

DESIGNING
MANAGED
OPERATING
ENVIRONMENTS

@andrewgalka

IT SERVICES

ONE SIZE DOESN'T
FIT ALL



Unpack
NetBoot
Restore

WHAT'S not TO LIKE!

CAT
and
MOUSE

OS X SERVER

WHERE TO NEXT?



'Thin' provisioning

Mountain **Lion**

Test
TEST
TEST



Agility favoured
over
what we knew

www.uow.edu.au/~xagalka/macmoe.pdf

@andrewgalka

andrew_galka@uow.edu.au

ANDREW GALKA
University of Wollongong
andrew_galka@uow.edu.au

June 2012

DESIGNING MANAGED OPERATING ENVIRONMENTS

Creating highly agile managed operating environments can be difficult using traditional disk imaging techniques. This article discusses creating a managed operating environment through bootstrapping existing installs, thin images and utilising package deployment solutions to create environments that are flexible and highly adaptable to any release of the operating system.

The traditional approach to creating a standard operating environment (SOE) consists of three basic phases; installation, customisation and imaging. This is a linear process that is performed once and then repeated. It is problematic in that the human involvement of configuration inevitably leads to different steps to completion each time and SOE is created and results in slightly different configurations. The problem is amplified when multiple versions of the SOE to support slightly different requirements are factored in. This is a problem that open source software such as InstaDMG was created to solve – that is, to remove some of the inconsistencies of creating SOEs by hand. InstaDMG utilised a modular approach by installing OS X right into a disk image, then installed packages into the image. Advantages to this modular approach were immediately obvious; the image was created exactly the same each time the process was initiated. As a result different kinds of SOEs could be created using the same package versions, and in addition, the modular approach was reusable in the post-deployment lifecycle of the computer. The InstaDMG process appeared to tick all the right boxes to create a robust SOE that addressed the needs of the deployment phase of the machine lifecycle. It is post-deployment of the SOE that a managed operating environment (MOE) has been evolved to address.

A managed operating environment attempts to address the entire lifecycle of the machine, from deployment to decommission. That is, a process that addresses not only the initial deployment of the machine, but also the ongoing management. Being able to make a change to support a new technology or even a new philosophy to the entire environment, to all deployed computers, not only to newly provisioned computers, is a distinctive goal of the managed environment. The reusability component of the InstaDMG SOE workflow is the defining feature of a MOE.

While designing the managed operating environment of the University of Wollongong, we analysed the parts we didn't like of our existing traditional approach. First, handling each computer separately after deployment had become troublesome; machine specific builds meant the SOE was often playing catch-up; relying on NetBoot had the same issues as maintaining the SOE (as previously discussed); and finally once the computer had been deployed we lost visibility of the computer. Upgrading software across the existing SOE fleet was possible but highly ineffective. Resolving problems was on a computer-by-computer basis was especially pronounced in a lab environment when a computer was re-imaged. It proved impossible to get the newly re-imaged computers in-line with all of the post-deployment

modifications made to the rest of the lab. Updating the SOE to support not only updates to OS X, but also machine specific builds of the operating system, was a frequent occurrence. The same issue occurred during our deployment process via NetBoot. Additionally, Apple's retreat from the data centre played a part in reassessing OS X Server in the data centre. Our requirements demanded robust and reliable services, however the lack of lights out management and redundancy in Apple's hardware after the discontinuation of the Xserve was a problem. At the time, the upcoming release of Lion provided an opportunity to design a new method to resolve the issues identified.

After much team discussions we agreed upon the following workflow: i. boot the computer into the pre-installed factory operating system: ii. install a package that installs a first run script and the open source software deployment agent Munki; and, iii. add the computer to Open Directory groups where Munki can determine what software to install. We also employed thin provisioning for the computers that need to be re-imaged through an InstaDMG-like script. By using the factory installed operating system we no longer had to spend time updating images with hardware specific builds of the OS. Likewise it removed our reliance on NetBoot (and consequently updating the NetBoot set). Configuration of our base settings that includes slight network modifications and binding to Open Directory and Active Directory were completed by the first run script. Munki then took over to install the software. We created a dynamic manifest for Munki that was determined by the machines' directory memberships that defines software through inheritance. For computers that may be re-imaged later in the computer's lifecycle, we used the InstaDMG-like script that installs the OS to a disk image and performs minor configuration changes such as silencing the registration process. For a number of reasons, it was decided that our team write our own script (and not use InstaDMG). The script outputs a never booted up-to-date OS, deployed through the usual NetBoot and DeployStudio combination we had in place. The first run and Munki packages are also installed via DeployStudio and then the same workflow as new computers is undertaken.

With our new managed environment in place, Lion was made available to our community in October 2011 - just months after Lion's release date in late July. This was miles ahead of other similar organisations who were installing Snow Leopard on their computers up to as recently as April 2012. By utilising the Apple developer program we were able to comprehensively test and provide bug reports to Apple during Lion's development. Unfortunately the two month delay we saw releasing our MOE to the University community was due to a bug in dsconfigad while binding to Active Directory. We are currently 'on track' to release Mountain Lion to the University as soon as Apple releases it (July 2012), due to the highly flexible 'plug and play' nature of our MOE - that is plug in a new OS and let the first run script and Munki configure the computer into a managed state.

Moving to a managed operating environment has a number of advantages that we believe allows us to provide a fully featured desktop experience to our users. Our environment boasts the most current release: our users expect that when Apple releases an update, so do we. The 'plug and play' nature of the environment allows us to test before a major release and work to resolve issues. As a result, our team are no longer playing 'catch-up' when updating our systems. Finally everything is reusable throughout the entire lifecycle of the computer. Building with the same system that we maintain, allows our Help Desk staff resolve calls at the

first point of contact, where traditionally calls were escalated to second level support. Our future plans include moving software installation to 'user land', connecting it to our internal procurement processes. Moving from a standard to a managed operating environment has enabled highly agile and flexible services to our users.