate driver, providing more applications to hardware vendors and more hardware alternatives to software vendors without additional programming requirements. XIL users require Solaris 2.*x*, and SPARC developers need both Solaris 2.*x* and SPARCworks, a comprehensive set of development tools optimized for SPARC.

XIL will be used by application developers in a wide variety of commercial and technical markets, including document imaging, graphic arts, multimedia, color prepress, and technical imaging. It will also be a key interface used by Solaris OEMs manufacturers of frame buffers, video boards, and imaging accelerators.

Technical Features

As an object-oriented library, XIL hides the details of its implementation from the programmer. This enables the internals of XIL to be extended by Sun without changing the interface.

XIL is structured hierarchically through a series of abstract data types called *classes*. Applications can create instances of classes, where an instance of a class is an *object*. The class of an object determines the attributes it possesses and which operators can act on it. The class-defined hierachically below an existing one is a *subclass* (child) of the existing class (parent) class. Although XIL is class-based, an application can only define instances of classes (as objects), and cannot extend the hierarchy by creating new classes.

In XIL, an object is an abstraction of a graphics resource. Objects consist of state information (known as *attributes* in XIL) and a set of operators. Operators perform predefined actions on objects of their related class.

The most frequently used object in XIL is the *image*, a digital representation of some object. The library supports images of virtually any size, as well as of numerous data types. XIL supports 1-bit, 8-bit unsigned and 16-bit signed data. XIL allows the programmer to treat an X Window as a special destination image. This approach permits an operation to draw directly on the screen without requiring an intermediate copy.

Other object types include compressed image sequence objects, region-of-interest objects lookup table objects, convolution kernel objects, and dither mask objects. Each of these objects have their own sets of attributes and operations. For example, the image object contains the following attributes: width, height, number of bands (data elements per pixel), data type (precision of a single element), region of interest, origin, color space, gamut, colormap, synchronization flag, and display type.

Image operations can be separated into groups based on the required inputs. Functions operating on two images, or a single image and a constant, include arithmetic operators such as addition, subtraction, multiplication, division, and logical operators ‘and’, ‘or’, and ‘exclusive or’. XIL’s geometric functions provide image scaling, rotation, and translation capabilities, as well as a combination thereof, referred to as an *affine transform*. XIL will optimize a series of affine transformation commands by combining them into a single affine operation. Using the scaling functions, XIL can scale images by factors of up to 8-to-1, while preserving most of the information in the image. Additional functions include convolutions with arbitrary integer kernels, dilations, erosions, thresholding, and lookup table conversions.

Individual primitives are known as *atoms*, and combinations of primitives are known as *molecules*. For commonly used molecules, XIL contains specific code which has been optimized to accelerate calculations. A sequence of atoms is recognized as a pre-coded molecule automatically. This allows for future optimization of XIL though customer input, and in-house evaluation through the development of specialized molecules for particular applications.

Coding Example

The functions in the XIL library fall into two main categories: *image processing* and *digital video*. We will examine the process of developing an image-processing application using XIL. Typically, a developer will establish a connection between XIL and the windowing system (X), load and categorize the data, process the images, display the results, and finally close the library when exiting the program. Actually, using the library routines constitutes a relatively small portion of a typical image-processing program, while the difficult steps include creation of the correct type of window to display the image and the correlation of the different image data types with the window system display capabilities.

The C compiler acquires the XIL prototyping information from the header library, and therefore all XIL programs must contain the following line:

```
#include <xil/xil.h>
```

This header file defines a number of symbolic constants and enumeration constants, and contains the function prototypes for all of the functions in the XIL library. It also contains the following lines:

```
#include <X11/Xlib.h>
#include <X11/Xutil.h>
```

As a result, the developer need not include these files separately. This also demonstrates the close connection between XIL and the X windowing environment.

Opening and initialization of the XIL library is accomplished using the `xil_open()` command, which returns a *state* variable used in subsequent XIL calls. The state variable contains information including the error handling routine, the current status of XIL, and a history of what images have been created during the current session. After initialization, the next step is to connect XIL with the windowing system. Assuming we have created an X window, we simply request a *display image* from XIL using the following call:

```
Display *Xdisplay;
Window Xwindow;
XilSystemState XILstate;
XilImage DisplayImage;

DisplayImage = xil_create_window(XILstate, Xdisplay, Xwindow);
```

It is important to note that the XID of the window to be used should be the actual display portion of the window. For example, when using XView, the XID of the Canvas, not the Frame, should be passed to XIL.

