



A Patch Management Strategy for the Solaris™ Operating Environment

Ramesh Radhakrishnan, Sun Professional Services

Sun BluePrints™ OnLine—January 2003



<http://www.sun.com/blueprints>

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-1115-10
Revision 1.0, 1/8/03
Edition: January 2003

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95045 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

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 United States and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Explorer, Solaris Live Upgrade, Sun Cluster, SunSolve Online, SunSolve EarlyNotifier, Solaris Patch Manager, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US 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 use. 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 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95045 Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

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 enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Sun BluePrints, Sun Explorer, Solaris Live Upgrade, Sun Cluster, SunSolve Online, SunSolve EarlyNotifier, Solaris Patch Manager, 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.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

A Patch Management Strategy for the Solaris™ Operating Environment

In today's mission critical information technology (IT) environments, reliability, availability, and serviceability (RAS) are indispensable. Software patches provide the means of performing software maintenance that, when handled properly, contribute to improved RAS, and thus the successful operation of your business. At the same time, managing patches has become complex and time consuming. It's no surprise that many IT professionals seek a comprehensive patch management strategy to reduce the complexity and enhance the overall operation of their IT environment.

This article offers a high-level strategy for managing patches in mission critical, business critical, and business operational, compute environments that are running on the Solaris™ operating environment (Solaris OE). This article divides the patch management process into seven phases, each of which can be tailored to suit your distinct IT environment. This article does not discuss the step-by-step process of installing Solaris OE patches, but instead addresses higher-level concepts that can be used with any patch installation utility.

This article is intended for IT managers, IT architects, lead system administrators, and anyone interested in developing a patch management strategy.

Why Develop a Patch Management Strategy?

Patches provide a means to update software without having to upgrade to a new operating system or application version. Patches are used to repair defects and to add or change software features. As with most software environments, patches are part of routine administration for the Solaris OE. Managers of mission critical environments and non-mission critical environments struggle to find the best patch management strategies that deliver the best possible software support while minimizing system downtime.

Change management and other IT service management processes have a standard. This standard is defined in the IT Information Library (ITIL). ITIL describes the goal of change management as a method “to provide procedures to facilitate any change into the IT infrastructure with minimal risk and maximum efficiency”. The process must provide a proper balance between the need for change and the impact of change. Where applicable, this article identifies places where the patch management strategy fits into the ITIL standard.

Your patch management strategy should be considered part of change management. Your patch management strategy is certainly a special case of change management because of the complexity involved. For example, a patch might be required to upgrade a system to take advantage of a new feature. The situation is further complicated by the unknown—applying this patch might introduce new problems. Avoiding the patch, in this example, would prevent the successful upgrade. This complex situation requires careful considerations that are best rolled into your standard change management processes.

Developing a sound patch management strategy is critical to successfully manage IT environments for the following reasons:

- You keep mission critical and business critical systems up-to-date with required patches for fixes to known problems, possibly preventing problems before they negatively affect your compute environment.
- In environments where there are several systems with varying patch needs, based on the type of applications running on them, managing patches and keeping the systems at appropriate patch versions can be very complex. By developing a strategy for managing this software maintenance, you'll simplify the process and improve the results.
- The environments required for successful patch management, namely development, test, integration test, and preproduction might already exist in your IT environment. The same environments can be used to manage patches effectively by rolling them out in a phased manner through these environments.

For those data centers in which these environments do not exist, this patch management strategy is an excuse to create these environments to roll out not just patches, but also application software upgrades and many other types of changes.

- Categorizing applications into mission critical, business critical and business operational environments is a good starting point towards developing detailed Service Level Requirements for these applications and providing them with appropriate infrastructure, architecture, people and processes to manage them successfully.

Phases of the Patch Management Process

This article separates a comprehensive patch management strategy into seven phases as shown in FIGURE 1 and discussed below.

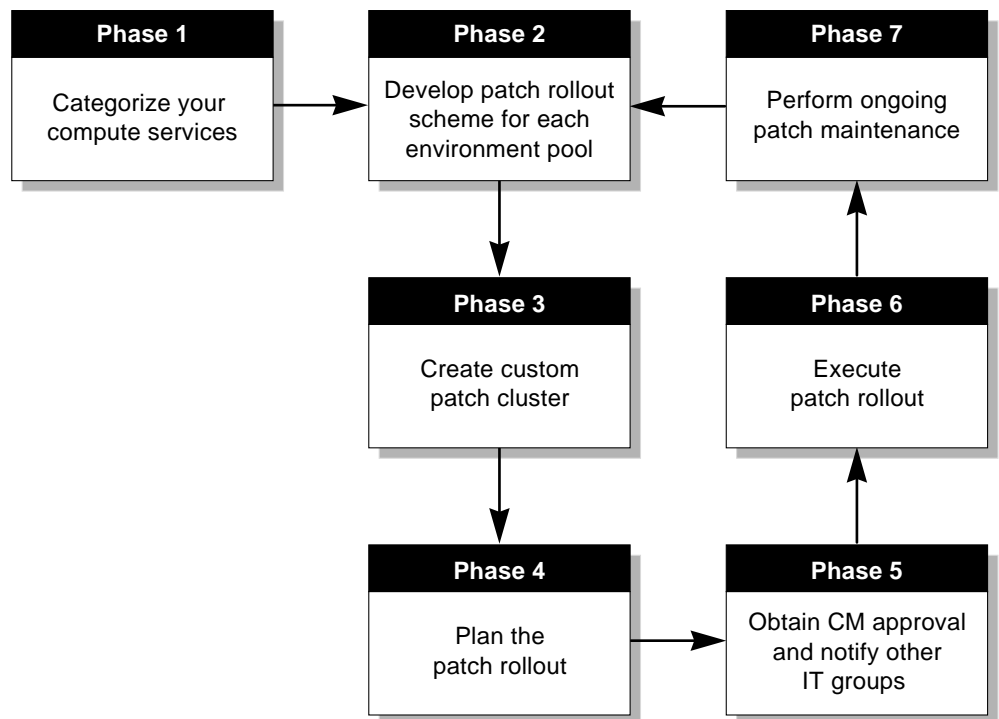


FIGURE 1 Fig 1. High-Level Patch Management Process

Phase 1—Identify and categorize the different types of compute services in your compute environment into three distinct categories; mission critical, business critical, and business operational.

Phase 2—Develop a rollout scheme for each of the three environment categories (mission critical, business critical, and business operational).

Phase 3—Create the patch clusters for the each type of environment based on the rollout method selected.

Phase 4—Plan the patch rollout so that the proper records are maintained, and identify the most appropriate contingency plan (patch back-out plan).

Phase 5—Obtain change management approval and notify other departments.

Phase 6—Execute the patch cluster rollout.

Phase 7—Perform ongoing patch monitoring and maintenance.

Refer to the patch management flow chart at the end of this article for an illustration of the kinds of decisions you make as you move through these phases.

Phase 1: Categorizing Your Compute Services

Your compute environment is probably comprised of different types of compute services. This includes a set of systems supplying mission critical services, business critical services and business operational services. You might have all or a subset of these environments. It is helpful to think in terms of the service-level requirements for reliability, availability, serviceability, scalability, security, performance, and so on. Prioritize your compute services based on functional business requirements.

The first step in the patch management process is to identify how critical the service is. The selection of patches, the rollout strategy and execution, all vary based on the prioritization of the compute environment services. For example, in a mission critical environment, system security is a high priority because a break-in can result in a compromised system that can cripple a business, either through reduced service or worse, intentional data theft or corruption. In this example, security is a critical factor and security patches need to be added to the list of patches in the patch cluster (new security patches come out every month as new ways of intrusions by hackers are discovered).

