

Rules of Engagement for the Support of Reduced or Minimal Configurations

A. Background

For over a decade, many of Sun's customers have deployed reduced or minimal Solaris Operating System (Solaris OS) configurations. While this has been done predominantly in the financial services and government/military sectors, this practice has become widespread in recent years with greater numbers of customers. These configurations are being used to promote heightened security under the premise that "software that is not installed cannot (easily) be re-enabled or exploited". In addition, some customers have indicated that their use of reduced or minimal Solaris OS configurations stems from their desire to also reduce the management burden and costs associated with patching and audit compliance.

In order to satisfy these customer requirements, reduced or minimal Solaris OS configurations are created and deployed throughout a customer's environment. In most cases, customers who have deployed such configurations have done so using supported, public and documented interfaces that Sun has delivered in the Solaris OS.

The goal of this document is to clarify Sun's position on the supportability of reduced or minimal configurations so that a clear expectation can be set (both for Sun organizations as well as for our customers). This document will set forth an acceptable procedure for building supported reduced or minimal configurations. Finally, it will also outline some of the risks and considerations that must be understood when using these types of configurations.

B. Supported Process for the Creation of Reduced/Minimal Configurations

The goal of this section is to document the process that must be used in order to create a supportable reduced or minimal Solaris OS configuration. To qualify as a supported configuration, this process must be followed.

1. Initial Installation

The Solaris OS system must be installed using one of the Sun provided (and supported) installation mechanisms (e.g., Graphical Installer, JumpStart, etc.) The initial installation of the system must be based upon one of the Sun provided (and supported) software installation clusters (i.e., metaclusters).

For the Solaris 10 OS, this would entail installing using one of the following metaclusters:

- Reduced Networking (SUNWC_{rnet})
- Core (SUNWC_{reg})
- End User (SUNWC_{user})
- Developer (SUNWC_{prog})
- Entire (SUNWC_{all})
- Entire + OEM (SUNWC_{Xall})

Note that the stated purpose of the Reduced Networking metacluster was to help customers who want to deploy more reduced or minimal Solaris OS configurations. The following statements were documented in the Sun Architectural Review case that introduced the Reduced Networking metacluster (PSARC/2002/254):

The Reduced Networking metacluster is intended to provide a supported foundation for the deployment and secure configuration of minimized systems.

This metacluster was introduced into the Solaris 10 OS as a direct result of customer feedback in the area of minimization. Additional Solaris OS software packages can be added to or removed from the selected metacluster provided the conditions defined in items B.3 and B.4 are satisfied.

2. Solaris OS Upgrade

The Solaris OS system must be upgraded using one of the Sun provided (and supported) upgrade mechanisms (e.g., JumpStart, Live Upgrade, etc.) It is supported to upgrade both reduced and minimal Solaris OS configurations using any of the Sun supported upgrade mechanisms. The Solaris OS upgrade functionality will analyze the current set of packages installed on a system, the list of packages derived from the `clustertoc(4)` file for the upgraded version of the Solaris OS (for the appropriate metacluster) as well as the package history file. Using this information, the upgrade functionality will install and/or remove packages as necessary to ensure that the resulting system configuration is left in a consistent and supported state.

Note – it may be necessary to install the Live Upgrade software as it is not present in some of the Solaris OS metaclusters (e.g., Reduced Networking).

3. Solaris OS Package Addition (Installation)

Any additional (i.e., not installed in the previous steps) Solaris OS packages that the customer would like added to the system must be installed using one of the Sun provided (and supported) installation mechanisms (e.g., Graphical Installer, JumpStart, `pkgadd(1M)`, etc.) If there exist any package dependencies on the software being installed, those dependencies must be satisfied prior to the installation of the package. To qualify as a supported configuration, all installed software dependencies must be satisfied.

Example: Assume that package A depends on package B and neither package is installed on a Solaris OS system. If a customer wants to install package A, the customer must first install package B, because package A depends on package B. This same process must be completed for any other software packages upon which package A depends. Forcible addition of package A (without package B being previously installed) would result in an unsupported configuration.

