



Solaris™ Patch Manager Base Version 1.0 Strategy White Paper

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

January 2004

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, SunOS, SunSolve, Java, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunOS, SunSolve, Java, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'INTERFACE D'UTILISATION GRAPHIQUE OPEN LOOK ET SUN™ A ÉTÉ DÉVELOPPÉE PAR SUN MICROSYSTEMS, INC. POUR SES UTILISATEURS ET LICENCIÉS. SUN RECONNAÎT LES EFFORTS DE PIONNIERS DE XEROX POUR LA RECHERCHE ET LE DÉVELOPPEMENT DU CONCEPT DES INTERFACES D'UTILISATION VISUELLE OU GRAPHIQUE POUR L'INDUSTRIE DE L'INFORMATIQUE. SUN DÉTIENT UNE LICENCE NON EXCLUSIVE DE XEROX SUR L'INTERFACE D'UTILISATION GRAPHIQUE XEROX, CETTE LICENCE COUVRANT ÉGALEMENT LES LICENCIÉS DE SUN QUI METTENT EN PLACE L'INTERFACE D'UTILISATION GRAPHIQUE OPEN LOOK ET QUI EN OUTRE SE CONFORMENT AUX LICENCES ÉCRITES DE SUN.

CETTE PUBLICATION EST FOURNIE 'EN L'ÉTAT' ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPOUDRE A UNE UTILISATION PARTICULIÈRE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Contents

- Executive Summary1**
- PatchPro Overview2**
 - PatchPro Database2**
 - PatchPro Properties3**
- Server and Client Configuration4**
 - Setting Up a Local Repository of Patches4**
 - Patch Depot Server4**
 - Read-Only File System5**
 - Synchronization With SunSolve Patch Portal5**
 - Setting Up a Patch Client5**
 - Editing the patchpro.conf File6**
 - File Aging6**
 - Locating Certificates7**
 - Avoiding Using Current Files7**
 - Proxy Configuration8**
 - Capturing the Patch List8**
- Summary: How to Populate the Local Patch Depot and
Create Client-Specific Patch Sets8**
- Additional Features10**
- Appendix A: PatchPro v2.211**
- Appendix B: Patch Lists13**

Executive Summary

Patch Manager is becoming Sun's standard patch management tool. This tool consists of 1) the Solaris™ Patch Manager Base Version 1.0 application installed on the customer's system and 2) a supporting infrastructure that resides at Sun. This infrastructure consists of a patch knowledge database, host analysis modules, digitally signed patches, and a set of internal processes to ensure that the components work together. Core to the Solaris Patch Manager Base Version 1.0 application is the PatchPro Analysis Engine.

The PatchPro Analysis Engine provides a generic mechanism for analyzing hardware and software configurations. It first constructs a "host object" (a representation of the system being analyzed), then uses a set of detection algorithms to identify the patches in the Patch Database that are applicable to the host. Both the Patch Database and the "detectors" are obtained from Sun during the analysis to ensure that the most current information is used. The analysis itself, however, is performed on the host system. No information about the host is transmitted to Sun.

Solaris Patch Manager Base Version 1.0 constructs a list of patches that is applicable to each system. Based on the results of the PatchPro analysis, it then downloads the patches using a secure `https` protocol (including all dependencies), sorts the patches in installation order, and applies them according to the policy set for that system. Lists are constructed automatically (not manually) based on host information and metadata associated with each of the patches.

While Solaris Patch Manager Base Version 1.0 is a powerful component in the patch management "toolbox," it does have some restrictions. It must be run on the local system for the PatchPro analysis to occur. It must also be able to connect to the Sun infrastructure to obtain the latest Patch Database and detector information. Additionally, using Solaris Patch Manager Base Version 1.0 in an environment with many systems results in inefficient use of bandwidth, downloading the same patch, the Kernel Update patch, for example, multiple times.

This paper further explores the capabilities of the PatchPro Analysis Engine (PatchPro) and describes how to leverage its capabilities to more efficiently allow its use in environments where all hosts do not have direct access to the Sun infrastructure.

