Scheduled Maintenance Windows (What, Why and How)

Christine Hogan and Tom Limoncelli

This paper is based on Chapter 12 of *The Practice of System and Network Administration* by Thomas A. Limoncelli and Christine Hogan [LH01].

1 Introduction

Computers and networks need maintenance. They need hardware and software upgrades, patches, and periodic re-architecting. This paper looks at a structured way to perform those tasks. All system administrators (SAs) need to do maintenance, and need to minimize the impact the maintenance has on their customers.

These days people rely heavily on the computer and network infrastructure to perform their daily work. They are not tolerant of ad-hoc approaches to maintenance. Bringing a server down in the middle of the day in order to replace a broken tape drive, or to add more disk space, is no longer acceptable. SAs are expected to perform these tasks outside of normal working hours. What we recommend is a structured approach to maintenance. Schedule periodic maintenance windows, delay intrusive tasks until the next maintenance window, and plan upgrades in advance of the need for them.

Scheduled maintenance windows will look different for different companies. Small companies typically have less equipment and therefore only need short maintenance windows. Fast-growing companies need frequent maintenance windows. High-availability companies need maintenance windows that use redundant systems to maintain service availability during the window. Mid-sized companies need infrequent, but long maintenance windows. Large companies need a combination of location-specific maintenance windows, and "high-availability" maintenance windows for shared infrastructure.

This paper will concentrate on the long, infrequent maintenance windows, and will include some specifics for high-availability sites. Other sites use a subset of these techniques. We call the technique described here the "flight director" technique, named after the role of the flight director in NASA space launches.[1]

The flight director technique guides the activities before the window, during execution, and after. It is summarized in Table 1.

Preparation	Schedule the window
	Pick a flight director
	Prepare change proposals
	Build a master plan
Execution	Disable access
	Shutdown sequence
	Execute plan
	Perform testing
Resolution	Announce completion
	Enable access
	Have a visible presence
	Be prepared for problems

Fig. 1: The Three Stages of a Maintenance Window

2 Motivation

Some companies are willing to schedule regular maintenance windows for major systems and networking work in return for better availability during normal operations. Depending on the size of the site, this could be one evening and night per month, or perhaps an entire weekend, from Friday evening to Monday morning, once a quarter.

System administrators often like to have a maintenance window during which they can take down any and all systems, and stop all services, because it reduces complexity and makes testing easier. For example, in cutting email services over to a new system, you need to transfer existing mailboxes as well as switching the incoming mail feed to the new system. Trying to transfer the existing mailboxes without cutting the feed and the read access for those mailboxes, and yet ensure consistency is a very tricky problem. But, if you can bring email services down while you do the transfer, it becomes a lot easier. In addition, it is a lot easier to check that the system is working correctly before you turn the mail feed and the read access back on again than it is to deal with having dropped or bounced mail if something didn't work quite right with the live cutover.

However, you will have to sell the concept to the company at large in terms of a benefit to them, not in terms of it making the SA's life easier. That means that

[1] The origin of this terminology, and a number of the approaches used, was with Paul Evans. Paul is an avid observer of the space program.

you need to be able to promise better service availability the rest of the time. In other words, you need to plan in advance: if you have one maintenance window per quarter, you need to make sure that the work you do this quarter will hold you through the end of the next quarter, so that you won't need to bring the system down again. You should also be prepared to provide metrics to back up your claims of higher availability[2] from before and after you have succeeded in getting scheduled maintenance windows. A single large outage also can be much less annoying to users than many little outages [LRNL97].

3 Scheduling

A maintenance window is by definition a short period of time in which a lot of systems work needs to be performed. It is disruptive to the rest of the company, and so the scheduling must be done in cooperation with the customers. You need to find out what dates do and don't work for them. In particular, you will almost certainly need to avoid the end-of-month, end-of-quarter and end-of-fiscal-year dates so that the sales team can get all of the rushed orders entered, and the accounting group can produce the financials for that period. You will also need to avoid product release dates, if that is relevant to your business. There may be major trade shows or other demos that must be avoided. Universities have different constraints around the academic year. Some businesses, such as toy and greeting card manufacturers, may have seasonal constraints.

4 Planning

As with all work on important systems, the tasks need to be planned by the individuals performing them so that no original thought, or problem solving, should be involved in performing the task on the day. There should be no unforeseen events, only planned contingencies.

Planning for a maintenance window also has another dimension, however. Since they occur only occasionally, the system administrators need to plan far enough in advance so that they can get quotes, submit purchase orders, get them approved, and have any new equipment arrive a week or so before the maintenance window. Since the lead-time on some equipment can be 6 weeks or more, this means starting to plan for the next maintenance window almost immediately after the preceding one has just ended.