The following three broad compute environment categories are used throughout this article:

- Mission critical environment—an environment in which even one hour of downtime will have a significant impact on the business service, and availability is required at almost any price.
- Business critical environment—an environment in which business services require continuous availability, but breaks in service for short periods of time are not catastrophic.
- Business operational environment—an environment in which breaks in services are not catastrophic.

Mission Critical Environment

A mission critical environment typically has the following sub environments (FIGURE 2):

- Development environment for developing applications that will eventually be moved into production.
- Test environment for unit test of applications.
- Integration environment in which several application units are integrated and tested together. The integration environment is also often used for endurance testing, failure testing, and load testing of applications after applying Solaris patches.
- Preproduction environment for final testing. The preproduction environment often closely resembles the production environment and is used for final testing in an environment that has most of the variables found in the actual production environment. In some cases, the integration environment is also used as the preproduction environment.

After moving through all these environments, the application (whole application or additional features or bug fixes) is promoted to the production environment.

Among the sub environments, only the production environment is actually mission critical. This can vary depending on whether each one of these sub environments are physically separated. In many data centers, the sub environments are located physically on the same system, but have a logical separation through logical domains or other virtual machine techniques.

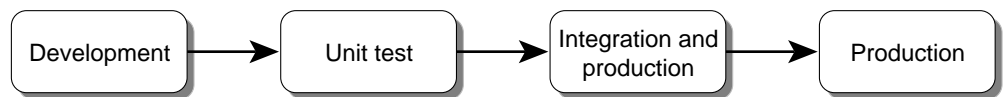


FIGURE 2 Sub Environments in a Mission Critical Environment

Business Critical Environment

A business critical environment might not be as strictly controlled as a mission critical environment and might only have one test environment for all types of testing. FIGURE 3 shows a typical business critical environment.

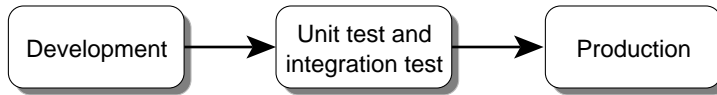


FIGURE 3 Sub Environments in a Business Critical Environment

Business Operational Environment

A business operational environment might have a loosely controlled production environment. Testing is typically done in the development environment and applications are promoted directly to the production environment. FIGURE 4 shows a typical business operational environment.

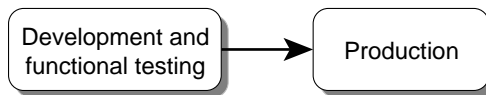


FIGURE 4 Sub Environments in a Business Operational Environment

Categorize Your Compute Services Into Environment Pools

For each service or application, determine the environment type and categorize each into various pools as shown in FIGURE 5.

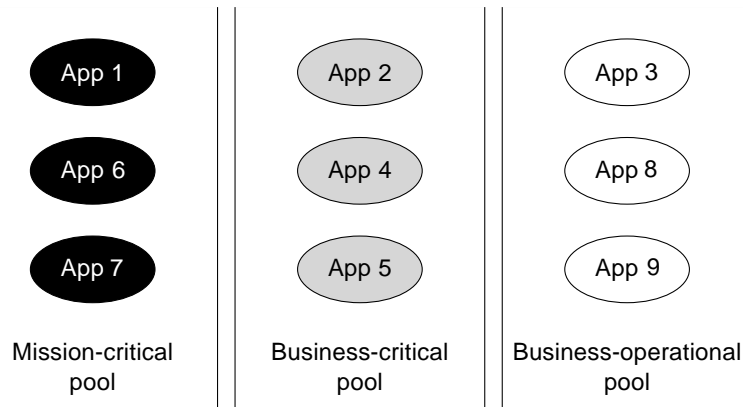


FIGURE 5 Environment Pools

For each application or service in each environment pool, all systems associated with the application (including the application servers, web servers, database servers, middleware servers, and so on), are placed in the same environment pool, and all of them have the same criticality level. For example, a fictitious company called Fortune 526, categorizes three of their environments into three pools as follows:

Mission Critical	Business Critical	Business Operational
<p>Fortune 526 identified their e-commerce services as the mission critical component of their compute services because this environment is used to sell the company's products and services online. If this environment is not available even for a few minutes, the impact is very high, including loss of revenue and customer satisfaction.</p> <p>All systems that support the following e-commerce services are placed in the mission critical pool:</p> <ul style="list-style-type: none"> • Web servers • Firewalls • Application servers • Database servers associated with their e-commerce application 	<p>The SAP environment is used by Fortune 526 for internal employee payroll. If this environment is not available for an hour or less, payroll processing will be delayed, but the company will not experience great revenue loss.</p> <p>All systems that support the SAP applications are placed in the business critical pool:</p> <ul style="list-style-type: none"> • SAP application servers • SAP database servers 	<p>Fortune 526's customer resource management (CRM) environment is used for sales forecasts and reporting. This environment is required to accurately predict revenue, but if it is down for a small period of time, the company will not lose revenue.</p> <p>All systems that support the CRM applications are placed in the business operational pool:</p> <ul style="list-style-type: none"> • CRM application server • CRM distribution server • CRM middleware servers that are used to integrate with a legacy system

Phase 2: Developing a Patch Rollout Scheme

In any IT environment, changes are initiated from many areas. A pending patch rollout is one of those areas where a change is initiated. In the standard change management process, your first task is to filter and prioritize the incoming requests based on potential importance and complexity. If a change is deemed urgent, the urgent change process is initiated that facilitates a quick solution, but also increases the risk of failure. The standard change process handles any other change request.

Phase 2 outlines a patch rollout scheme for each type of environment pool, taking into account the need for quick solutions (emergency rollouts) to those environments that can follow the standard change process (rapid and regularly scheduled rollouts).

Schemes for Mission Critical Environments

In a mission critical environment, the patch rollout scheme can be classified into the following three categories:

- Regularly scheduled rollout scheme
- Rapid rollout scheme
- Emergency rollout scheme

In the case of patch management, you should integrate the regularly scheduled rollout scheme into your standard change process. The rapid rollout scheme and emergency rollout schemes for patches should be special cases of the urgent change process within the context of your change management process.

Regularly Scheduled Rollout Scheme

Use the regularly scheduled rollout scheme (FIGURE 6) as the normal mode of operation for applying patches in a mission critical environment. The following steps define this scheme:

1. The first step is to analyze the available patches at the SunSolve Onlinesm program¹ and create a patch cluster customized to your mission critical environment. Perform this analysis on a monthly basis to ensure that your mission critical services are as up-to-date as possible.
2. In the first month, apply the patch cluster to a development environment. This provides the opportunity to test the patch cluster on systems in which developers are developing code that will eventually be promoted to the production environment. If problems are encountered, software developers can correct the problems or you can coordinate with Sun to resolve any problems².

1. SunSolve Online program is Sun's online support portal, accessible at <http://sunsolve.sun.com>.

3. In the second month, roll the patch cluster into the test (integration test) environment. This provides for testing of the patches with software that will soon be moving into the production environment.
4. In the third month, roll the patch cluster into the preproduction environment. Use this environment for endurance, stress, and regression testing, inducing heavy loads on the systems that are equivalent to the mix of loads that will be generated in the actual production environment. In addition, conduct failure testing to determine if high availability features (such as redundant networking) are working properly.
5. At end of the third month, roll the patch cluster into the production environment.

Rapid Rollout Scheme

Use the rapid rollout scheme (FIGURE 6) when a lead system administrator has identified a known problem that requires a few patches to be added fairly quickly. A good example is a situation in which a system administrator has been notified by a security administrator about a CERT advisory that describes a recent attack into a Solaris system. The system administrator uses the patch analysis tools available from the SunSolve Online program to determine that a patch has been released that fixes this security hole. This process can be used also to add a feature to a system that is critical to the business service and that cannot wait for more than a month.