4. Solaris OS Package Removal

If there are any Solaris OS packages to be removed from the existing system configuration, they must be removed using a Sun provided (and supported) mechanism (e.g., JumpStart, `pkgrm(1M)`). If there exist any dependencies on the software package being removed, those dependencies must be addressed and satisfied prior to the removal of the package. To qualify as a supported configuration, the dependencies for all of the remaining installed software must be satisfied.

Example: Assume that package A depends on package B and that both packages are installed on a Solaris OS system. If a customer wants to remove package B, then the customer must first remove package A, because package A depends on package B. This same process must be completed for any software packages that depend on package B being installed. If the customer cannot remove a package that depends on package B, then package B cannot be removed. Forcible removal of package B would result in an unsupported configuration.

It is not permissible to remove any software packages that are found in the smallest Sun-provided installation cluster. For the Solaris OS versions 9 and newer, this means any packages that are delivered by the `SUNWcmreq1` metacluster. For earlier releases of the Solaris OS, this means any packages that are delivered in the `SUNWcreq` metacluster. Removal of packages delivered in the smallest Sun-provided installation cluster will result in an unsupported configuration.

It is prohibited to remove Solaris OS software packages using any means other than `pkgrm`. Files and directories delivered by Solaris OS packages must not be removed individually using commands such as `rm(1)`. Further, the `pkgchk(1M)` command must report no errors with respect to Solaris OS package accuracy and integrity after any installation or removal operation has completed.

¹ The `SUNWcmreq` metacluster, available starting in Solaris 9, is special. It is not listed as a choice or able to be selected when a system is installed. Its sole purpose is to contain those software packages that must be installed on every Solaris system regardless of its purpose or configuration.

C. Solaris OS Patching

Using a reduced or minimal Solaris OS configuration can have a beneficial impact on patching. According to research completed for Solaris 10, as of March 10th, 2006, there were 345 patches released for the SPARC platform version of the OS. Only 149 of those patches applied to the Reduced Networking metacluster. A savings of over 40% (in the best case) is extremely significant especially when applied to customers with large Solaris OS deployments. Even customers who take a reduced approach to Solaris OS configurations will benefit. Using the SUNWCuser metacluster as the foundation for a Solaris OS installation will still yield a patch savings of nearly 30% per system.

These benefits are not achieved without some measure of risk. A single Solaris OS patch can contain fixes for files that are delivered across multiple Solaris OS packages and metaclusters². When working with reduced or minimal Solaris OS configurations, this could mean that patches are partially applied since one or more of the packages to be patched may not be installed on the system.

When working with reduced or minimal configurations, it is not easily apparent which patches may be partially installed. Furthermore, if a package (originally not installed) is added after the patches had been applied, it will not have been patched - even though the system will declare that the patch had been successfully installed. This kind of inconsistency is not only an annoyance; it can furthermore contribute to a security vulnerability inadvertently being exposed due to the use of unpatched code.

To qualify as a supported configuration, one of the following methods must be used:

- Do not install any additional Solaris OS packages after a system has been patched
- Re-install the most recent version of each previously applied patch after the new packages have been installed

If neither of these options are acceptable, then a customer may want to make use the Entire + OEM software installation cluster to avoid this problem.

D. Recommendations, Caveats and Warnings

1. Additive Approach to Minimization

It is recommended that customers take an additive approach to building reduced or minimal configurations. That is, rather than start with the entire software distribution and attempt to remove packages that are not wanted, customers should strive to begin their installations using the smallest metacluster that most closely approximates their needs. Using this foundation, additional Solaris OS software packages can be added or removed as appropriate.

2. Modification of Installed Software Attributes

It is prohibited to change the ownership, group, or permissions of files delivered by Solaris OS software packages. If a customer believes that one of these attributes is set in error or can be improved, the customer should file a bug or a request for enhancement (RFE) report so that Sun can make an evaluation and take action accordingly. The `pkgchk (1M)` command must report no errors with respect to Solaris OS package accuracy and integrity.