The information in this document is based on PatchPro v2.1. Because PatchPro v2.2 should be available by the time this paper is widely distributed, changes from what is described in this document and additional functionality that might be available in PatchPro v2.2 are included in Appendix A.

PatchPro Overview

PatchPro performs a “static” analysis of a system. It bases the analysis on a host system's configuration, hardware platform and configured devices, the operating system version, and installed software packages and patches. PatchPro does not consider active processes, runtime resource utilization, or which applications are used on the system being analyzed. In addition to this static system information, PatchPro uses the Patch Database, downloaded from the SunSolveSM Support Patch Portal (SunSolve Patch Portal) as `patchprodb.zip`, and a set of detectors, downloaded as `pprodetectors.jar`, in its analysis.

PatchPro Database

The `patchprodb.zip` file contains metadata entries for each available patch, including one or more “named characteristics” about a system to which it could be applied. There are currently about 1000 such host characteristics, standardized and registered across Sun, that PatchPro can detect. A named characteristic might look like this:

```
BaseOS.SolarisCore-5.9
Software.VeritasVolumeManager-3.2
DiskArray.SunT300-1.0
```

Detection algorithms in the `pprodetectors.jar` file determine whether the characteristic exists on the analyzed system. These algorithms are Java™ class files that are loaded and run, one at a time, against the host object being analyzed. Each detector looks for its specific characteristic and, if found, adorns the host object with its “tag.” After all of the detectors have been run, the host object contains tags for all the characteristics for which a patch might be available.

PatchPro then generates a patch list for the host by scanning the contents of the downloaded `patchprodb.zip` file and matching the host object tags against the tags associated with each patch. During this phase, patch dependencies are resolved. Each patch listed in the `patchprodb.zip` file names the patches on which it depends. If a candidate patch is found to have a dependency on a patch that has not been added to the patch list, it is added at this time, along with any of its dependencies.

The database entry might also specify conditional expressions that must be evaluated to further determine applicability to that host. An example of such an expression might be as follows:

```
PATCH_REQUIRES='(if isarchitecture sparc; then
    if isosversion 5.8; then
        echo "108987-12 110934-11";
    elif isosversion 5.9; then
        echo "112951-04 113713-02";
    fi
    elif isarchitecture i386; then
        if isosversion 5.8; then
            echo "108988-12 110935-11";
        elif isosversion 5.9; then
            echo "114194-01 114568-02";
        fi;
    fi )'
```

In this case, if a tag is matched, the specific patches to be recommended are dependent on the version of the Solaris Operating System that is installed on the host.

After all patch data has been processed, a list of patches that are applicable to the host is generated and sorted to resolve installation order dependencies. The patch list is then passed to the download and install facilities, depending on the options used in calling the PatchPro process.

PatchPro downloads patches in the patch list to a designated download location as signed jar-format files, authenticating the digital signatures during the download process to ensure that the files come from Sun in the same condition in which they were signed. The jar-format files are not extracted to directory-format patches in the download directory. PatchPro extracts the files during the installation and verifies their digital signatures again as further protection against tampering.

PatchPro Properties

Each of the downloaded jar-format patches specifies a handling property that PatchPro uses to govern its ability to install automatically. PatchPro currently recognizes eight handling properties:

- `<null>` - No special actions are required for pre/post patch installation.
- `singleuser` - Patch must be installed in single-user mode.
- `rebootafter` - Reboot is required to enable delivered functionality.
- `rebootimmediate` - Reboot is required immediately following patch installation to ensure reliable system behavior.
- `reconfigafter` - Reconfiguration reboot (`reboot -r`) is required to enable the functionality delivered by the patch.
- `reconfigimmediate` - Reconfiguration reboot (`reboot -r`) is required immediately following patch installation to ensure reliable system behavior.
- `interactive` - User intervention is required for installation, for example, a README file contains Special Install Instructions.
- `nonstandard` - Patch does not conform to Sun's Patch Automation Conformance Specification, for example, most firmware patches.