[2] The most effective way of doing this is to use an existing monitoring system that can produce historical graphs. The most popular ones at the time of writing are MRTG [Oet98] and Cricket [All99].

5 Flight director

The flight director is the one who is responsible for crafting the announcement notices and making sure that they go out on time. She is also responsible for scheduling the submitted work proposals based on the interactions between them and the staff required, deciding which ones (if any) don't make the cut for that maintenance window, monitoring the progress of the tasks during the maintenance window, ensuring that the testing occurs correctly, and communicating status to the rest of the company at the end of the maintenance window.

The person who fills the role of flight director must be a senior system administrator who is capable of assessing work proposals from other members of the SA team, and spotting dependencies and effects that may have been overlooked. The flight director must also be capable of making judgment calls on the level of risk versus need for some of the more critical tasks that impact the infrastructure. She must have a good overview of the site and the implications of all the work.

In addition, the flight director cannot perform any technical work of her own during that maintenance window. This means that typically the flight director is a member of a multi-person team, and the other members of the team take on the work that would normally have been the responsibility of that individual. The flight director is not normally a manager, unless the manager was recently promoted from a senior SA position, because of the skill requirements described above.

6 Change proposals

One week prior to the maintenance window, all change proposals should have been submitted. A good way of managing the change proposal process is to have all the change proposals online in a revision-controlled area. Each SA edits documents in a directory with his name on it. The documents supply all the required information. One week before the change, this revision-controlled area is frozen and all subsequent requests to make changes to the documents have to be made through the flight director.

A change proposal form should answer at least these questions:

- What changes are going to be made?
- What machines will you be working on?
- What are the pre-maintenance window dependencies, and due dates?
- What needs to be up for the change to happen?
- What will be affected by the change?
- Who is performing the work?

- How long will the change take in active time and elapsed time, including testing, and how many additional helpers will be needed?
- What are the test procedures? What equipment do they require?
- What is the back-out procedure and how long will it take?

7 The master plan

One week before the maintenance window, the flight director freezes the change proposals, and starts working on a master plan. The master plan takes into account all the dependencies, elapsed and active times for the change proposals. The end result is a series of tables, one for each person, showing them what task they will perform during what time interval and who the coordinator for that task is. There is also a master chart that shows all the tasks that are being performed over the entire time, who is performing them, the team lead, and what the dependencies are. The master plan must also take into account complete system-wide testing after all work has been completed.

If there are too many change proposals, the flight director will find that scheduling all of them produces too many conflicts, either in terms of machine availability, or in terms of the people required. There needs to be slack in the schedule to allow for things to go wrong. The difficult decisions about which projects should go ahead and which ones have to wait should be made beforehand, rather than in the heat of the moment when something is taking too long and blowing the schedule, and everyone is tired and stressed. The flight director makes the call on when some change proposals need to be cut out, and gets all the parties involved in the scheduling conflict to discuss it and pick the best course for the company.

7.1 Building a Master Plan Template

Once we had run a few maintenance windows, we discovered that there was a formula that worked well for us. The systems on which most people were dependent for their work were operated on during the Friday evening. The very first thing to be upgraded or changed was the network. Next on the list was console service, then the authentication servers. While these were in progress, all the other SAs helped out with hardware tasks like memory, disk or CPU upgrades, replacing broken hardware, or moving equipment within, or between, data centers. Last thing on Friday night, large data moves were started so that they could run overnight.

The remaining tasks were then scheduled into Saturday, with some people being scheduled to help others in between their own tasks. Sunday was reserved for comprehensive system-wide testing and debugging because of the high importance placed on testing.

8 Disabling access

The very first task in the maintenance window is to disable or discourage system access, and provide reminders that it is a maintenance window. Depending on what the site looks like, and what facilities are available, this process may involve:

- Placing notices on all doors into the campus buildings with the maintenance window times clearly visible.
- Disabling all remote access to the site, whether by dial-in, dedicated lines, wireless or over a public network.
- Making an announcement over the public address system in the campus buildings to remind everyone that systems are about to go down.
- Changing the helpdesk voicemail message to announce that this is a maintenance window, and stating when normal service should be restored.

These steps reduce the chance that someone will try to use the systems during the maintenance window, which could cause inconsistencies, or accidental loss or damage of their work. It also reduces the chance that the person carrying the on-call pager will have to respond to urgent helpdesk voicemails saying that the network is down.

