

## What can software vendors do to make the lives of system administrators a little easier?

BY THOMAS A. LIMONCELLI

# A Plea from Sysadmins to Software Vendors: 10 Do's and Don'ts

A FRIEND OF mine is a grease monkey: the kind of auto enthusiast who rebuilds engines for fun on a Saturday night. He explained to me that certain brands of automobiles were designed in ways to make the mechanic's job easier. Others, however, were designed as if the company had a pact with the aspirin industry to make sure there are plenty of mechanics with headaches. He said those car companies hate mechanics. I understood completely because, as a system administrator (sysadmin), I can tell when

software vendors hate me. It shows in their products.

A panel discussion at the Computer-Human Interaction for Management of Information Technology (CHIMIT) 2009 conference focused on a number of do's and don'ts for software vendors looking to make software that is easy to install, maintain, and upgrade. This article highlights some of the issues uncovered at that meeting. CHIMIT is a conference that focuses on computer-human interaction for IT workers—the opposite of most CHI research, which is about the *users* of the systems that IT workers maintain. This panel turned the microscope around and gave sysadmins a forum to share how they felt about the speakers who were analyzing them.

Here are some highlights:

**1. DO have a “silent install” option.** One panelist recounted automating the installation of a software package on 2,000 desktop PCs, except for one point in the installation when a window popped up and the user had to click OK. All other interactions could be programmatically eliminated through a “defaults file.” Linux/Unix tools such as Puppet and Cfengine should be able to automate not just installation, but also configuration. Deinstallation procedures should not delete configuration data, but there should be a “leave no trace” option that removes everything except user data.

**2. DON'T make the administrative interface a GUI.** Sysadmins need a command-line tool for constructing repeatable processes. Procedures are best documented by providing commands we can copy and paste from the procedure document to the command line. We cannot achieve the same repeatability when the instructions are: “Checkmark the 3rd and 5th options, but not the 2nd option, then click OK.” Sysadmins do not want a GUI that requires 25 clicks for each new user. We want to craft the commands to be exe-

cuted in a text editor or generate them via Perl, Python, or PowerShell.

### **3. DO create an API so the system can be remotely administered.**

An API gives us the ability to do things with your product you didn't think about. That's a good thing. Sysadmins strive to automate, and automate to thrive. The right API lets me provision a service automatically as part of the new employee account creation system. The right API lets me write a chat bot that hangs out in a chat room to make hourly announcements of system performance. The right API lets me integrate your product with a USB-controlled toy missile launcher. Your other customers may be satisfied with a "beep" to get their attention; I like my way better (<http://www.kleargear.com/5004.html>).

### **4. DO have a configuration file that is an ASCII file, not a binary blob.**

This way the files can be checked into a source-code control system. When the system is misconfigured it becomes important to be able to "diff" against previous versions. If the file cannot be uploaded back into the system to recreate the same configuration, then we can not trust that you are giving us all the data. This prevents us from cloning configurations for mass deployment or disaster recovery. If the file can be edited and uploaded back into the system, then we can automate the creation of configurations. Archives of configuration backups make for interesting historical analysis.<sup>1</sup>

**5. DO include a clearly defined method to restore all user data, a single user's data, and individual items (for example, one email message).** The method to make backups is a prerequisite, obviously, but we care primarily about the restore procedures.

**6. DO instrument the system so we can monitor more than just, "Is it up or down?"** We need to be able to determine latency, capacity, and utilization, and we must be able to collect this

data. Don't graph it yourself. Let us collect and analyze the raw data so we can make the "pretty picture" graphs that our nontechnical management will understand. If you are not sure what to instrument, imagine the system being completely overloaded and slow: what parameters would we need to be able to find and fix the problem?

### **7. DO tell us about security issues.**

Announce them publicly. Put them in an RSS feed. Tell us even if you don't have a fix yet; we need to manage risk. Your public relations department does not understand this, and that's OK. It is your job to tell them to go away.

### **8. DO use the built-in system logging mechanism (Unix syslog or Windows Event Logs).**

This allows us to leverage preexisting tools that collect, centralize, and search the logs. Similarly, use the operating system's built-in authentication system and standard I/O systems.

### **9. DON'T scribble all over the disk.**

Put binaries in one place, configuration files in another, data someplace else. That's it. Don't hide a configuration file in /etc and another one in /var. Don't hide things in \Windows. If possible, let me choose the path prefix at install time.


### **10. DO publish documentation electronically on your Web site.**

It should be available, linkable, and findable on the Web. If someone blogs about a solution to a problem, they should be able to link directly to the relevant documentation. Providing a PDF is painfully counterproductive. Keep all old versions online. The disaster recovery procedure for a five-year-old, unsupported, pathetically outdated installation might hinge on being able to find the manual for that version on the Web.

Software is not just bits to us. It has a complicated life cycle: procurement, installation, use, maintenance, up-

grades, deinstallation. Often vendors think only about the use (and some seem to think only about the procurement). Features that make software more installable, maintainable, and upgradable are usually afterthoughts. To be done correctly, these things must be part of the design from the beginning, not bolted on later.

Be good to the sysadmins of the world. As one panelist said, "The inability to rapidly deploy your product affects my ability to rapidly purchase your products."

I should point out this topic was not the main point of the CHIMIT panel. It was a very productive tangent. When I suggested that each panelist name his or her single biggest "don't," I noticed the entire audience literally leaned forward in anticipation. I was pleasantly surprised to see software developers and product managers alike take an interest. Maybe there's hope, after all. 

#### Related articles on [queue.acm.org](http://queue.acm.org)

##### **Error Messages: What's the Problem?**

Paul P. Maglio, Eser Kandogan  
<http://queue.acm.org/detail.cfm?id=1036499>

##### **Facing the Strain**

Kode Vicious  
<http://queue.acm.org/detail.cfm?id=1160442>

##### **A Conversation with Phil Smoot**

<http://queue.acm.org/detail.cfm?id=1113332>

#### **Reference**

1. Plonka, D., Tack, A. J. An analysis of network configuration artifacts. In *Proceedings of the 23rd Large Installation System Administration Conference* (Nov. 2009), 79–91.

**Thomas A. Limoncelli** is an author, speaker, and system administrator. His books include *The Practice of System and Network Administration* (Addison-Wesley) and *Time Management for System Administrators* (O'Reilly). He works at Google in New York City.

#### **Acknowledgments**

I would like to thank the members of the panel: Daniel Boyd, Google; Aileen Frisch, Exponential Consulting and author; Joseph Kern, Delaware Department of Education; and David Blank-Edelman, Northeastern University and author. I was the panel organizer and moderator. I would also like to thank readers of my blog, [www.EverythingSysadmin.com](http://www.EverythingSysadmin.com), for contributing their suggestions.

© 2011 ACM 0001-0782/11/0200 \$10.00