If configured to do so by the administrator, PatchPro, executed with the install option will attempt to install all patches that specify no special handling using a call to the `patchadd` utility. The administrator may modify the configuration file to set a policy for PatchPro to automate installation of patches with other handling properties, such as `singleuser` and `rebootafter`. However, PatchPro will never attempt to install a nonstandard patch automatically. PatchPro does not execute a reboot or a reconfiguration reboot as part of the automation. This must be done by the administrator.

Patches that have a nonstandard property and those that the administrator-defined installation policy does not allow to be installed automatically are sequestered. Sequestered patches are placed in a "holding" directory so the user can manually install them later.

The analysis, download, and installation functions described in this document can be done separately or with a single command. However, the host object with its characteristic tags and the resulting patch list are not persistent. This means that regardless of the options PatchPro is called with (analyze, download, install, or update), a new analysis is done, and for each analysis process that is started, both the host object and the resulting patch list are re-created.

Server and Client Configuration

By using some of the PatchPro capabilities and excluding others, and with some additional end-user processes, this tool can be used in a more efficient manner without sacrificing functionality. It should also be possible to incorporate some features that administrators would find useful, even though they are not currently provided by the product.

For instance, by saving copies of the `patchprodb.zip` and `pprodetectors.jar` files for use by PatchPro in analyzing a client, a baseline 30, 60, 90, or any period of time could be established for aged patches. Or by saving a copy of the patch list created during the analysis of each of a number of hosts, it would be possible to aggregate the lists into a single download request, making more efficient use of storage and bandwidth.

Consider a Data Center with multiple hosts, possibly with minimal access to Sun, multiple operating systems, and some degree of automation required, and the following is possible and might be desirable:

- Local repository of patches
- Baseline of patches based on system configuration, operating system, or date
- Security of digitally signed patches
- Automation of patch baseline construction
- Audit of system conformance to patch baseline

Setting Up a Local Repository of Patches

A local repository of patches can be created in several ways, including the use of `wget` (click Automate Downloads at the SunSolve Patch Portal), patches delivered on CD, or using PatchPro functionality. Regardless of the method used to obtain the patches, a set of patches can be used by the patch clients in the Data Center without the following:

- Communicating with an external location
- Duplicating a given patch “n” number of times
- Being restricted to the latest version of the patch

While the local repository does not have to be segregated by operating system, baseline date, and so on, it might ease maintenance to do so.

Patch Depot Server

The creation of Patch Depot server, using PatchPro functionality to populate it, requires that it be able to communicate with Sun to obtain the patches and the associated `patchprodb.zip` and `pprodetectors.jar` files. It also requires that the Solaris Patch Manager Base Version 1.0 client be installed and configured.

PatchPro will be executed from this host with the download option and passed an aggregated list of the patches desired. The patches will be obtained from the SunSolve Patch Portal (with their digital signatures verified for security) and stored in the directory designated by the `patchpro.conf` file in the `/opt/SUNWppro/etc` directory. If the copy of the `patchprodb.zip` and `pprodetectors.jar` files downloaded from Sun and cached in `/opt/SUNWppro/lib/cache` is saved at this time and later copied to the patch clients, this will be the set of patches the clients see as “current,” easily creating a set of patches by date, system configuration, or other criteria desired by the administrator.

Note that a patch downloaded on day X and installed on day Y inherently has two drawbacks. The patch might not still be current and might not contain a fix from a later revision that is important in the environment. Though it does not occur often (less than 1 percent historically), a patch might be withdrawn due to problems discovered after its release. There are no fail-safe methods of mitigating

either issue, but thorough testing in the deployment environment, subscribing to the Patch Club reports at <http://sun.com/newsletters>, and reviewing the Sun Alert Reports at http://sunsolve.sun.com/private-cgi/show.pl?target=sunalert_patches prior to deploying a set of patches will help address these issues.

A Depot populated with PatchPro requires few, if any, specific changes to the Solaris Patch Manager Base Version 1.0 installation. There are, however, a couple of issues the administrator must be aware of.

Any option used when calling PatchPro, including the download option, results in an analysis being done on the current host, in this case, the Patch Depot. As a result, the downloaded patch set will include any patches that are applicable to the host, in addition to those included in the aggregated patch list passed to the download option. While this might not concern those creating a patch set based on date, it can be quite disconcerting when attempting to create a patch set for an E10K and using an Ultra 2 as a patch server, or while trying to create a set of Solaris 9 patches on a patch server with Solaris 7 installed. This issue can be mitigated by using administrator-created scripts that remove the unwanted patches or by moving the desired patches (based on the aggregated patch list) to another, possibly more permanent location.