In this process, the identified patches are first rolled into a test environment, tested for 15 days or so, and then rolled into the preproduction environment and stress and regression tested for 15 more days, and finally rolled into the production environment. This scheme moves the patches into the environment sooner than the regular rollout scheme, but still gives them a total of one month for endurance testing the patches before moving the patches into production.

Note – For some e-business applications, fixing a security hole might warrant using the emergency rollout scheme discussed in the next section.

Emergency Rollout Scheme

Use the emergency rollout scheme (FIGURE 6) in highly critical situations that require application of one or two patches immediately. Typically, this is a situation in which a mission critical system has a serious problem, and after analysis, it is found that one or two patches exist that would fix the problem. Care should be taken to apply only those patches that are directly relevant to fixing the current problem. This opportunity should not be used to apply other patches, otherwise you introduce

2. Sun thoroughly tests released patches, but mission critical environments should further test all patches to make sure they function properly with all the mission critical applications.

more unknown entities into the environment, further complicating a challenging situation, and it won't be clear which patch actually resolved the problem. Consistently following the regularly scheduled rollout scheme ensures that emergency patching situations occur rarely.

In this process, the senior system administrator or architect determines, with the help of Sun, what patches are required to immediately rectify the problem. Sometimes, this can be a temporary patch, also called a *T-patch* or a patch found in the Sun Alert list¹. Once the patches are determined, they are carefully applied to the preproduction environment, tested in that environment at least for a few hours, then an urgent maintenance window is scheduled, and the patch is applied to the production environment. Care should be taken to create a well-tested back-out scheme before applying these patches to the production environment.

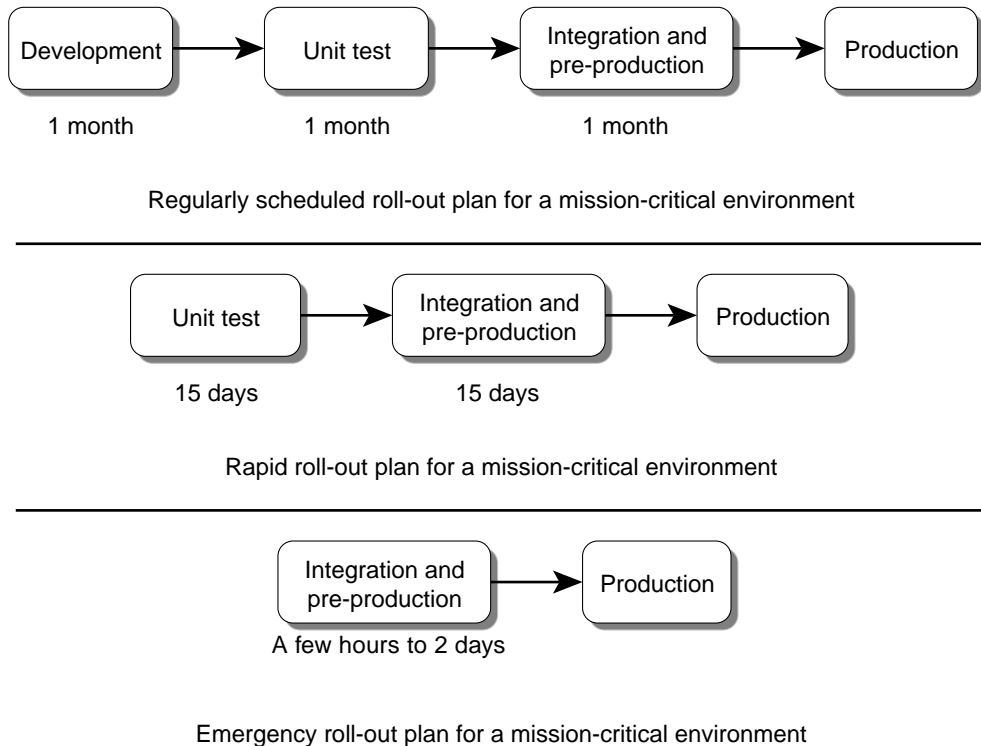
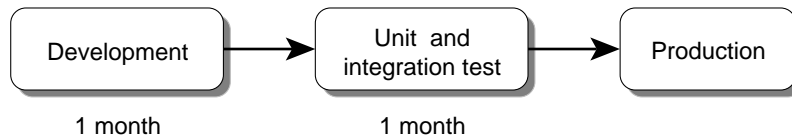


FIGURE 6 Rollout Schemes for a Mission Critical Environment

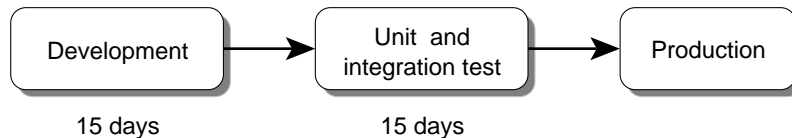
1. The Sun Alert list is available from the Patch Portal at <http://sunsolve.sun.com>.

Schemes for Business Critical Environments

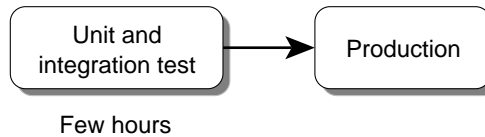
The patch rollout schemes (FIGURE 7) for the business critical environment is similar to the schemes used in mission critical environments. Use the three rollout schemes; regularly scheduled, rapid, and emergency rollout schemes. But you might choose to skip testing in two environments, and instead opt to test only once in the integration environment as part of the regularly scheduled rollout process. In addition, for the rapid rollouts, you can opt to cut down the testing time to 15 days per environment. For the emergency rollout scheme, testing for just a few hours in the integration environment before rolling into production might suffice.



Regularly scheduled roll-out plan for a business-critical environment



Rapid roll-out plan for a business-critical environment



Emergency roll-out plan for a business-critical environment

FIGURE 7 Rollout Schemes for Business Critical Environments

Schemes for Business Operational Environments

A business operational environment may not require specific rollout schemes for regularly scheduled, rapid, and emergency rollouts. If no problems were found after applying the patch cluster to the development or test environment, the patch cluster can be rolled into production. Some companies have a higher availability for business operational environments and might opt to treat their business operational environments like business critical environments and follow the strategies outlined in the previous business critical section.

Plan to test the patch cluster by installing it in a development environment (test for a few hours), and apply the patch cluster to your production environment.

Phase 3: Creating a Custom Patch Cluster

Phase 3 is complex, but becomes simpler with practice. Effective use of the tools that are available in Sun's Patch Portal from the SunSolve Online program is essential.

The following table summarizes the tools that you can use to create a patch cluster customized for your target environments. More information about each tool is available from the Patch Portal at <http://sunsolve.sun.com>.

TABLE 1 Patch Management Tools

Patch Management Tools	Description
PatchCheck ¹	Performs a basic analysis of a system. Generates an HTML report of all installed patches and compares it with a cross-reference file that has a list of all current patches, including recommended and security patches. From the resulting data, you can create a patch suite that can be downloaded.
Patch Finder	Search for a particular patch from the SunSolve patch database. Use the Patch Finder to display a specific patch description.
PatchPro	Analyzes and generates a custom patch list for your system.
Signed patches	Verifies that a patch is from Sun. Using signed patches is a secure method of applying patches because they include a digital signature that can be verified.
Solaris™ Patch Manager Tool	Provides configuration-specific patch analysis, automated patch downloads, patch dependency resolution, and patch installation. The Solaris Patch Manager Tool requires Perl version 5.0 at a minimum.

