

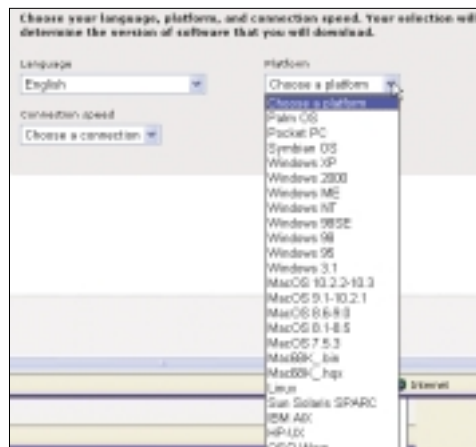
MULTILINGUAL TESTING ON MULTIPLE PLATFORMS

Roman Civin

As much as they need to be multilingual, software products need to qualify as multiplatform to truly provide for the global user. Take Palm OS 5, Linux Red Hat Advanced Server, Solaris 9, Windows XP or Mac OS X. Many questions and issues may come up when you are trying to achieve efficient testing of a product on some or all of these platforms. How do you prepare for efficient internationalization, localization and functional testing on software products that combine these platforms? This article can serve as a beginner's guide for testing a product on several platforms. It describes the concepts you need to pin down, preempts some common issues and provides tips, solutions and best practices for teams involved in multiplatform testing.

What is a platform? *Platform* can mean many things. For our context, we assume that platform is a unique operating system running on unique hardware. So with platform, you should always distinguish between Solaris on an Intel machine and Solaris on a SPARC workstation. Sometimes, the same platform can have different names. Windows NT 4 and Windows XP, for instance, can be considered the same platform, only different versions.

How can you manage multiplatform testing? The answer to efficient multiplatform testing is to have infrastructure, tools and processes that allow quick and repeated setup of your clean testing environment — your platforms. How to do this? One of the most widely used and easiest ways is to create



Choosing your language and platform

disk images of configured systems that are deployed (or *ghosted*, a term deriving from one of the tool names) on the testing machine. Ghosting is much faster than installing and configuring your operating system from scratch. You can create and save a sector-by-sector "image" of your tuned-up installed system in a file and load it back whenever needed.

Some of these tools are available as part of your system server (Mac Server) while some are utilities (Flash for Solaris). If you need to store hundreds of images for different platforms on several different file systems to be used by dozens of users simultaneously, tools such as Symantec PowerQuest Deploy Center or Symantec Ghost allow advanced server features such as multicasting sessions, remote deployment and server management.

The architecture used for platforms such as Windows and different versions of Linux as shown in the graphic "Architecture of testing management system" gives you a good idea of how these tools are deployed.

USAGE SCENARIO

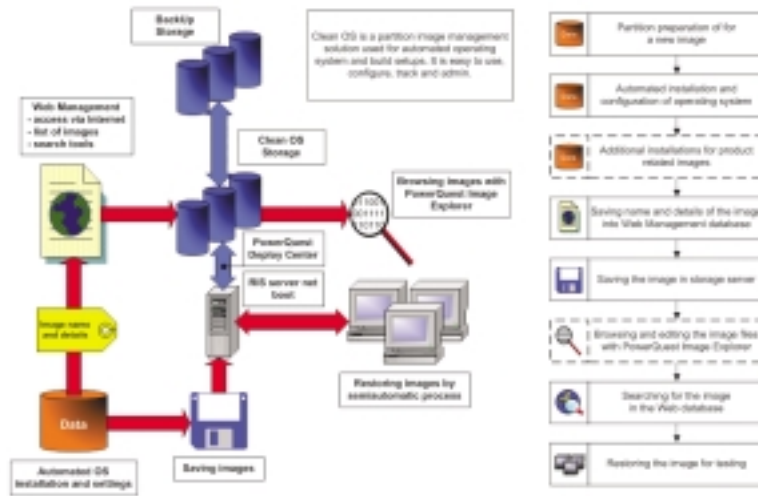
Imagine you are a test engineer and have three machines on your desk. You don't know what is installed on these machines, and you need a fast, clean start of your test pass. Assuming your project test platform images are ready on the server, all you need to do is turn on the machines, boot from

the local network to connect to the image server, select the image file you want on each machine and deploy them. Then you just need to restart, log on, run a script to change the computer name and add it to your testing domain, and you are ready to go — in less than 15 minutes.