To gain insight on the correct method for manipulating and displaying images, it is important to understand how XIL stores its data. An XIL image is a data structure containing many components defined using the four parameters to the `xil_create()` call: width, height, number of bands, and data type. While width and height are intuitively obvious, the concept of bands and their data types is less clear. The number of bands is the number of values needed to describe a single pixel; for example, in a grayscale image, one value describes the graylevel of the pixel, so the image is said to have one band. In an RGB image, three values (red, green, and blue) are required to describe a single pixel, so such an image has three bands. Each band in a multi-band image must have the same width and height. The XIL library supports three data types: 1-bit data (`XIL_BIT`), 8-bit unsigned data (`XIL_BYTE`), and 16-bit signed data (`XIL_SHORT`).

XIL can process images in any combination of supported data types and number of bands; however, current versions of the X windowing system only support 1-, 8-, and 24-bit frame buffers, so there are some restrictions on the types of images which can be display images. The only possibilities are 1-band, 1-bit images; 1-band, 8-bit images; and 3-band, 8-bit images. Therefore, it is good practice to keep a copy of the image in memory for image processing operations, and then to create a special version of the image cast into the most appropriate display format for visualization on the workstation. XIL contains functions for inquiring an image's attributes, and once the information is known, a simple `xil_copy()` can be used to move the data from the stored image into the display image. However, as stated above, it is also possible to use the display image as the storage for image calculations, although the supported data types are limited in this case.

Before exiting, all XIL programs should deallocate the memory associated with any XIL data structures not previously destroyed. Because the system-state data structure keeps a list of all the data structures XIL has created (and not destroyed), you can destroy all of these data structures by calling `xil_close()`, which takes the system state as its only parameter.

Software Release Versions 5

Current Sun Software Products and Release Versions

The following tables contain lists of Sun software products and their respective release versions, current as of February 28, 1993. These tables appear monthly in the STB for your convenience.

Please contact your sales representative if you would like to update the release level of a Sun software product you already use, or wish to purchase another product. Use the tables to determine whether your release is the most current.

Please note the following software product change(s):

c. As of the release of Solaris 2.0, the C compiler will be unbundled.

Operating Systems

Product Name	Current Release
SunOS (Sun-4 architectures)	5.1 (Solaris 2.1)
SunOS (Sun-3 architectures)	4.1.1_U1

Window Systems & Desktop Utilities

Product Name	Current Release
OpenWindows	Version 3
SearchIt for Solaris 1.*x*	1.0
ShowMe	1.0

System Management — High-Availability

Product Name	Current Release
Online: Backup Copilot for Solaris 1.*x*	1.0
Online: DiskSuite for Solaris 1.*x*	1.0
Online: IPI Dual Port Tools	1.0

System Management — Security

Product Name	Current Release
Sun SHIELD ARM and ASET	1.0

System Management — Network Management

Product Name	Current Release
SunNet Manager	2.0

System Management — Performance

Product Name	Current Release
Sun Database Excelerator (SunDBE)	1.3

Graphics Software

Product Name	Current Release
XGL Graphics Library	3.0
SunPHIGS	2.0
SunPEX	1.0
Direct Xlib	1.0

Software product availability is current for U.S. releases.
Availability may be delayed outside the U.S.

Software Development

Product Name	Current Release
Teamware	1.0
SPARCworks	2.0.1
SPARCompiler C *	2.0.1
SPARCompiler C++ *	3.0.1
SPARCompiler FORTRAN *	2.0.1
SPARCompiler Pascal *	3.0.1
Sun Ada Development Environment	1.1
Sun COBOL	1.0
Sun COBOL Runtime System	1.0
Sun COBOL Forms-2	1.0
OpenWindows Developer's Guide (DevGuide)	3.0.1
ONC / RPC Application Toolkit	1.0
NetISAM	1.0

* Available in both SPARCworks Professional and SPARCompiler versions

Other Products

Product Name	Current Release
NeWSprint	2.1
VideoPix	1.0

Software product availability is current for U.S. releases.
Availability may be delayed outside the U.S.