TABLE 1 Patch Management Tools (*Continued*)

Patch Management Tools	Description
Sun alert patches	Temporary patches for recently discovered problems. These patches are part of the SunSolve EarlyNotifier sm Service.
Sun TM Explorer software	Collects system data. Run the Sun Explorer software tool on your system to create an output file that is a snapshot of your system configuration. Send the Sun Explorer software output to Sun and request a list of missing patches.
Sun Root CA certificate	SUNWcert package used to verify the digital signatures in Signed Patches.

1. PatchCheck is being superseded by Patch Manager, but Patch Manager can only be used for the Solaris OE, Sun Network Storage, SunTM Cluster systems, and Sun EnterpriseTM Server products.

Based on service level requirements, some mission critical environments require a manual selection of patches to create a custom patch cluster. The more critical your environment, the more selective you should be in building your patch cluster. Some situations require application of one patch at a time, and avoiding application of a whole patch cluster in one single run. This article assumes that your senior system administrator and the senior system architect have the expertise to use the patch management tools to come up with a customized patch cluster for your target production environment. Some things to consider are as follows:

- Consider the Solaris Jumbo Kernel Patch

This patch is one of the most critical patches that you should consider as a patch candidate. Often times the version of this patch is reported in the output of the `uname -a` command. Review the `README` for this patch to see if this patch has important fixes and updates for the platform that you are using. For example, if most of the updates in the latest jumbo patch have fixes for the Sun FireTM 15K system, and your environment does not have any Sun Fire 15K systems, then you might not want to apply this patch. However, you might consider this patch anyway to satisfy support issues with Sun (for example, SunTM Service might require the installation of the latest Solaris Jumbo Kernel patch in order to support your Sun systems).

- Investigate Sun Alert Patches

Review the `README` files for the latest Sunsm Alert patches to see if any of the problems or fixes are applicable to the platforms in your environment. If so, dig a little deeper and call Sun support to find out more about these patches. Do a thorough risk analysis with Sun, and with your own IT management, and then select the patches for candidates in creating your patch cluster.

- Check for Recommended and Security Patches

Review the `README` files for each one of these patches and see which ones are applicable for your environment. For example, if there is a fix for a Fibre Channel Arbitrated Loop (FC-AL) card, and you don't use Fibre Channel in your environment, this patch may not be required for your system. When it comes to security patches, it is a best practice to apply all Recommended and Security patches, and then be selective about all other patches.
- Verify Patches With Your Vendors

While selecting patches, ensure that applying a patch or a cluster of patches will not break an application that works well with the current version of Solaris OE. The patch selection process should include contacting the applications and RDBMS vendors to make sure that applications work well after applying these patches to Solaris OE. Testing, as discussed in Phase 2, will also surface compatibility issues.
- Analyze Patch Dependencies

Analyzing the patches for dependencies is also an important factor in patch selection. The Patch Manager tool listed in TABLE 1 helps with dependency analysis. It is also important to check for nested dependencies.

Phase 4: Planning the Patch Rollout

Proper patch management strategy includes a plan that tracks all patches, and provides a means to back out of any patch. This part of the strategy must take place before the patch rollout.

Patch Tracking—Building a Patch Store

It is advantageous to create a server (with sufficient bandwidth and processor power to deploy patches through the network) that can serve as a patch repository, or *patch store*, and can be used for the following:

- Providing an archive of all patch clusters that were ever rolled out.
- Functioning as a patch server for patch clusters that are planned for rollout.

- Aiding with version control for all patches to keep a history of patches applied to various systems and environments. The following table sample can be used as a template for a simple version control and record keeping:

Patch Cluster Name and Number	Directory	Installation Date/Time	Environment Names	System Names
SUNWyyy/yy-24	/home/patches	09/30/02 18:00 PST	Retail Web Env/ Integration Test Subenv.	Mercury, Venus, Mars

- Tracking troublesome patches. First remove the problem patch from the patches available in the patch store, and create the following table that can be used as a template for a simple way of tracking problem patches:

Patch Cluster Name and Number	Directory	Problem Details	Type and Name of System Where Problem Occurred	Name and Type of Environment
SUNWxxx	Security fix	Broke rsh command	Sun Fire 4800 (Mercury)	Integration test sub- env under retail web env/mission critical

- Tracking change management. A table can be stored in the patch store with the following details:

Patch Cluster Name and Number	Business Reason for Patch	Patch System Administrator	Lead System Administrator and Architect Approver	Management Approver
SUNWxxx	Improves users' response time.	John K. Administrator	Bill D. Lead	Tom S. Manager

Once the patch store is created and the proper records are kept, you have the following patch information available:

- A patch archive with tested patches ready for installation in your production environment.
- A record of all systems and all patches that have been applied in your environment.
- A history of patches that caused problems and required the patch to be backed out. The patch store can be used to generate a list of all systems and environments in which a particular patch has been applied.

Contingency Planning—Devise a Patch Back-out Method

There is always a chance that the installation of a patch cluster will cause something to break or degrade, even after testing in preproduction environments. No matter which type of environment you are planning to patch, you should make appropriate contingency plans by defining a method to back out of an installed patch cluster. The following methods offer some ideas.

Method A

This method is the simplest way to back out of a patch and can be used in a business operational environment. Every Sun Patch comes with a `README` file with instructions on how to install the patch and how to back out of a patch. By default, installing a patch cluster saves an original version of the objects being patched in the `/var/sadm/patch` directory. The back-out script uses the saved objects to restore the software to pre-patched state. In some cases, even though you've installed an entire patch cluster, patches can be backed out only one patch at a time using the `patchrm` command. To take extra measures to make sure that your back-out plan works, apply the patch cluster to a test system, and follow the back-out instructions in the `README` to make sure you are able to successfully back out of the patch cluster.

Note – The patch installation command offers a `-nosave` option which prevents the saving of the existing objects. It is highly recommended not to use this option, just in case the need arises to back out of a patch cluster.

Method B

This method can be used in a business critical environment. This method involves using method A, but requires that you back up your Solaris OE to a tape before applying the patch cluster. Use the `ufsdump` or `vxdump` commands to back up to tape depending on what type of file system (`ufs` or `vxfst`) your Solaris OE is using. The system must be in a quiesced state when a backup is taken. In this case, in addition to the back-out script, you have a backup to restore your Solaris OE to the a pre-patched state. Alternatively, since tape restores are very slow, business critical environments that cannot wait for a slow recovery from tape, can maintain an additional disk and use the UNIX[®] `dd` command to make a copy of the boot disk.

Method C

This method can be used in a nonclustered mission critical environment where downtime to the production system can be minimized by using a tool called Solaris™ Live Upgrade¹ software during the patch rollout process. Solaris Live Upgrade allows you to make a replica of the boot image of the production system while it is up and running and in a low-usage period. Solaris Live Upgrade software can first be used to make a copy of the boot environment, the patch cluster can be applied to the copy, and the copy can be taken through a rigorous testing process. The copy can be moved to a test system for thorough testing. If everything goes well with the tests, Solaris Live Upgrade software can be used to switch the copy to become the active production boot environment. The only downtime will be the time taken to do the Solaris Live Upgrade software switch, which is equivalent to the time taken to reboot a system. Following the procedure outlined in Method B is a prerequisite. In addition to that, consider making two tape copies of your Solaris OE in case one of the tapes becomes unusable.

Method D

This method can be used in a clustered mission critical environment. Sun Cluster software is usually put in place for high availability in mission critical environments. A clustered environment gives the opportunity to minimize downtime of the actual production system during patch rollouts. The minimizing process requires breaking the cluster, keeping the primary node running in a production mode, applying the patch cluster to the standby node, and conducting stress and regression tests using the standby node. After successful testing, the standby node is made the active production system, the patches are applied to the primary production node, stress tested and regression tested, and finally both nodes are moved into the cluster. Downtime is minimized to the time it takes to remove the nodes from the cluster and the time to put the cluster back together. This is typically equivalent to the time it takes to reboot a production system. Following the procedures outlined in Method B is a prerequisite.