9 Mechanics and coordination

There are some key pieces of technology that enable the maintenance window process described here to proceed smoothly. These are not solely useful for maintenance windows, but they are critical to their success.

9.1 Shutdown/boot sequence

In most sites there are some systems, or sets of systems, that are required to be available in order for other systems to shutdown or boot cleanly. If a machines tries to boot when machines and services that it relies on are not available, it will fail to boot properly. Typically, the machine will boot, but it will fail to run some of the programs that it usually runs on start-up. These programs might be services that others rely on, or just programs that run locally on someone's desktop. In either case, the machine will not work properly, and it may not be apparent why. When shutting down a machine, it may need to contact file servers, license servers or database servers that are in use in order to properly terminate the link. If it cannot contact those servers it may hang for a long time, or indefinitely, trying to contact those servers before completing the shutdown process. It is important to understand and track the dependencies that machines have on each other during boot up and shutdown. You do not want to have to figure it out for the first time when a machine room unexpectedly loses power.

The most critical systems, such as console servers, authentication servers, name-service machines, license servers, application servers and data servers, typically need to be booted before other machines such as compute servers and desktops. There will also be dependencies between the critical servers. It is vital to maintain a boot sequence list for all data center machines, with one or more machines at each stage, as appropriate. Typically the first couple of stages will have few machines, maybe only one machine in them, but later stages will have many machines. All data center machines should be booted before any non-data center machines, since no machine in a data center should rely on any machine outside a data center.

The shutdown sequence is typically very close to, if not exactly the same as, the reverse of the boot sequence. There may be one or two minor differences.

The shutdown sequence is a vital component to starting work at the beginning of the maintenance window. The machines that are operated on at the start of the maintenance window are typically those with the most dependencies on them, so any machine that needs to be shut down for hardware maintenance/upgrades, or moving, has to be shut down before the work on the critical machines starts, and it is important to get the machines shut down in the right order, so as to avoid wasting time bringing machines back up so that other machines can be shut down cleanly.

The boot sequence is also critical to the comprehensive system testing that is performed at the end of the maintenance window, as we shall see later.

9.2 Console service

All of the equipment in the computer room that is capable of supporting a serial console should have its serial console connected to some kind of console concentrator such as a networked terminal server. The console servers should be running console server software[3] which securely allows authenticated access to machine consoles from authorized hosts across the network. This enables people to work from their own desks, rather than having to try to coordinate access for many people to the very limited number of monitors in the computer room, or having to waste computer room space, power and cooling with more monitors. If possible, the console software should log all output to disk for future reference.

[3] http://www.conserver.com

It is also more convenient for the individual SAs to work in their own workspace with their preparatory notes and reference materials around them.

9.3 Radios

Since the maintenance window is tightly scheduled, there are many dependencies, and system administration work can be unpredictable at times, everyone has to check with the flight director to let him or her know when they are finished with a task, and before they start a new task, to make sure that the prerequisite tasks have all been completed.

We recommend using hand-held radios to communicate within the group. Rather than seeking out the flight director, an SA can just call her over the radio to check in. Likewise, the flight director can contact the SAs to find out status, and team members and team leaders can find each other, and coordinate, over the radio. If someone needs extra help, they can also ask for it over the radio. There are multiple radio channels, and long conversations can move to another channel to keep the primary one free.

The radios are also essential for system-wide testing at the end of the maintenance window, which will be described later, in Section 11.

If radios won't work, or work badly, in your data center due to RF shielding put an internal phone extension with a long cord at the end of every row. That way SAs in the data center can still communicate with other SAs while working in the data center. At worst, they can go outside the data center, raise someone on the radio and arrange to talk to her on a specific telephone inside the data center.

10 Deadlines for change completion

A critical role of the flight director is tracking how the various tasks are progressing, and deciding when a particular change should be aborted and the back-out plan for that change executed. For a general task that had no other dependencies, and where those involved had no other remaining tasks, that time would be 11pm on Saturday evening, minus the amount of time that it would take to implement the back-out plan, in the case of a weekend maintenance window. Though sometimes the sanity and effectiveness of the people working on the task needs to be taken into account. If they are exhausted and frustrated, the flight director may decide to tell them to take a break or to start the back-out process early if they won't be able to implement it as efficiently as they would when they were fresh.

If there are other tasks that depend on that system or service being operational, it is particularly critical to pre-define a cut-off point for task completion. For example, if a console server upgrade was going badly, there might not be time to start the large data moves on Friday night, which would prevent them from happening at all, which could cause a cascade of catastrophic effects for people within the company until the next downtime.