Estimating setup time. We use these infrastructures, tools and processes to save time and ensure consistency in testing. So, right from the start, be sure you know the extent of platforms coverage that is required on your project or in your operations department. Of course, you'll have a good test plan. But you will also have to estimate how much time it might normally take you to install a clean system and how long it might take if you minimize the setup time by using an automated and enhanced preparation process. A lot of effort can be saved here, depending on the number of setups required for your test plan. To see the possible time savings, estimate how many times you will need a clean platform installed to perform all testpasses on it. The basic formula for the time that you'll need is "buildcount" multiplied by "langcount" multiplied by "testercount," where "buildcount" is the number of builds produced during the project, "langcount" is the number of languages that exist for these builds and "testercount" is the number of test engineers working on the project. You will need to multiply this result by the number of any specific configurations within your testing matrix (such as a file system or browser). The higher number you get, the more it pays to use a well-prepared disk image setup process. Based on your "setup" activity estimates, you can allocate the appropriate time required for the preparation of disk images. To give you an idea, it typically takes an hour to install and configure Windows XP, while loading and adjusting a disk image takes only 10 minutes.

Platform specifics — image properties. Each platform has its own specifics that are worth tracking and need to be identified before saving the disk image. Windows is a fairly well-known platform, so let's take a look at what is good to track on Linux:

- FileSystem: Ext2 and so on
- Kernel build: 2.2.1.3
- Distribution: Mandrake, Red Hat, SuSE, Debian and so on
- PC Type: GX260, GX270
- Language: Core language (English, French, Czech)
- Regional settings: English, French, Czech
- Created by: person who installed the system



Architecture of testing management system

Name	Storage System	OS Type	OS	Lang	Language	Reg. Settings	Assigned Product	Usability	OS	Platform
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	ENGLISH (FRANZOSISCH)	ENGLISH (FRANZOSISCH)	ENGLISH	Tablet PC	99%	99%	Tablet PC
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	ENGLISH (FRANZOSISCH)	ENGLISH (FRANZOSISCH)	ENGLISH	Tablet PC	99%	99%	Tablet PC
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	ENGLISH	ENGLISH	ENGLISH	Tablet PC	99%	99%	Tablet PC
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	FRANCOIS	FRANCOIS	FRANCOIS	Tablet PC	99%	99%	Tablet PC
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	FRANCOIS	FRANCOIS	FRANCOIS	Tablet PC	99%	99%	Tablet PC
Normal (QFT)088.P2G	QFT 03260	Win XP Tablet PC	SP 3	FRANCOIS	FRANCOIS	FRANCOIS	Tablet PC	99%	99%	Tablet PC

Image properties for a Dell GX260 Tablet PC

Version and type of Web browser: Mozilla, Netscape

Tracking disk images in one interface. You can use a script or, better, a universal java utility to extract specific system information from an installed platform and store it automatically as image tracking information for other users on a server. It is wise to implement an image

naming convention that is unique, easy to read and consistent. Using an automated image properties reading tool together with a Web-based interface for tracking images for multiple platforms can prove to be very useful,

as the Web tracking interface can be displayed from any platform (if designed properly). The graphic "Image properties for a Dell GX260 Tablet PC" provides an example of image properties for a Tablet PC using QASight's Clean OS image tracking application.

Test on appropriate browsers. Which browser support to cover across platforms is an important question that should be answered in your test plan. Each platform tends to have its "flagship" browsers — for example, Mozilla or Netscape on Linux and Mac, Internet Explorer for Windows. Make sure you identify browser versions and locales for testing. And, of course, in Web application testing, the browser is actually part of the testing platform.

Setting up older operating systems on new hardware. Sometimes, you will be asked to test on an older platform such as Windows 98 SE or Mac OS 9. It may be hard to configure your latest hardware with an old system when drivers don't exist for it. The best advice here is to use the hardware manufacturer's Web site to download the latest drivers. (Tip: when configuring a new image, it might save you time to paste your drivers' setups on the local logical drive beforehand, in case the network adapter does not work and you cannot connect to the Web.) In some cases, you will have to use older hardware if your test plan requires it. While this may seem like a "step backwards," keep in mind that there are many good reasons for testing on older hardware.

Storing a clean platform. A clean platform is usually a machine with the operating system set up and ready for testing. Apart from the software to be tested, it contains no additional installations (no other applications, desktop bundles or tools). All exceptions to this should be listed in the test

plan. Importantly, a clean machine contains no antivirus tools, as they sometimes interfere with the product to be tested. Clean also means having nothing other than the target locale support installed. Also, be careful about the variety of system security patches that are installed because they often change other products' behavior. Agree with your Test Lead and client about what is included as part of your clean system and what is not.

One of the projects completed by QASight recently included localization and functionality testing of an enterprise server product on Red Hat Advanced Server and Solaris 9 (both for Intel machines). Our project preparation checklist looked like this:

Have a valid test plan (defining what is covered and what is not covered).

Install the required system.

Configure all relevant properties (system settings, locales and drivers).

Install/add components, tools and utilities needed for testing and allowed on a clean system in accordance with the test plan.

Record all relevant properties to be stored for disk image tracking.

Sign, upload and protect the image on the server.

Test image health (usually includes reloading and system properties checks).

Repeat this for all images you need. ■

Roman Civin is the manager of Moravia Worldwide's testing and engineering QASight business units in the Czech Republic and China. He can be reached at romanc@qasight.com