1. Refer to <http://www.sun.com/software/solaris/liveupgrade> for Solaris Live Upgrade details.

Phase 5: Obtaining Change Management Approval and Notifying Other Groups

Once the required patch cluster has been created, the next step is to get approval from the appropriate change management authorities. In some companies, there is a change advisory board (CAB), or sometimes referred to as a change control board (CCB). The CAB exists to approve changes and to assist in the assessment and prioritization of changes.

The kinds of steps that relate to patch management change control are as follows:

- Communicate the nature of the change in the established format for your organization. This might entail completing a change management form that describes the planned changes, the business justifications, list of departments and systems affected, dates, expected outcome, contingency plans (patch back-out plan), required resources, and so forth.
- Obtain formal change management approvals. This might require paper or electronic signatures as defined in your change management processes. For example, all CAB members, Business Unit Manager for the system in which the change is being made, IT director, and so on might have authority. Some companies' processes require some end-user approvals also. Approvals must be obtained before moving to the next step. Approved forms must be filed carefully for future reference and future internal and external audits.
- Update the change control information in the patch store as explained in Phase 4.
- Employ the use of *implementation management*¹ for patches that constitute major changes. This is a function of the ITIL standard change management process that facilitates the build-out and preparation necessary for successful deployment of significant changes. If the patch rollout results in minor changes the implementation management portion can be skipped.

Make sure that you notify all the appropriate departments. Some departments to consider are as follows:

- Help desk—Tell them about the impending change, with date and time of introduction of change in production. This will give help desk an opportunity to be prepared for additional calls during that period by arranging for additional personnel or by asking the existing personnel to be on high alert.
- System management and monitoring center—They should know about the change so that they can anticipate any alerts and prepare for problem diagnosis as needed.
- Call Center or Escalation center—If a problem occurs after the change is made, and if the problem was caused by the new change, the escalation managers will at least know where to start looking to conduct root cause analysis.

1. Implementation management manages the process of specifying, designing, construction, testing and deploying business information and communication services and IT organization services. Implementation management process also prepares the introduction of new projects and large impact changes into the production environment.

- Vendors related to the system—This includes application vendors, database vendors, hardware vendors and other critical infrastructure vendors.
- Users—Provide contact information so they can communicate any issues as they arise.

These are some typical considerations for mission critical environments. Your data centers no doubt have a set of policies and procedures for communication and approvals that might not be outlined in this article. Your polices obviously must be taken into consideration.

Phase 6: Executing the Patch Rollout

The standard change management process requires that this phase be included in the *execution management*¹ process for promotion to the production environment.

In Phase 6, it's finally time to install the patch clusters. Depending on the environment (mission critical, business critical, or business operational) and the rollout scheme (regularly scheduled, rapid, or emergency), perform the initial rollout into the appropriate test environment. The testing phase should include three levels of testing:

1. Solaris OE level testing

These tests involve creating and running a generic system script that performs commands such as `ping`, `netstat`, process comparisons, NFS test, `iostat`, and so on. The script helps to validate the basic operation of the operating environment after the patch cluster is installed.

2. Application level testing

These tests check various application functionality and ensure that everything still works. For example, in an environment with web servers, application servers, middleware servers, and database servers in various *tiers*, application of a patch cluster to any one tier might affect other tiers in terms of functionality. Testing end users' functionality helps ensure that all tiers are in working order, and that all tiers work well together after applying the patch cluster.

3. Preproduction level testing

Testing in a preproduction environment ensures that application functionality continues to work in an environment that closely resembles the production environment with similar storage area networks (SANs), network infrastructure, security infrastructure, and so on. For example, the combination of applying a patch to a *security hardened* production environment might show problems that are not seen in the development or test environments.

When the patch cluster testing is done, perform the patch rollout in your production environment.

1. Execution management is the management of scheduling, executing, optimizing, and planning business compute processing, and staff work associated with the deployment and administration of changes to the operation.

Phase 7: Performing Ongoing Patch Maintenance

Even though following the previous phases keeps the patches up to date on all the systems, there can be occasional emergency rollouts. This can occur over a period of time and result in systems that are out of sync with each other. Over a period of time, patches might get out-of-date relative to the latest patches from the SunSolve Online program. The process used by system managers might fall through the cracks if there is no on-going audit of this patch identification process. A simple way to automate this maintenance process is described below.

Sun Professional Services supplies a tool called `i-status` as part of the Application Readiness Service (ARS) offering. This tool is a configuration monitoring tool. It is also called a *dynamic run book* because any changes made to a system configuration are automatically updated within `i-status`. The `i-status` tool has an easy-to-use GUI that shows the latest configuration of the system. Without `i-status`, a system administrator has to manually update a static run book. In addition to maintaining up-to-date configuration data of one or more systems, `i-status` can be used to monitor and compare patch levels of various systems.

FIGURE 8 shows an `i-status` screen that has a list of hosts. The % current column shows how current the patches are on each system.

Updated December 04, 2002

Sun Proprietary/Confidential



Select Single Host for Patch Diagnostics (patchdiag)

Date of Last XREF File: Sep/16/02

	Hostname	% Current	System Type	OS	Data as of
☺	A1D1	78%	Ultra-Enterprise-10000	SunOS 5.7	Mon Oct 15 20:45:50 2001
☺	A1D2	79%	Ultra-Enterprise-10000	SunOS 5.7	Mon Oct 15 20:45:50 2001
☺	A1D3	78%	Ultra-Enterprise-10000	SunOS 5.7	Mon Oct 15 20:45:51 2001
☺	A3D1	78%	Ultra-Enterprise-10000	SunOS 5.7	Mon Oct 15 20:45:52 2001
☺	A3D2	78%	Ultra-Enterprise-10000	SunOS 5.7	Mon Oct 15 20:45:53 2001
☺	MACPROD	80%	Sun Enterprise E4500	SunOS 5.8	Mon Oct 15 20:46:05 2001
☺	acsint600-sc0	89%	Ultra CP 1500	SunOS 5.8	Tue Apr 9 13:42:18 2002
☺	acsint600-sc1	89%	Ultra CP 1500	SunOS 5.8	Tue Apr 9 15:29:36 2002
☺	acsint600c	90%	Sun Fire 15000	SunOS 5.8	Tue Apr 9 12:34:59 2002
☺	anadas10	85%	Sun Fire 6800	SunOS 5.8	Fri Nov 9 15:36:28 2001
☺	anads20	85%	Sun Fire 6800	SunOS 5.8	Fri Nov 9 15:36:26 2001

FIGURE 8 Current Patch Report in `i-status` Tool

FIGURE 9 shows another *i-status* screen. There are five columns showing five different systems. The rows that have underlined patches show patches that are out of revision level. The first row shows that a patch is missing on some systems.

Note – Some systems intentionally do not have the latest patches installed. Use the *i-status* tool as a way of gathering information, not as a definitive guide for building your next patch cluster.

The *i-status* tool requires the latest patch cross reference file from the SunSolve Online program. This file must always be up-to-date. To ensure you have the latest file, you can use a scheduler like the UNIX `crontab` utility to periodically download the patch cross reference file. The scheduler can also be configured to remind the you to check the *i-status* console periodically.