Connectivity Products

Product Name	Current Release
SunLink CG3270	7.1
SunLink SNA3270	7.0.1
SunLink BSC3270	6.1
SunLink BSCRJE	6.0
SunLink SNA Peer-to-Peer	7.0.1
SunLink DNI	7.0.1
SunLink TE320	7.0
SunLink OSI	7.0.1
SunNet MHS	7.0
SunLink IR	6.0
SunLink X.25	7.0
SunLink TRI/S	2.0
SunLink HSI/S	1.1
Serial Parallel Controller	1.1
SunLink FDDI/S	1.0
SunLink MCP	6.0
SunLink FDDI/DX	2.0
NetWare SunLink	1.0
SunLink Frame Relay	1.0
SunLink PPP	1.0
SunLink Local 3270	6.1.1
SunLink Channel Gateway	7.1
PC-NFS	4.0
PC-NFS LifeLine	2.0
PC-NFS Advanced Telnet	1.0
PC-NFS Programmer's Toolkit	4.0
SunPC	3.0

Software product availability is current for U.S. releases.
Availability may be delayed outside the U.S.

Updated Sun Operating System Support Matrix

When a platform has been designated as a final release, Sun's policy is to provide hardware and software support for out-of-production or transitioned systems and the latest compatible software for those systems for a minimum period of five years from the last order date.

As an ongoing policy, the level of support for SPARC products will generally decline at an increasing rate as new operating system releases are made available. Typically, the two most current versions of the operating system will be supported at level 1.

The details of Sun's software support policy have been published in a Sun Operating System Support matrix in past STB issues. As illustrated in the support matrix, older versions of the operating system generally receive lower levels of software support than newer versions. Thus, we recommend that you upgrade your system to the most current versions to ensure you receive the best quality of support from Sun.

Since we continually enhance the operating system to meet our customers' needs, we promised to keep you abreast of updates to the support matrix via the STB. Now that SunOS 4.1.3 and Solaris 2.0 are available, we provide an updated matrix.

Sun uses "support levels" to describe its software support policy. Please note that the services described in these levels are subject to local availability—contact your local service manager should you have any questions.

Level 1 Support
Level 1 offers the complete array of services contained in Sun's Personal AnswerLine Service, AnswerLine Service, and Software Update Service (see Note ①). Level 1 support also includes the full availability of options such as SunStart™, Sun's educational services, and supplementary consulting products and services.

Level 2 Support
Level 2 includes Sun's Personal AnswerLine and AnswerLine levels of phone support, limited availability of software and documentation, bug report submission and tracking, and emergency fixes and workarounds (see Note ①). Level 2 support also includes availability of supplementary consulting products and services.

Level 3 Support
Level 3 provides Time and Materials (T&M) support only.

Note ① — Software updates will only be available for the most current release. When a version of software has been designated as the final release for a platform, only critical bug fix releases will be available for that platform.

Sun Operating System Software Support Matrix

The following tables illustrate Sun's OS software support matrix. See Note ② below regarding this section.

For Sun-2	3.5	4.0/4.0.1	4.0.3	(see Note ③)
1/91-1/92	level 3	level 3	level 2	
1/92-1/94	level 3	level 3	level 3	

For Sun-3	3.5	4.0/4.0.1	4.0.3/4.0.3e	4.1	4.1.1	(see Note ③)
1/91-1/92	level 3	level 3	level 2	level 1	level 1	
1/92-1/95	level 3	level 3	level 3	level 2	level 2	
1/95-1/96	level 3	level 3	level 3	level 3	level 3	

For Sun386i	4.0.1i	4.0.2i	(see Note ③)
1/91-1/92	level 2	level 1	
1/92-1/95	level 3	level 2	
1/95-1/96	level 3	level 3	

For SPARC	3.5	4.0/4.0.1	4.0.3/4.0.3e	4.1	4.1.1	4.1.2	Solaris 2.0	4.1.3
1/91-present	level 3	level 3	level 3	level 2	level 2	level 1	level 1	level 1

Note ② — The above levels 1 and 2 are for contract customers only. Non-contract customers have the option to purchase support on a Time and Materials basis or use Sun's consulting services. The then-current unbundled products will be supported at the same level as the customer's operating system. Support for emergency fixes and workarounds may require the customer to upgrade to a more current version of the product.

