
[Why learning the sysadmin job may make you a better developer \(http://www.sandon.it/node/81\)](http://www.sandon.it/node/81)

Posted on: Thu, 08/22/2013 - 00:08 **By:** Idsandon

Many developers learnt first to code on isolated machines - DOS ones if they are old enough - or within simple networks. DOS was a single process, single thread environment, almost always on isolated machines. Maybe a modem to connect to a BBS, LAN were much rarer. That meant developers had little to care about "the environment" their application where "living in". Once the application was started, it was the only one running (but some tricks like TSR) and could use all the resources available. Slowly, the environment changed. First Windows introduced multitasking (cooperative, then preemptive), applications were no longer running alone, and could no longer consume all the resources. Then LANs started to make the environment more complex. Then came domains and directory services, then the Internet. But often too many developers look to ignore all that. Often applications are still coded if they are run alone on an isolated machine by a highly privileged user, or connected directly to the Internet.

My experience in the past years taught me that understanding - and learning - the sysadmin job may make you develop better applications. When you start to understand all the implications of an application running in a large, complex network, together with many other applications, you are forced to think at development from a different perspective. For example, you start to understand that not any application may run with administrator elevated privileges, because any sensible sysadmin won't allow any user to be an administrator. You start to understand you have to write global and per user data files (including configurations) in the proper places, because a sysadmin may use roaming profiles, and a machine may be used by more than one user. You start to understand you can't force a sysadmin to manage n user databases because each application uses its own, and managing them becomes a nightmare. You start to understand your application has to play nicely with other applications running on the same machine - you may not know which ones - and can't think to be the only one and use resources at will. You start to understand that if your application integrates well with the underlying OS, it can take advantage of many advanced services it offers today. And finally, you start to understand that communicating over a network requires to understand how a network works - sometimes some features are transparent to your application, i.e. a VPN, IPsec or a firewall, sometimes they are not, like a proxy.

I received some critical comments to my previous post "Embarcadero and IT Security, still an oxymoron". It was not intended as a personal attack. It was just my surprise about the perception of proxies not as useful tools to protect a network, but as an evil tool used by hackers or spammers to hide. Yes, if you just look at them from a user or developer perspective, they are just an obstacle - "oh my god, I've to configure it, I've to authenticate, I can't see the source IP, I can't use Facebook at work, etc. etc.". But if you look at it from a network administrator, you see it as a very valuable tool to manage internet traffic, protect your users (today proxies can perform a lot of checks on traffic) and your network from threats, thereby you'll deploy them - probably not simple transparent ones. When you understand it, you know you have to make your applications fully proxy-aware if they need to communicate over the Internet, and when you understand you can't ask every user in a large

organization to configure proxy settings and authentication in every application, you know you have to use OS facilities to get the proxy settings - usually delivered automatically using some network service - and authenticate against it, if needed, probably using automatically the user credentials because you know any sensible sysadmin will force user to change passwords every n days, will integrate the proxy with the directory authentication and authorization services, and users don't want to update passwords in each application they use every time the password expires.

This is even more important if you are a development tools company, because you have to fulfill your users' needs, especially when you sell expensive tools at prices well beyond the hobbyist range. The proxy one is a very simple example, but not surprisingly, the first release of the "new" Datasnap lacked proxy support. Its developers looked to be unaware of a fundamental requirement for any application running in any well designed network. When you're in charge of the architecture and/or design of an application, today you have to broaden your view beyond the internal application functionalities, and think about how it will integrate with the whole "environment" it has to work within. There are not only the user needs, but also the needs of the personnel who has to deploy and maintain it, and you have to ensure it works properly and maintenance efforts are minimized, because - again - it may not be the only application needing maintenance.

Don't work in a silo, and spend some time to understand your sysadmin goals and needs, and if you can, learn to be a sysadmin yourself. You'll become a better developer.

- [Log in \(http://www.sandon.it/user/login?destination=/comment/reply/node/81/comment_node_blog%23comment-form\)](http://www.sandon.it/user/login?destination=/comment/reply/node/81/comment_node_blog%23comment-form) to post comments

Source URL: <http://www.sandon.it/node/81>