11 Comprehensive system testing

A maintenance window is like taking a very complicated piece of machinery completely to pieces, and then putting it back together again, under a time constraint. Sometimes there are bits left over when you think you should be finished.

The final stage of a maintenance window must involve comprehensive system testing. Depending on the length of the window, you may only be able to test the few components that you worked on. However, if you have a weekend maintenance window, and have a lot of people performing work on may different components, you should plan on spending all day Sunday doing system testing.

Sunday system testing starts with a complete shutdown of all machines in the data center, and then stepping through the boot sequence in order. There is an individual assigned to each machine on the reboot list. The flight director calls out the stages during the shutdown sequence over the radio, and each individual calls in each machine that they are responsible for when it has completely shut down. When all the machines at the current stage have shut down, the flight director calls the next stage. When everything is down, the flight director calls out the stages on the boot list, and tracks the progress. If there are problems with a machine at any stage, those must be debugged and fixed before moving on to the next stage. The person assigned to each machine is responsible for ensuring that all services started correctly before they call it in as being up.

Finally, when all the machines in the data center have been successfully booted in the correct order, the SA team is split into groups. Each group has a team leader, and is assigned an area in one of the campus buildings. The teams are given instructions on what to do and tests to perform. The instructions always include rebooting every machine to make sure it comes up cleanly. They could also include logging in, checking for a particular service, or trying to run a particular application, for example. Each person in the group has a stack of colored sticky tabs (e.g. green) used for marking offices and cubicles that have been completed, and verified working. They also have a stack of sticky tabs of a different color (e.g. red) that they use to mark cubicles that have a problem. When SAs run across a problem, they spend a short amount of time trying to fix it, before calling it in to the central core of people who are assigned to stay in the main building to help debug problems. As a team finishes its area, it is assigned to a new area, or to help another team to complete an area, until the whole campus has been covered.

The central command center, with the flight director and the senior SA trouble-shooters, keeps track of problems on a white-board. The SAs and flight director decide who should tackle each problem, based on the likely cause and who is available. By the end of testing, all offices and cubicles should have tags, preferably all indicating success. If any offices or cubicles still have tags indicating a problem, there should be a note left for that customer explaining the problem, and someone assigned to meet with that person to try to resolve it first thing in the morning.

This method has caught many potentially serious problems before people came into work the following day. For example, disconnected, or incorrectly connected network segments after wiring closet clean up, a failed push to a software depot server, inoperable NIS slaves, AppleTalk problems, and license server problems.

Be warned, however, that there may be machines that weren't working in the first place. The reboot teams should always make sure to note when a machine did not look operational before they rebooted it. They can still take time to try to fix it, but it is lower on the priority list, and does not have to happen before the end of the maintenance window.

Ideally, the system testing, and site-wide rebooting should be completed sometime on Sunday afternoon, giving the SA team time to get some rest after a stressful weekend, before coming into work the next day.

12 Post-maintenance communication

Once the maintenance work and system testing is completed, the flight director sends out a message to the company informing everyone that service should now be fully restored. The message briefly outlines the main successes of the maintenance window. It also briefly lists any services that are known not to be functioning, and when they will be fixed. Hopefully there won't be any nonfunctional services, of course.

This message should be in a fixed format, and largely written in advance, since the flight director will be too tired to be very coherent or upbeat if she writes the message at the end of a long weekend. There is also little chance that anyone who proofreads the message at that point is going to be able to help, either.

13 Re-enable remote access

The final act before leaving the building should be to re-enable remote access, and restore the voicemail on the helpdesk phone to normal. Make sure that this appears on the master plan, and the individual plans of those responsible. It can be very easily forgotten after an exhausting weekend, but it is a very visible, inconvenient and embarrassing thing to forget. Especially since it can't be fixed remotely if all remote access was turned off successfully.

14 Visible presence the morning after

It is very important for the entire SA group to be in early, and to be visible to the company the morning after a maintenance window, no matter how hard they have worked during the outage. Have the people who look after particular departments roam the corridors of the departments they look after, keeping an eye and an ear open for problems.

Have the flight director and some of the senior SAs (from the central core-services group, if there is one) sit in the helpdesk area to listen to what calls are coming, and listen for things that may be related to the maintenance window. They should be able to detect and fix them sooner than the regular helpdesk staff, who won't have such a good overview of what has happened.

15 High availability sites

By the very nature of their business, high availability sites cannot afford to have large planned outages.[4] This also means that they cannot afford to *not* make the large investment necessary to provide high availability. Sites that have high availability requirements need to have lots of hot redundant systems that continue providing service when any one component fails. The higher the availability requirement, the more redundant systems are required to achieve it.[5]

These sites still need to perform maintenance on the systems in service. While the availability guarantees that these sites make to their customers typically exclude maintenance windows, they will lose customers if they have large planned outages.