Note ③ — These will be the final releases for these product lines, with the exception of critical bug fix releases. Sun's future SVR4-based operating system will function primarily on SPARC processor-based systems (Intel-based support for SVR4 is available with SunSoft's INTERACTIVE UNIX product line). Interoperability between Sun-3 or Sun386i systems and SPARC SVR4 systems will be fully supported through services such as NFS, NIS, remote login, and the like.

In addition, Sun is committed to supporting Sun-3 diskless clients running SunOS 4.1.1 on Solaris 2.0 servers.

Notes & Comments 6

World Hotlines

Sun Customers throughout the world have service hotlines available for both software and hardware support questions. The service hotlines are shown below. If your country is not shown in the table, please phone your local Sun sales office. World hotlines are divided into those for the U.S.A., Europe, and Intercon.

United States

All	**1-800-USA-4SUN** (**1-800-872-4786**)

Europe

Finland	Sun Microsystems Oy	**+358-0-5255-6300**
France	Sun Microsystems France SA	**+33-1-30-67-50-80**
Germany	Sun Microsystems GmbH	**+49-89-46008-321**
Italy	Sun Microsystems Italia SpA	**+39-6055228**
Netherlands	Sun Microsystems Nederland BV	**+33-501234**
Spain	Sun Microsystems Iberica SA	**+34-1-5560243**
Sweden	Sun Microsystems AB	**+46-8-623-9100**
Switzerland	Sun Microsystems (Schweiz) AG	**+41-1-825-7272**
	Sun Microsystems (Suisse) SA	**+41-22-364-3639**
United Kingdom		
	Sun Microsystems UK Ltd	**+44-276-691974**

Intercon

Canada:

	Quebec District	
	Montreal, Quebec	**(514) 744-9240**, **(800) 361-1554**
	Ottawa District	
	Ottawa, Eastern Canada	**(613) 723-8112**, **(800) 465-4415**
	Central District	
	Toronto, Central Ontario	**(416) 490-9490**, **(800) 263-1680**
	Prairies District	
	Manitoba, Saskatchewan, Alberta, Calgary, Western Canada	**(403) 262-6722**, **(800) 661-9256**
	Edmonton District	
	Edmonton	**(403) 421-7264**, **(800) 661-3643**
	B. C. District	
	Vancouver, British Columbia	**(604) 684-4120**, **(800) 663-0440**
	National CSS/Education	**(416) 477-6745**
Australia	Sun Australia	**(008) 02-4417**
New Zealand	Sun New Zealand	**(0800) 446172**
Hong Kong	Sun Asia	**852-802-4188**
Japan	Nihon Sun	**044-819-3121**

Latin America: Contact the Latin American hotline at ICON headquarters:
U.S. Office **(415) 688-9160**

Within ***Brazil***, you may also contact:
Sun do Brasil **55-11-887-9011**

Caribbean: Call your local distributor.

All Other Countries

Call **(415) 688-9372**, which is a U.S. number.

Reporting Bugs

This article contains three sections for submitting bugs. The first section describes procedures to use within the United States and Canada. The second section describes procedures within Europe. The third section describes procedures within ICON.

Submitting Software Bugs: United States and Canada

This section contains information on reporting bugs within the U.S., for customers holding and not holding support contracts.

Sun's United States Solution Center accepts software bug reports from Sun users via electronic mail and by phone. The method you use to submit a bug report varies with your needs.

U.S. users holding support contracts can report bugs to US Solution Center via the **(800) USA-4-SUN** phone hotline. Canadian users holding support contracts should call their local support center. The U.S. Solution Center phone hotline is the fastest way for a customer to find out if a problem is known and if a workaround exists. The status of previously-reported bugs can also be obtained in this way.

U.S. customers holding support contracts can also submit bug reports electronically to the address *sun!hotline* (or *hotline@Sun.COM*). Canadian customers need to use *hotline@Canada.Sun.COM*. This method generates a service order, and can be used when lines of code or other information difficult to relay over the phone is needed to describe the bug.