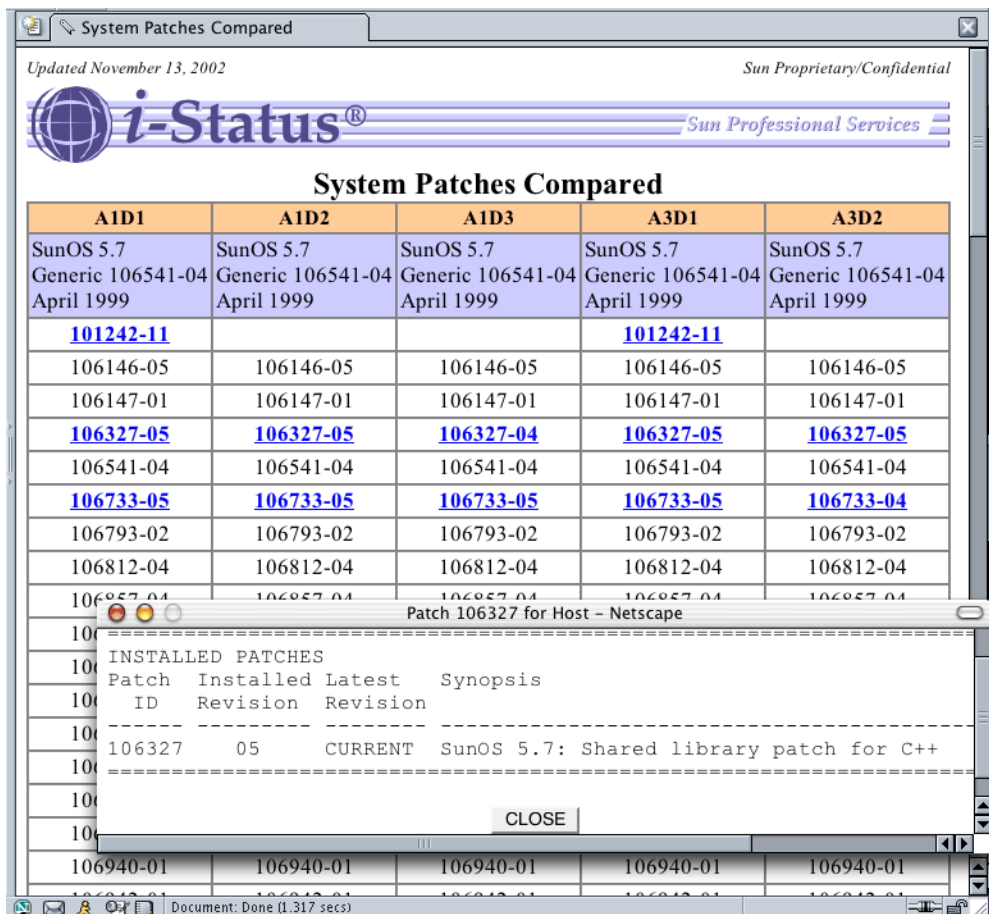


FIGURE 9 Patches Compared in *i-status* Tool

Patch Management Flow Chart

The following flow chart illustrates the kinds of decisions you make as you develop and execute the patch management strategy discussed in this article.

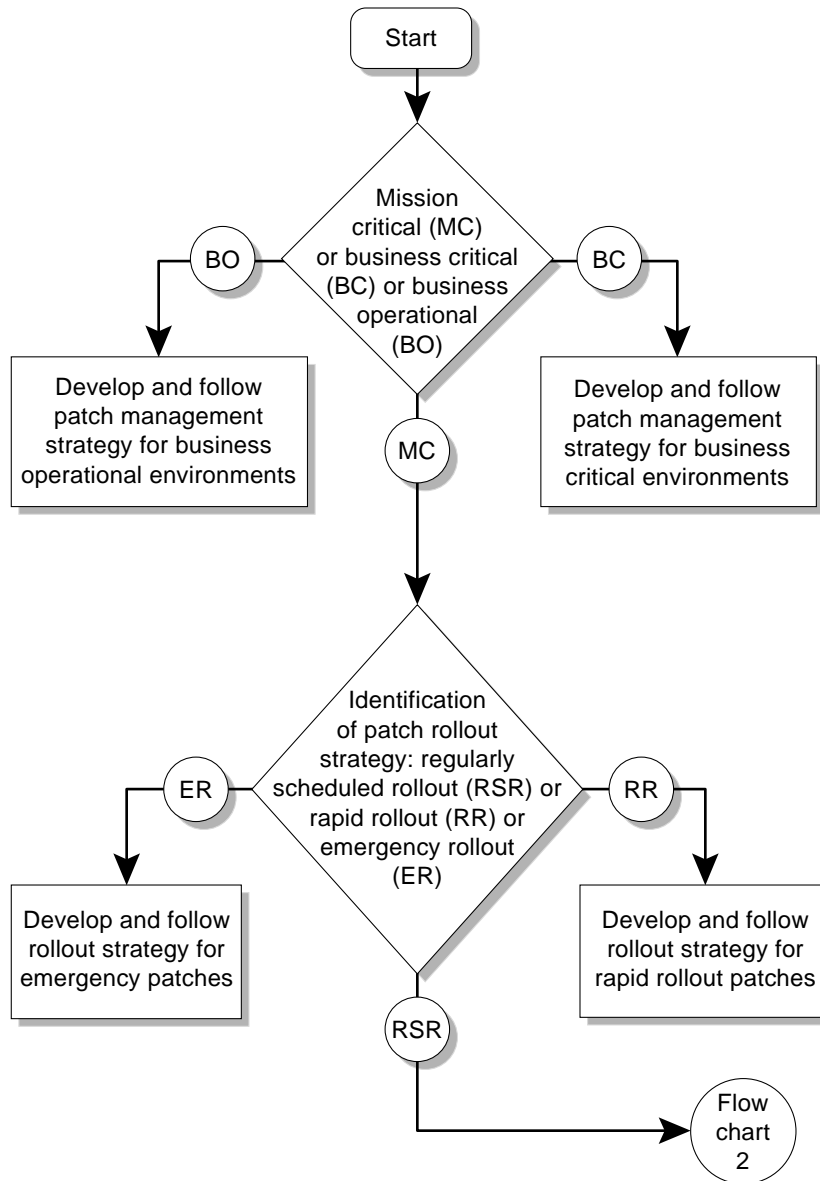


FIGURE 10 Patch Management Flow Chart (1 of 5)