15.1 The similarities

Most of the principles described above for maintenance windows at a corporate site apply at high availability sites.

- They need to schedule the maintenance window so that it has the least impact on their customers. For example, Internet service providers often choose 2am (local time) mid-week; e-commerce sites need to choose a time when they do the least business. These windows will typically be quite frequent, such as once a week, and shorter, perhaps 4-6 hours in duration.

[4] High availability is anything above 99.9% availability. Typically sites will be aiming for three nines (99.9%) (9 hours down-time per year), four nines (99.99%) (1 hour per year) or five nines (99.999%) (5 minutes per year). Six nines (99.9999%) (less than 1 minute a year) is more expensive than most sites can afford.

[5] The term $n+1$ redundancy is used for services where any one component can fail without bringing the service down, $n+2$ means any two components can fail, and so on.

- They need to let their customers know when maintenance windows are scheduled. For Internet service providers, this means sending an email to the customer contacts. For an e-commerce site, this means having a banner on the site. In both cases it should only be sent to those customers who may be affected and should contain a warning that small outages or degraded service may occur during the maintenance window, and give the times of that window. There should only be a single message about the window.

- Planning and doing as much as possible beforehand is critical because the maintenance windows need to be as short as possible.

- There still needs to be a flight director who coordinates the scheduling, and tracks the progress of the tasks. If the windows are weekly, this may be a quarter-time or half-time job.

- Each item needs to have change proposal, as described in Section 6. The change proposal needs to list the redundant systems and include a test to verify that the redundant systems have kicked in and service is still available.

- They need to tightly plan the maintenance window. Maintenance windows are typically smaller in scope and shorter in time. Items scheduled by different people for a given window should not have any dependencies on each other. There still needs to be a small master plan that shows who has what tasks and their completion times.

- The flight director must be very strict about the deadlines for change completion.

- Everything must be fully tested before it is declared complete.

- Console servers benefit all sites.

- The SAs need to have a strong presence when the site approaches and enters its busy time. They need to be prepared to quickly deal with any problems that may arise due to the maintenance.

15.2 The differences

There are also a number of differences in maintenance windows for high availability sites.

- There is a prerequisite that there is redundancy at the site if it is to have high availability.

- It is not necessary to disable access. Services should remain available.

- It is not necessary to have a full shutdown/boot list, since a full shutdown and reboot does not happen. However, there should be a dependency list if there are any dependencies between machines.[6]

- Since the customers of Internet service providers and e-commerce sites are not on-site, being physically visible the morning after is irrelevant. However being available and responsive is still important. Find ways to increase your visibility and assure excellent responsiveness. Advertise what the change was, how to report problems, and so on.

- A post-maintenance communication is usually not required, unless there are remaining problems that the customers need to know about. Customers don't want to be bombarded with email from their service providers.

- The most important difference is that the redundant architecture of the site needs to be taken into account during the maintenance window planning. The flight director needs to make sure that none of the scheduled work can take the service down. The SAs need to make sure that they know how long fail-over takes to happen.[7] If redundancy is implemented within a single machine, the SA needs to know how to work on one part of the machine while keeping the system operating normally.

- Availability of the service as a whole needs to be closely monitored during the maintenance window. There needs to be a plan for how to deal with any failure that causes an outage due to temporary lack of redundancy.

[6] Usually high availability sites avoid dependencies between machines as much as possible.

[7] For instance, how long does the routing system take to reach convergence when one of the routers goes down or comes back up?

References

[All99] Jeff R. Allen. Driving by the rear-view mirror: Managing a network with cricket. In *First Conference on Network Administration (NETA '99)*, pages 1–10, Santa Clara, California, April 7-10 1999. USENIX.

[LH01] Thomas A. Limoncelli and Christine Hogan. *The Practice of System and Network Administration.* Addison-Wesley, August 2001, ISBN: 0201702711.

[LRNL97] Thomas A. Limoncelli, Tom Reingold, Ravi Narayan, and Ralph Loura. Creating a Network for Lucent Bell Labs Research south. In *Eleventh Systems Administration Conference (LISA '97)*, page 123, San Diego, California, October 26-31 1997. USENIX.

[Oet98] Tobias Oetiker. MRTG - the multi router traffic grapher. In *Twelfth Systems Administration Conference (LISA '98)*, page 141, Boston, Massachusetts, December 6-11 1998. USENIX.