Read-Only File System

The directory of patches downloaded by PatchPro is intended to be shared as an NFS-mount to the patch clients. PatchPro does not extract the jar-format patches during the download process. For security purposes, the jar-format patches are maintained in the same configuration as they were published by Sun.

If left in this format, the file system would have to be shared read-write for the patch client to install the patches. The side effect of doing this is that once installed, the patches are removed from the directory as part of the cleanup PatchPro does during install. To prevent the removal of patches that are needed by another client, the file system must be shared read-only, with the result that the patches must be extracted to directory-format patches prior to deployment. This can be automated via an administrator-created script, but also review the notes on extracting .jar files at <http://sunsolve.sun.com/private-cgi/retrieve.pl?doc=fsalert/46964>.

Synchronization With SunSolve Patch Portal

The `patchprodb.zip` file is based on the patches available on the SunSolve Patch Portal at the time the file is created. Every effort has been made to ensure that the downloaded files and the patches on the Portal are “in sync.” Due to the time required to populate the various SunSolve servers worldwide, however, a mismatch is possible. To ensure that PatchPro is getting the database, detectors, and patches from the same location at the same time, use the `pprosetup` command to point PatchPro to the following location:

```
# /opt/SUNWppro/bin/pprosetup -P \  
https://patchpro.sun.com/servlet/com.sun.patchpro.server.PatchProServer \  
Servlet
```

Note: This location is expected to become the default for PatchPro.

Setting Up a Patch Client

As with the Depot Server, installing Solaris Patch Manager Base Version 1.0 on a patch client is slightly different from the default setup. The differences include the following:

- Modifying the `patchpro.conf` file to prevent the client from attempting to download a current version of the `patchprodb.zip` and `pprodetectors.jar` files
- Identifying the location of the certificates used in verifying digital signatures
- Setting a different location for the PatchPro server
- If required by the environment, setting an appropriate proxy to be used

As discussed previously, to analyze and create a host-specific patch list, PatchPro must be executed locally on each patch client. This can be accomplished by installing Solaris Patch Manager Base Version 1.0 on each individual client or by installing a single copy to a location that can then be NFS-mounted by the patch client as needed.

PatchPro is not operating system specific, but it does require operating system-specific patches for the Java runtime environment. These patches are included with and installed with the Solaris Patch Manager Base Version 1.0 product. If deploying PatchPro via a shared directory, the Java patches must be installed first on each client. A list of the patches required for each of the supported operating systems appears in Appendix B.

Note: The `patchpro.conf` file requires different edits for the Depot and for the patch clients. Consequently, the same installation of Solaris Patch Manager Base Version 1.0 cannot be used for installing both. Both installations can be on the same system, however. Solaris Patch Manager Base Version 1.0 can be installed using the defaults, which places PatchPro in `/opt`.

By using the `-R alternate_root` option, a `-a admin_file` argument with the setup script, or both, an installation for use by the patch client could exist in `/export/opt`. Subsequently, the `/export/opt/SUNWppro` directory could be mounted to `/opt/SUNWppro` on the patch client so the same commands could be run on both for process consistency.

Editing the `patchpro.conf` File

File Aging

When executed, PatchPro will attempt to download a current copy of the `patchprodb.zip` and `pprodetectors.jar` files based on the aging policy in the patch client `patchpro.conf` file. Using the file aging defaults helps ensure that the files used during analysis and the patches available on the SunSolve Patch Portal are in sync. If the server cannot be reached, PatchPro will use a previously cached copy of the files, if they exist and are “current enough” based on the aging policy in `/opt/SUNWppro/lib/cache`.