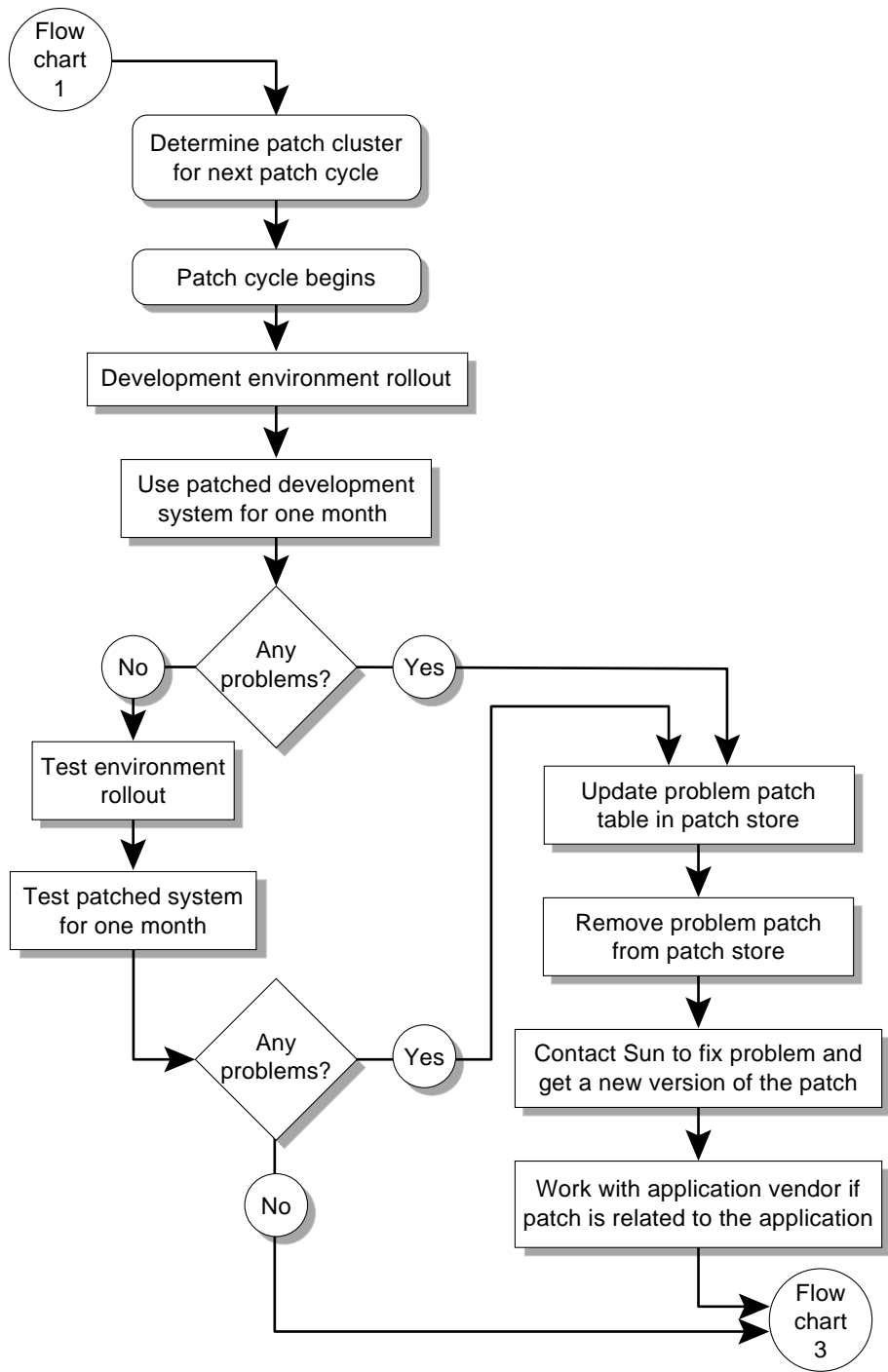


FIGURE 11 Patch Management Flow Chart (2 of 5)

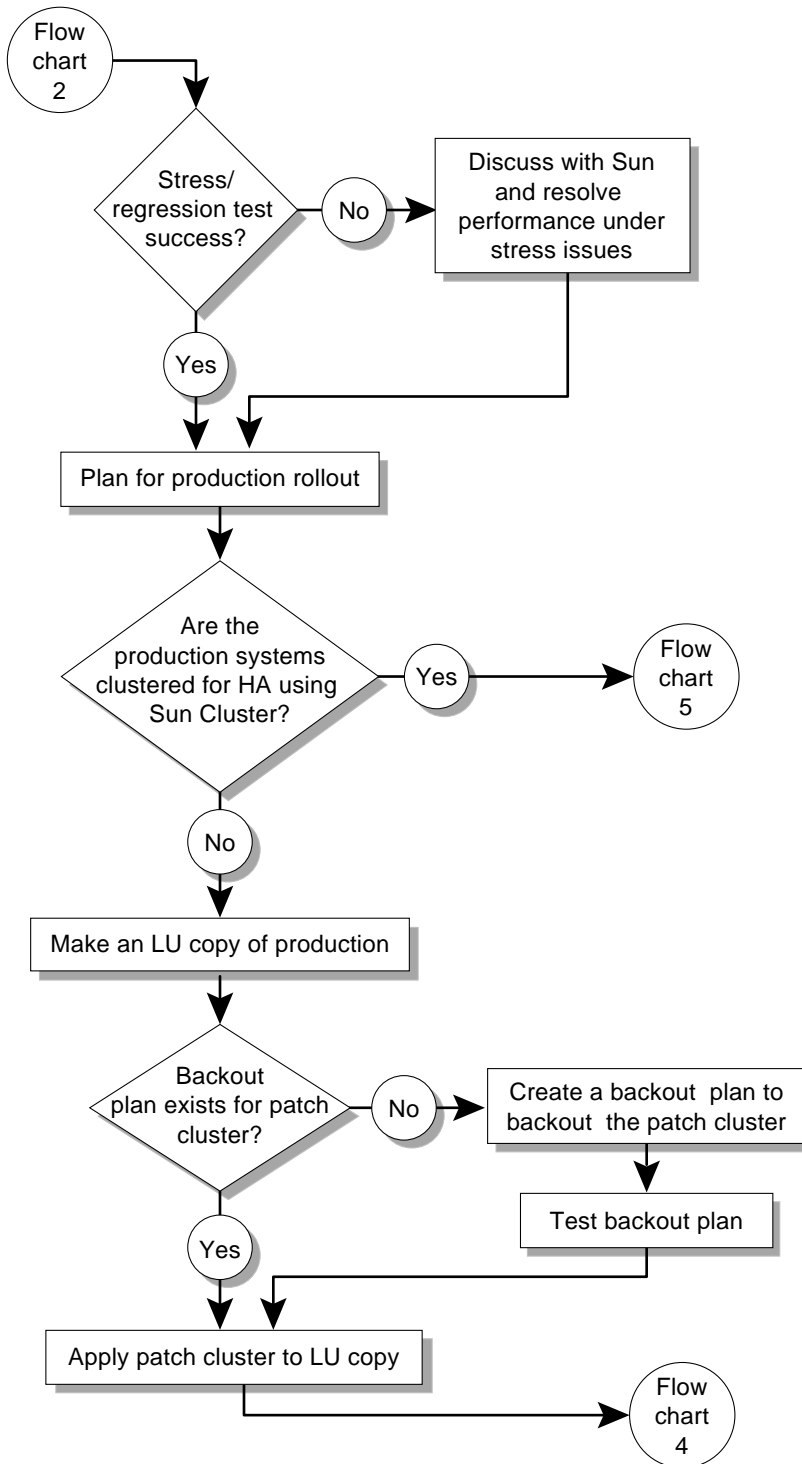


FIGURE 12 Patch Management Flow Chart (3 of 5)