Customers who do not hold Sun software support contracts can report bugs via electronic mail to the address *sun!sunbugs* (or *sunbugs@Sun.COM*). These reports are reviewed periodically to determine proper disposition. Those reports determined to be from supported customers are forwarded to the U.S. Solution Center for handling. Reports from customers who cannot be verified as holding a support contract are reviewed by Sun's engineering and support personnel. An internal bug report is generated if the reported bug is new and verifiable.

Finally, customers not holding software support contracts may call the **(800) USA-4-SUN** phone hotline to report a problem and request support on a Time and Materials (T&M) basis. Canadian customers should call their local support center. In this case, please have a Purchase Order (PO) number for billing purposes.

Summary: United States and Canada

For U.S. contract customers, **(800) USA-4-SUN** is the best method to report bugs. Canadian contract customers should report bugs to their local support center. The hotlines to use appear monthly in the note entitled "World Hotlines."

The electronic mail address *sun!hotline* (or *hotline@Sun.COM*) is available for U.S. customers to submit materials that are difficult to relay over the phone. Canadian customers should use *hotline@Canada.Sun.COM* .

For non-contract customers, the electronic mail address *sun!sunbugs* (or *sunbugs@Sun.COM*) is available to report bugs. To help us serve you better, please include the following information with all electronic mail reports:

- Your name
- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing

Submitting Software Bugs: Europe

This section contains information on reporting bugs within Europe, for customers holding and not holding support contracts.

Procedures for submitting bugs are similar to those used in the United States. All customers should use their local country Answer Center to report bugs, with contract customers receiving a specific follow-up.

Sun customers not holding software service contracts can call their local Answer Center, and will need to provide a Purchase Order (PO) number at the time of the call.

Summary: Europe

To help European service centers serve you better, please include the following information with all electronic mail reports:

- Your name
- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing

Finland

Report bugs to the Finland Answer Center at the postal address shown below.

Hotline	Hotline Telephone: +358-0-5255-6300
Sun Microsystems Oy	Telefax: +358-0-5255-6252
Sinikalliontie 12	
SF-02630 Espoo, Finland	E-mail Address: *Hotline@Finland.Sun.COM*

Special Dispatch Arrangements: Provide Dispatch with the Contract number.
Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

France

Report bugs to the France Answer Center at the postal address shown below.

Service Hot Line	Hotline Telephone: +33-1-30-67-50-80
Sun Microsystems France SA	Telefax: +33-1-30-67-53-70
13, Avenue Morane Saulnier	Minitel: 36 14 code SUNFRANCE
BP 62	
78142 Velizy, France	

Special Dispatch Arrangements: Please provide Dispatch with System serial number *or* Contract number.
Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis, and order this support at the above address.

Germany

Report bugs to the Germany Answer Center at the postal address shown below.

Hotline	Hotline Telephone: +49-89-46008-321
Sun Microsystems	Telex: 5218197 sun
Bretonischer Ring 3	Telefax: +49-89-46008-320
8011 Grasbrunn, Germany	E-mail Address:
	{*sunuk,unido*}*!sunde!hotline* or
	hotline@Germany.Sun.COM

Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Italy

Report bugs to the Italy Answer Center at the postal address shown below.

Hotline Sun Microsystems Italia SpA Centro Direzionale Colleoni Palazzo Andromeda (Ingresso 1) Via Paracelso 16 20041 Agrate Brianza Milan, Italy	Telephone: +039-6055228 Telex: 326423 MBOX SUNITA Telefax: +039-6056764

Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

The Netherlands

Report bugs to The Netherlands Answer Center at the postal address shown below.

Sun Microsystems Nederland BV Computerweg 1 3821 AA Amersfoort Netherlands	Hotline Telephone: +33-501234 Telefax: +33-553058 Telex: 79297 sun nl

Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Spain

Report bugs to Spain Answer Center at the postal address shown below.

Sun Microsystems Iberica SA Plaza Pablo Ruiz Picasso s/n Torre Picasso Planta 25 28020 Madrid, Spain	Hotline Telephone: +34-1-5560243 Telefax: +34-1-5564097

Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Sweden

Report bugs to the Sweden Answer Center at the postal address shown below.