3. Undocumented Software Dependencies

As with any complex system, it is not possible to evaluate every possibility. As a result, there may be software package dependencies that are not documented in the Solaris OS. Should a customer encounter such a problem, Sun will work with the customer to determine which software package or packages may be required. To satisfy the dependency, the customer must install any required Solaris OS software packages. Further, Sun will file a bug report so that this issue can be corrected a future version of the Solaris OS.

² As of the writing of this document, the Solaris 10 kernel patch (118833 for SPARC) contains fixes for code delivered in five different packages which themselves are delivered in three different metaclusters: SUNWCmreq (SUNWcakr, SUNWckr, SUNWcsu), SUNWCuser (SUNWmdbr, SUNWtoo) and SUNWCprog (SUNWhea).

4. Dynamic Software Dependencies

Some software packages may have dynamic or optional dependencies. This may be the case when a software package only requires another based on runtime settings defined by the user or those stored in a configuration file. In these cases, the software package may not list these dynamic dependencies. If the customer encounters this problem, Item D.3 above must be followed to determine which software packages are required. Any identified required packages must be installed.

5. Unbundled and Third Party Software Impact

The majority of software vendors (including Sun) do the majority of their testing using systems that have been installed using the complete set of Solaris OS software (e.g., `SUNWCXa11`). Testing is rarely completed using reduced or minimal configurations. As a result, there is a higher risk that software added post-installation will fail when using a reduced or minimal configuration.

Further, the actual list of Solaris OS software packages required for a given reduced or minimal configuration will very much depend on the entire suite of hardware devices and software functions and services that will be used on the target platform. As a result, there could be an impact to the Solaris OS configuration if additional unbundled or third party software packages are later added (or upgraded) to a system's configuration. Customers must always perform adequate testing to ensure that their reduced or minimal Solaris OS configurations perform as expected within their environment. This testing must be completed before placing such systems into production.

E. Clarification of Supported versus Qualified Configurations

If a Solaris OS system is installed, configured and managed using Sun provided (and supported) methods, Sun will support the customer should a problem arise. The same is true for customers deploying reduced or minimal configurations when the approach and requirements described above are followed.

To Sun, "support" means that should a customer encounter a problem:

1. Sun will work with the customer to determine the root cause of the problem and recommend a fix or workaround.
2. If there exists a bug in the Solaris OS, a bug report will be filed on behalf of the customer. Sun will work to correct the problem, and the customer will have the ability to escalate the issue to obtain a fix.
3. If it is determined that required software is missing from the customer's Solaris OS configuration, then the customer will be instructed to install that software and to determine if the problem still exists.
4. If it is determined that the supported process (described above) was not followed then the customer will be advised that they are running a non-supported configuration and given advice as to how to rectify the situation. While the actual advice will depend on the customer's situation, in extreme cases, the customer may be advised to completely reinstall the system according to the process described above.

Customers are advised that "supported" is not the same as "qualified." Solaris Engineering designs, implements and tests the Solaris OS using the `SUNWCXa11` metacluster. The explicit package dependencies included with Solaris are therefore not certified to be correct. The exact interdependencies in the software are not formally known, evaluated or documented in a reliable way. As a result, when building reduced or minimal configurations, it is possible that undocumented dependencies or unexpected behaviors may be discovered.

The Solaris OS is a complex software product with many software packages, components and settings. It is simply not possible to design or test for every permutation. In fact, the vast majority of Sun and ISV products are only tested on the entire distribution of Solaris. As a result, customers who choose to deploy reduced or minimal Solaris OS configurations must be advised that they are taking on additional risk and are strongly advised to fully exercise and test their configurations before promoting them into production use.

Those customers who are uncomfortable with that risk should be advised to install the complete Solaris OS distribution in a hardened configuration in order to reduce their exposure. Solaris OS hardening can be achieved using native tools, introduced by the Solaris Secure by Default³ project, starting with Solaris 10 Update 3. For current and earlier versions of the Solaris OS, the Solaris Security Toolkit⁴ should be used. Customers can also employ additional Solaris OS security controls (e.g., Role-based Access Control, IP Filter, etc.) to further control access to system services and functions.