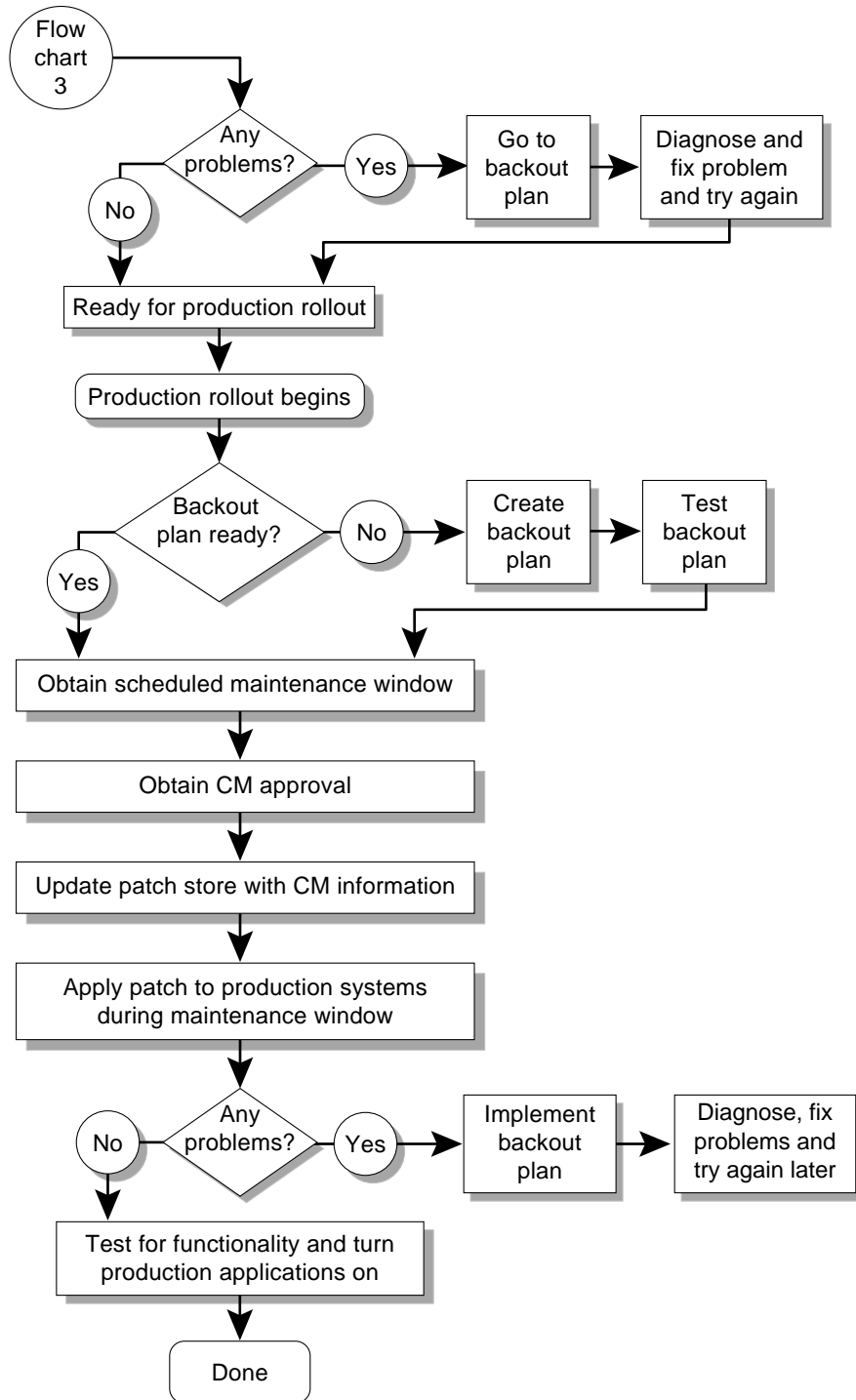


FIGURE 13 Patch Management Flow Chart (4 of 5)

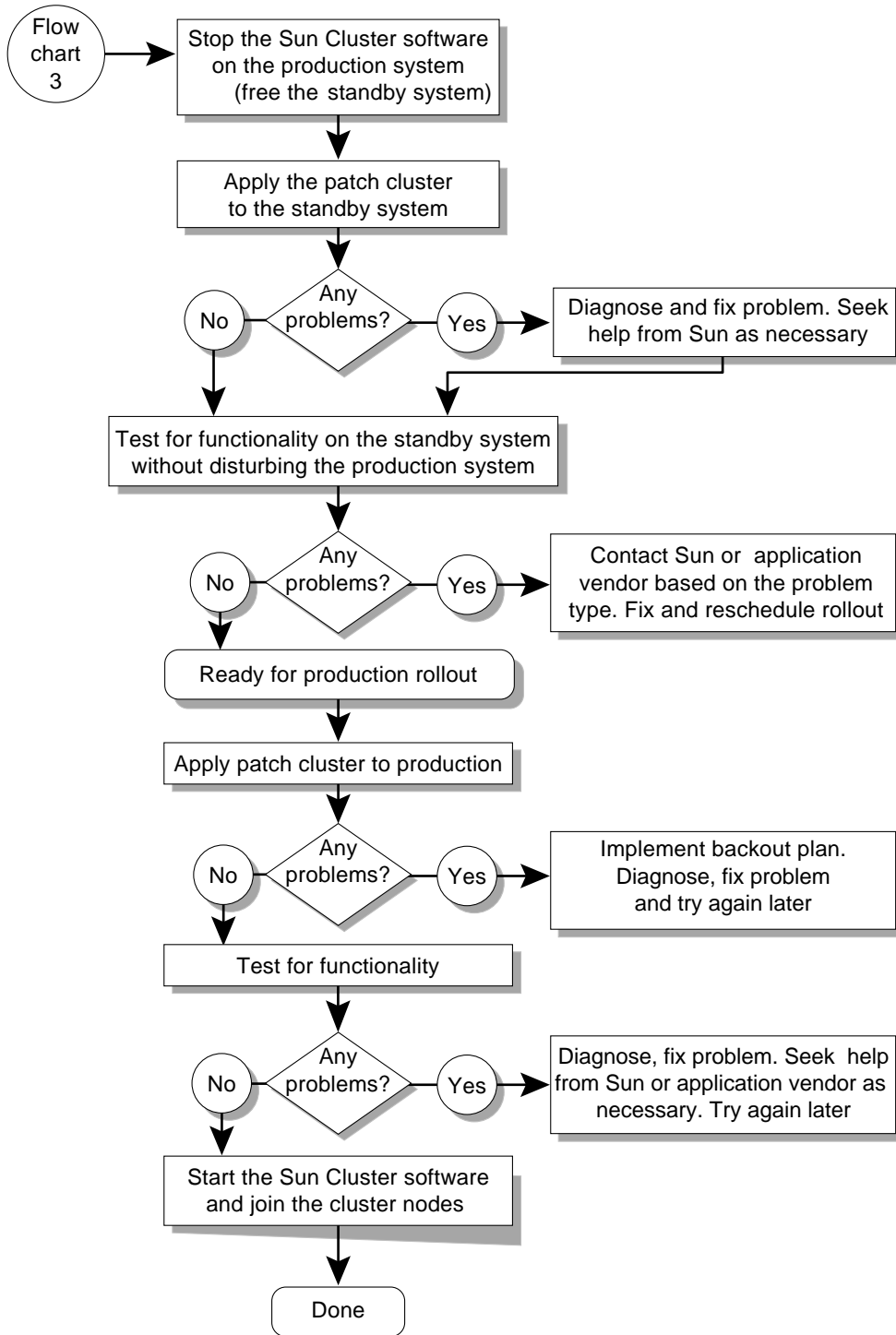


FIGURE 14 Patch Management Flow Chart (5 of 5)

About the Author

Ramesh Radhakrishnan has been an IT Architect and Consultant at Sun Microsystems for the past three years. He conducts availability assessments, architecture assessments, and designs IT environments for several of Sun's mission critical customers. Before joining Sun, Ramesh worked as a system administrator, IT consultant, and ClearCase Consultant. He has a Master's degree in Computer Science from Old Dominion University. Over the years, he has gained experience in the areas of backup and recovery architecture, disaster recovery architecture, and IT processes, along with many other IT infrastructure management areas. Ramesh is currently part of a team developing an architecture basics course for Sun engineers.

Acknowledgements

The author would like to recognize the following individuals for their contributions to this article:

- Michael Barto
- Brad Blumenthal
- Ron Diersen
- Kevin Magnes
- Rajesh Radhakrishnan
- Edward Wustenhoff

References

The *Solaris Patch System Testing and Performance Regression Testing Overview* is an article available at <http://sunsolve.sun.com>. Click on the Patch Portal link.

Sun BluePrints articles on Solaris Live Upgrade software are available at <http://www.sun.com/blueprints/online>

Ordering Sun Documents

The SunDocsSM program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`