Sun Microsystems AB
Box 51
S-164 94
Kista, Sweden

Hotline Telephone: +46-8-623-9100

E-mail Address: *hotline@Sun.se* or *hotline@Sweden.Sun.COM*

Arrangements for Non-Contract Customers:
Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Switzerland

Report bugs to the Switzerland Answer Center at one of the postal addresses shown below.

(*German*)
Sun Microsystems (Schweiz) AG
Postfach
Eschenstrasse 8
CH-8603 Schwerzenbach, Switzerland

Telephone: +41-1-825-7272
Telefax: +41-1-825-7222
E-mail: *hotline@Swiss.Sun.COM*

(*French*)
Sun Microsystems (Suisse) SA
Route des Avouillons 12
CH-1196 Gland, Switzerland

Telephone: +41-22-364-3639
Telefax: +41-22-364-1176
E-mail: *answerline@Swiss.Sun.COM*

Special Dispatch Arrangements: Provide Dispatch with the Contract number and System serial number.
Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

United Kingdom

Report bugs to the UK Answer Center at the postal address shown below.

Sun Microsystems UK Ltd
Watchmoor Park
Riverside Way
Camberley, Surrey
GU15 3YL, England

Hotline Telephone: 0276-691974
Telefax: 0276-681546

Special Dispatch Arrangements: Please provide Dispatch with either the System serial number *or* Contract number.
Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Submitting Software Bugs: Intercon

This section contains information on reporting bugs within Intercon, for customers holding and not holding support contracts.

Procedures for submitting bugs are similar to those used in the United States. All customers should use their local country Answer Center to report bugs, with contract customers receiving a specific follow-up.

Sun customers not holding software service contracts can call their local Answer Center, and will need to provide a Purchase Order (PO) number at the time of the call.

Summary: Intercon

To help Intercon service centers serve you better, please include the following information with all electronic mail reports:

- Your name
- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing

Detailed information for individual countries follows.

Australia

Report bugs to the Australian Answer Center at the postal address shown below.

Hotline
Sun Microsystems Australia Pty. Ltd.
PO Box 320
Artarmon, New South Wales 2064
Australia

Hotline Telephone: (008) 02-4417
Telefax: 02 436 1084
E-mail Address: *hotline@Aus.Sun.COM*

Special Dispatch Arrangements: Please provide Dispatch with either the System serial number *or* Contract number.
Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Japan

Report bugs to the Nihon Sun Answer Center at the postal address shown below.

Nihon Sun Microsystems K.K
KSP Technical Center - Answer Center
KSP R&D Bldg. B-1F-112
Sakato 100-1, Takatsu-ku,
Kawasaki, Kanagawa, 213
Japan

Hotline Telephone:
044-819-3121, 044-819-3111
FAX: 044-819-3120

Special Dispatch Arrangements: Provide Dispatch with the Contract number and Serial Number.

Arrangements for Non-Contract Customers: Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

New Zealand

Report bugs to the New Zealand Answer Center at the number shown below.

Hotline Telephone: (0800) 446172

Your STB Subscription

Your company's software service contract includes a monthly issue of the STB. Each month, the copy of your STB is mailed to your company's primary contact person or department named in the contract. Sites with more than one contract may receive more than one STB copy, depending on how the contracts are arranged.

Duplicating Your STB

Your primary contact person or department may duplicate this "master" STB copy for all Sun workstation end-users. As long as you duplicate copies and route them internally only, there are no copyright infringement problems.

This limited permission for duplication is for your convenience only, however, and does not include any duplication for resale, for distribution outside your company, or for distribution to employees of companies not having a Sun software service contract.

Multiple STB Subscriptions

You can now order extra copies of the STB for a low monthly fee for each additional copy. These extra copies will sent to the same address as your "master" STB copy. Please contact your local Sales/Service Office or Sun Authorized Distributor.

Changes or Updates

Changes of address or other updates, including change of contact name, should be submitted to your Customer Service Administrator (CSA), not to the STB Editor. Your CSA will register the requested changes through proper channels with Sun headquarters.

Further STB Questions

If you have any questions or comments regarding the STB, including corrections, please send your comments to *stb-editor@Sun.COM* or *sun!stb-editor*.

Index