That said, Sun will support any customer (assuming a valid support contract) who follows the process declared in this document. In this context, supported means that Sun will work with the customer to identify and resolve problems that are encountered. Qualified configurations on the other hand refers to those that Sun specifically has tested and are known to work. While reduced or minimal Solaris OS configurations are supported - they are typically not qualified – unless the customer leverages a process such as the High End Systems Customization program.

F. Exceptions

Currently, a complete list of exceptions is not available. The following list includes those known exceptions and others will be added as they are found.

1. Star Fire / Sun Fire System Controllers. By default, Solaris OS minimization is not supported on any Star Fire or Sun Fire platform that has a System Controller capable of running the Solaris OS. Customers interested in deploying supported minimal configurations should consult the High End Systems Customization program.

G. Definitions

Hardened Configuration. A hardened configuration is a Solaris OS configuration that has been tuned for enhanced security. Very often this includes disabling those system and network services deemed unnecessary for the intended purpose of the system, enabling non-default security features or capabilities, and enabling or enhancing access control policies, auditing and logging.

Minimal Installation. A minimal installation is a special case of a reduced installation whereby the only software packages that are installed (or remain) are those that directly contribute to the operational and management requirements of the system. All unnecessary software is removed (or not installed in the first place). Minimal installations are highly dependent on the actual hardware and software configuration being used (including any Sun and third party software and devices).

Reduced Installation. A reduced installation (or reduced software installation) is a Solaris OS configuration created by installing fewer than all of the software packages delivered by the Solaris OS. This includes installations that are based on any software installation cluster that is not SUNWCXa11.

³ For more information on the Solaris Secure by Default project, see: <http://www.opensolaris.org/os/community/security/projects/sbd/>

⁴ For more information on the Solaris Security Toolkit, see: <http://www.sun.com/security/jass/>

Appendix A. Contributors

The following people have contributed to the creation and/or review of this document:

<i>Contributor</i>	<i>Organization</i>
Belfer, Warren	Marketing, Web Platform Engineering
Brunette, Glenn	Global Sales and Service, CTO (Customer Champion)
Carlson, James	Software, OPG, KISS
Di Pol, Joe	Software, Java Enterprise System
Elling, Richard	Servers, Performance and Availability
Foxwell, Harry	Global Sales and Service, OS Ambassador
Gerhard, Chris	Global Sales and Service, PTS
Hall, Jim	Global Sales and Service
Igouchkine, Vasya	Software, OPG, KISS
Jenks, Kathy	Software, OPG, Security (Engineering Champion)
Kriz, Robert	Global Sales and Service, Security Ambassador
Laurent, Jim	Global Sales and Service, OS Ambassador Board Member
Memishian, Peter	Software, OPG, KISS
Miner, Dave	Software, OPG, KISS
Moffat, Darren	Software, OPG, Security
Rotondo, Scott	Software, OPG, Security
Rozenfeld, Isaac	Global Sales and Service
Shameem, Jafar	Global Sales and Service
Smaalders, Bart	Software, OPG, Kernel
Thacker, Mark	Solaris Product Marketing, Security
Vaidun, Anand	Global Sales and Service (Support Champion)
Williams, Nicolas	Software, OPG, KISS
Winiger, Gary	Software, OPG, Security

Appendix B. Bugs and RFEs

The following is a listing of bugs and RFEs filed on behalf of this effort. All of these bugs or RFEs have been tagged with the *minimal-support* keyword. The goal of these bugs and RFEs is to improve the overall customer experience when deploying minimal configurations as well as to improve the ability of our support organization to detect reduced or minimal configurations and determine whether the approach outlined in this document has been properly followed.

<i>Change Request ID</i>	<i>Description</i>
6438671	*pkgrm* should not let users remove software from SUNWcmreq
6438674	*patchadd* should record the fact that a package is not installed
6438680	*pkgchk* should be able to detect partially patched package conditions
6438687	*pkgchk* should be able to report on missing package dependencies