To ensure that the client will use the desired copy of these files, the `patchpro.conf` file should be edited to modify the “old” and “dead” age associated with each file. The values used in the following extract from the `patchpro.conf` file are examples, and much longer times may be used if needed (tested using over 1000 days). These edits must be done by hand or by an administrator-created script (not by using the `pprosvc` command).

```
#
patchpro.cache.file.database=lib/cache/patchprodb.zip
patchpro.cache.old.age.days.database=10
patchpro.cache.dead.age.days.database=20
patchpro.cache.file.detectors=lib/cache/pprodetectors.jar
patchpro.cache.old.age.days.detectors=10
patchpro.cache.dead.age.days.detectors=20
#
```

Note: While identifying a different location for the `patchprodb.zip` file has been successful in a lab environment, PatchPro does not accept a change in location for the `pprodectors.jar` file. Do not modify the paths to these files.

Locating Certificates

Part of the setup for the Solaris Patch Manager Base Version 1.0 application is the population of certificates used in authenticating the digital signatures during patch download and installation. If the application is installed on each patch client individually, there is no need to go beyond the instructions in the README file for populating these certificates.

However, if a single installation of the application is to be shared, from the Depot server for instance, the location of the certificates will not be local to the patch client as indicated in the `patchpro.conf` file defaults. The lines in the `patchpro.conf` file need to be modified to point to the location on the server where the certificates are located. Alternatively, authentication during download and installation could be turned off (set to false) on the client since the download will be accomplished in this case by the Depot server.

```
#
patchpro.patch.download.authenticate=true
patchpro.patch.install.authenticate=true
patchpro.security.patch.signing.alias=patchsigning
patchpro.security.kslocation=/usr/j2se/jre/lib/security/cacerts
patchpro.security.crl.source=file:lib/crl.jar
#
```

Avoiding Using Current Files

In the Data Center environment discussed in this document, it might not be desirable (or possible) for the patch clients to connect to the SunSolve Patch Portal because the current `patchprodb.zip` and `pprodectors.jar` files will be downloaded. These files will be downloaded even if the file aging information has been changed as previously noted because the file versions on the SunSolve Patch Portal will be recognized as more current (presumably) than the copy the clients should use.

If no connection can be made to the specified patch server URL, the request will time out. The cached files will be used, assuming they are within the specified aging parameters. To avoid having to manage different configurations and depending on the client's access to the Internet, the default patch server URL should be set to a bogus address (such as shown in this example) because the request will only time out if no connection is made to the server.

```
patchpro.patch.server.url=https://nonexist.patchmanager.sun.com/patchmanager
```

Proxy Configuration

In some environments where proxies are configured on the intranet, PatchPro might not time out when attempting to reach the bogus patch server URL. This condition should be identifiable in a truss by a line like `syn_sent to proxy` and leave PatchPro hanging indefinitely. If needed, change the `patchpro.conf` file to resolve the problem:

```
patchpro.proxyserver.host=nonexist
```

Capturing the Patch List

The created patch list is sent to standard output or, if the analysis is initiated via the scheduling function, mailed to the administrator. This list is in installation order with all dependencies resolved. No file is created on the client. To use the results of the analysis in aggregating patches to the Depot server or to install the patches at a later time, the administrator needs to develop a means to capture, save, and when needed, retrieve the analysis output.

Summary: How to Populate the Local Patch Depot and Create Client-Specific Patch Sets

Populating the local Patch Depot and creating the client-specific patch sets is a multistep process, much of which can be automated via administrator-created scripts. These tasks can also be done manually, for example, when testing the configuration prior to deployment.

1. Execute the PatchPro analysis on the Depot server.

This step must be done even if the resulting patch list will be aggregated for the download step, as it obtains the required `patchprodb.zip` and `pprodetectors.jar` files. The downloaded files are not operating system specific and can be found in `/opt/SUNWppro/lib/cache`.

2. Distribute the downloaded `patchprodb.zip` and `pprodetectors.jar` files to each of the clients being managed, or populate the appropriate shared directory if Solaris Patch Manager Base Version 1.0 is being shared via NFS rather than installed on each client.

3. Analyze each of the clients being managed.

The analysis of the system provides a patch list, including dependencies, in installation order. When run interactively, the file is sent to standard output. When executed via the scheduler, the list is included in an email to the administrator.

