

# Prep your patch policy

Mark D Scott

I WAS RECENTLY working with a client when we ran into an interesting service pack issue. The IT team decided to apply SP2 to its SQL Server production servers. They did what one would expect from a responsible team: they applied the service pack to the servers in the test environment to ensure that the applications that used SQL Server did not break. Everything seemed to test out fine. They notified everyone in the organisation and scheduled the patch to be applied to the servers.

Things went into place in production without a hitch. They continued to distribute the patches to the servers in stage, test and finally development. A textbook deployment. Until one of the developers tried to open his Reporting Services project. Suddenly, he had myriad quirky little problems. Reports would open, but parts of the reports would be unavailable. Another developer opened an Analysis Services database. No problem. Until he tried to open the Calculated Members tab. Something was definitely broken.

As you may have guessed, the issue was that while the servers had received the service pack, the developers had not. Since the SQL Server client tools installed on the desktops were considered part of the desktop deployment and not the server deployment, they were not considered in the upgrade. And distributing the service pack to the developers involved another round of testing and setting up the distribution. (We will not even mention the fact that not all the developers were put out of commission – which meant that some had already applied an out-of-policy service pack to their desktop. That is another article altogether).

While this mess was being straightened out, maintenance on some of the analytic applications used by the company ground to a halt. The event brought to the fore how inconsistent and unclear the policy was on service packs. It also demonstrated some frac-

Unfortunately,  
sometimes we  
become as siloed  
in our thinking as  
the applications we  
shepherd

tures in interdepartmental communications. Distribution of service packs is very important because it keeps the ecosphere more secure.

Some organisations resist the application of service packs, fearing the strife and confusion of distributing the software more than the vulnerabilities the service pack can help overcome. Other organisations have no policy at all, allowing individual users and system administrators to apply or ignore patches as they see fit. This creates its own maintenance nightmares as patches appear on one server but not another. Some are fully protected while others remain at the same state as when they were installed.

Most larger and/or more sophisticated IT organisations have read the infrastructure optimisation models and realise that they should be proactively establishing policy and implement-

ing procedures that ensure this type of thing does not happen. They know that they can automate many of these issues and prevent this kind of gaffe. However, at the end of the day and at the edge of the tools lies the fact that people need to understand that what they do affects other people and other departments. And then they need to communicate with one another.

It is easy to think of the ways a service pack will affect *your* domain, to consider the subset of things that touches your systems, and to cover your own bases. It is more difficult to think outside your private box and consider the effects of the change outside of your realm. Sometimes we become as siloed in our thinking as the applications we shepherd.

The good news is that this event was relatively easily corrected. The organisation did have a mature model. Once the issue was identified (again, somewhat confused because the problem only existed for the developers that played by the rules), the patch was quickly (and automatically) distributed. A fresh dose of clear communications and a well-designed system helped correct the problem with minimal downtime or disruption.

It seems that the answer lies in allowing our infrastructure, and ourselves, to mature. An effective service pack policy addresses procedures, products and people. ■

---

**MARK D SCOTT** is a senior consultant with Microsoft Consulting Services. He works closely with clients to help design and build large-scale, data-centric applications. You can reach Mark at [granddaddy2002@msn.com](mailto:granddaddy2002@msn.com) or [mascott@microsoft.com](mailto:mascott@microsoft.com).