4. Maintain a copy of the patch list for each client (to be used during the installation phase), gather all of the client lists and the list created on the Depot, and collate the lists into a single list to be used during download.
5. Use the PatchPro download option with the collated list as an argument to obtain all the patches required by the Depot and client systems.

For security, PatchPro will verify the signatures on each of the downloaded files.

Note: The `patchprodb.zip` and `pprodetectors.jar` files are updated daily, normally around 2:00 a.m. Pacific Time. To ensure that the files and the contents of the patch server at Sun are in sync, complete Steps 1-5 within the same 24-hour period. The remaining steps can be completed at any time the administrator chooses.

6. Extract each of the `patch_id.jar` files into regular directory-format patches to share the directory read-only.

In normal use, PatchPro will extract the patch during installation, but requires that the directory be shared read/write and has the disadvantage in this case of doing a cleanup (removal) of the `.jar` file after the patch has been applied. Since several systems are likely to need the same patch, this is inadvisable. At the administrator's discretion, the digital signatures on the jar-formatted files can be checked again at this time. Documentation describing the process can be found at the SunSolve Patch Portal site.

Caution! A bug in jar archives results in permissions not being preserved during extraction (BugID 478892). Both Patch Manager and PatchPro work around this bug when they are used to extract and install signed patches. However, since implementation of the Depot will require that the patches be extracted manually, type the following to provide the proper permissions on procedure scripts that are included in some patches.

```
# /usr/bin/chmod -R u+x path/patch_id
```

Even though this might result in setting execute permissions on some files that do not need them, such as the README file, it will have no adverse effect on the patches. It does ensure that any other files that must be executed during the installation, or potentially a future backout, have the proper permissions.

Note: A solution is currently under investigation. If a new signing process is successfully implemented, existing patches will be resigned, making this step unnecessary.

7. When ready to install patches on the client systems, NFS-mount the Patch Depot directory of the extracted patches (read-only), and use the `patchadd` utility to apply the patches:

```
# patchadd -M patch_dir patch_list
```

where `patch_dir` is the NFS-mounted directory of patches, and `patch_list` is the client-specific patch list created in Step 3 and saved in Step 4.

Note: Consider modifying the `install_cluster` script (included with the Recommended Cluster) as an installation tool. This script will install the patches using `patchadd -M` as shown above, but it will also create a log file of the installation in the patch client's `/var` directory. This can be useful in determining the results of the individual patch installations and troubleshooting anything that might occur out of the ordinary.

Additional Features

A few customers might have these additional issues that PatchPro can address:

- Excluding patches to avoid impacting locally deployed applications
- Including patches, possibly created locally, that should be deployed concurrently with the Sun patch set

The first case involves breaking down the `patchprodb.zip` file into individual `patchinfo` files, each containing the “named characteristics” discussed earlier. Removing the `patchinfo` file that corresponds to the patch that should not be installed and re-creating the `patchprodb.zip` file ensures that the patch will not be included in the analyzed host's patch list.

The two scripts can be downloaded from <http://patchpro.sun.com/accessories/database/topatchinfos> and <http://patchpro.sun.com/accessories/database/topatchdb> to assist in extracting the individual `patchinfo` files and reconstructing the `patchprodb.zip` file, respectively. However, the decision to use them should not be made lightly. Sun provides no support in either their application or troubleshooting any issues that might arise from their use. It is solely the user's responsibility to ensure that dependencies on any removed patch are properly resolved.

The second case involves doing the following:

1. Identifying a configuration to PatchPro that results in a characteristic tag being added to the host object
2. Creating a `patchinfo` file for the locally developed patch
3. Creating a modified `patchprodb.zip` file as described above

Note: The patch identified in this manner must be installable using the `patchadd` utility.

Since most customers will not need to explore these options, a detailed description of the processes for implementing them is beyond the scope of this document. These options are included to show that PatchPro can be used in many environments, including those with special requirements. If other requirements arise, send a request for more information, including the specifics of what is desired, to patchpro_feedback1@sun.com.

PatchPro v2.2

PatchPro v2.2 should be available by the time this paper is widely distributed. PatchPro v2.2 adds greatly to the PatchPro capabilities discussed in this document.

To get the most out of the release, including the configuration `-c` option, patch 116126-02 must be installed.

`-c` Option

The `-c` option is normally used with the `pprosetup` command to create a file in the `/etc/opt/SUNWppro/etc` directory. Files can also be created manually, in any number, as long as the location is not changed. These files may contain one or more lines of configuration options that “overlay” the `patchpro.conf` file after it has been read, which allows greater flexibility in configuring PatchPro for individual, or groups of, patch clients.

For example, when setting up the Depot server to download patches from the SunSolve Patch Portal and configuring the patch clients to contact a nonexistent server, rather than modifying the `patchpro.conf` files as documented earlier, two files, “depot” and “client” can be created by using the following commands:

```
# /opt/SUNWppro/bin/pprosetup -c depot -P \  
https://patchpro.sun.com/servlet/com.sun.patchpro.server.PatchProServer \  
Servlet  
  
# /opt/SUNWppro/bin/pprosetup -c client -P \  
https://nonexist.patchmanager.sun.com/patchmanager
```

These commands result in two files being created in `/etc/opt/SUNWppro/etc` with the following contents:

```
# cat depot  
  
patchpro.patch.server.url=  
https://patchpro.sun.com/servlet/com.sun.patchpro.server.PatchProServer \  
Servlet  
  
# cat client  
  
patchpro.patch.server.url=  
https://nonexist.patchmanager.sun.com/patchmanager
```

When a `pprosvc` command is subsequently executed with the `-c` option and one of these files as an argument, the `patchpro.conf` file is read, and then the appropriate file, overlaying the default `patchpro.conf` keyword/value pair with the one contained in the file.

Not all keyword/value pairs can be modified using the `pprosetup` command, though virtually all of them can be contained in files used as arguments to `pprosvc -c`. To include these modifications in the file, manual editing is required as shown in the following example.

Patches installed using the `-i` option in PatchPro v2.1, and PatchPro v2.2 by default, are removed from the download directory. PatchPro v2.2 includes an option to retain patches in the `patchpro.conf` file. Manually edit the file as follows:

```
patchpro.retain.patches=true
```

Or, if patch 116126-02 is installed and the `-c` option is to be used, the previous line could be included in a file, for example, the previously described “depot” or “client” files, or a new file named `retain` and called in the following manner:

```
# pprosvc -c retain -i
```

This command results in the `retain` file (in this example) being read, overlaying the default in the `patchpro.conf` file and retaining patches in the download directory after installation. See the `pprosvc(1)` man page for a description of the `-i` option.

The `-c` option eliminates having to perform two installations of the Solaris Patch Manager Base Version 1.0 application since differences in the `patchpro.conf` file are no longer necessary.

Future Patch Manager Releases

Future Patch Manager releases are expected to automate many of the capabilities that are now done manually, including the ability to create and maintain a local Patch Depot.

Also, future releases are expected to have the ability to pass PatchPro a specific realization, or description of a set of patches (for example, the Recommended Cluster) and have that specific set of patches downloaded, installed, or both automatically. Other “sets” of patches are under consideration at this time, but no specifics are available.

Finally, there are plans to add an option to defer patches, installing them automatically, but only at a time specified by the user, for example, when the system can be taken to single-user mode or rebooted.

APPENDIX B

Patch List

	SPARC	i386		
SunOS 5.6	105181-20	105210-27	106041-13	105182-20
	105284-33	105568-17	105211-27	105569-16
	105591-09	105633-38	105200-11	106248-27
	105669-10	106040-13	106844-05	106410-01
	106125-09	106409-01	105491-05	105670-09
	106841-01	106842-09	105285-33	108642-03
	107733-06	108091-03	104678-07	106126-09
SunOS 5.7	107544-03	106541-11	107545-03	106542-11
	106980-10	107153-01	106981-10	107152-01
	107636-03	108376-07	107637-03	109105-01
	109104-01	106950-09	108377-06	106951-07
	107081-20	106327-08	107082-20	106328-08
	106300-09			
SunOS 5.8	108940-07	108941-07		
SunOS 5.9	None	None		