

## Proactive Maintenance for Oracle E-Business Suite

As a remote services provider, often we are often asked to provide proactive maintenance support for clients. For the purposes of this discussion, I'm going to separate operational support from maintenance support.

Operational support surrounds management of the application operations, as they exist. For example, if I'm running 11.5.10.2 on 10.2.0.3, operational support would include availability monitoring, capacity planning, backups, and other tasks that support the reliable operation of the existing system. If I let patching slide, I can still provide operational support, but I'm not exactly maintaining the applications very well. Maintenance is certainly a component of operations; if something breaks, you fix it, but break-fix maintenance certainly can't be called proactive.

Proactive maintenance, on the other hand, is about keeping things up-to-date in order to avoid operational problems and provide incremental improvements. A proactively maintained system may still break, but it is more likely to have a fix available if the maintenance plan is well thought-out and well executed. A proactively maintained system is also less likely to need major overhaul, as it never gets too far behind.

Everyone knows that if we eat well, get plenty of rest, exercise regularly, and avoid (or at least minimize) damaging habits like smoking and drinking too much, our bodies are more likely to live longer and more reliably. An apple a day, as the saying goes. Why does this work? Because we are taking care of the core functions that power our internal engine. Without a functioning body, we cannot work, play, think, or do much of anything. Modern exercise routines carry this philosophy to a deeper level: Core training is currently a popular approach to fitness. Keep the core of your body strong, and the rest will follow.

Operationally speaking, we humans really only have to eat and go to the doctor when something breaks. But over time, that approach will lead to more and more broken body parts and more and more onerous fixes, which isn't a great way to live.

We can easily apply the same maintenance philosophy to Oracle E-Business Suite: Keep the core functionality current and strong, and the rest will follow. The core components for E-Business Suite vary slightly from install to install depending on what is most important to the users; however, there are some constants (in no particular order):

- ATG family pack
- TKX autoconfig suite
- Java (jre) version
- Java plug-in
- IZU diagnostics suite
- AD utilities
- Developer6i version and patchset
- iAS version and patchset (including apps, disco, OID, etc)
- Database version and patchset
- Quarterly Critical Patch Updates (CPU)

Why this particular group? Because these are the core components that the rest of the applications rely upon. The ATG family pack affects user management, application security, Workflow, concurrent managers, the applications framework, and a host of other components that all application modules rely upon. Likewise, the ATG family pack depends on the java version, the Developer 6i patchset, and the operating system level in some cases.

Individual components such as AD, IZU, and TKX are management utilities that need to stay as up-to-date as possible for better management of the E-Business Suite ecosystem as well. These products not only provide greater administrative functionality and convenience, but are often pre or co-requisites to larger patching efforts such as an ATG family pack or even a product family pack.

Finally, as the foundation of E-Business Suite, the database release and patchset is critical to the overall health of the system. Database releases are fewer and farther between than application patches, but by their nature they affect all components and should be evaluated just as closely in a proactive maintenance plan.

Conspicuously absent from this list are product family packs or product patches (i.e. AP, GL, etc) in general. A good maintenance plan should not start with the product components, but should start with the core. The reason for this is simple: Proactively patching application family packs or application modules will force core application patching. Applying the latest financials family pack will likely require a later ATG release, which will require a later Developer 6i release, which will require a later java release, and on and on. Applying the latest financials family pack to a system with no core maintenance will absolutely take longer, require more testing from more application users, and require a longer outage than a system that has been deliberately and intelligently managed by keeping core components up-to-date.

Forced core application patching is not a strategy; it is the absence strategy. Forced core application patching leads eventually to that expensive, time-consuming, understaffed, poorly tested uber-patching project that inevitably results in extended downtime, performance problems, or loss of some functionality somewhere in the application. Why? Because applying 50 or more patches at the same time (as amazing as adpatch is) is not a good idea if you can avoid it; E-Business Suite contains so many moving parts that the odds of a problem increase with the number of patches applied at once. Large, forced application patching exercises all end with the Apps DBA wedged in between the users and Oracle support trying to figure out what some ambiguous java error is all about.

And speaking of Oracle support, I also included in my list the quarterly CPU updates; keep in mind CPU updates require a system to be N-1 from the latest component release for the patch to work at all. For E-Business Suite this generally means the ATG family pack and the database patch level. Keeping up-to-date on core patching keeps the very, very expensive software support we all pay for relevant: A critical patch that cannot be applied because the system has not been kept up-to-date is a waste of support dollars and a potential security risk.

So specifically, what is a good baseline strategy? A good core maintenance strategy has a few common characteristics:

- The core comes first.
- Intelligence and relevance is applied when deciding what components to patch.
- User testing is well defined and minimized.
- Downtime is minimized.

If normal system operations will be interrupted, we want the biggest and safest bang for our buck.

Quarterly

- Database patchset (not version upgrade)
- Database CPU patch
- AD utilities

- IZU utilities
- TKX utilities

Every six months

- Java version
- Application CPU patches
- Developer 6i release

Annually

- ATG family pack
- iAS version
- Java plug-in version

Quarterly patching should be short, as unobtrusive to the users as possible, and provide maximum benefit to the administrative staff. I like to think of this as the “administrative” patch release. By keeping the database release, AD, and diagnostic (IZU) utilities updated every three months, you will streamline interaction with Oracle support, enable time-saving features that may be leveraged across all E-Business Suite maintenance (Remember when patch merging and non-interactive patching were introduced?) and provide the latest support for cloning and configuration management. Most importantly, the burden of user testing for these applications is light; application functionality does not change much if at all with these patches.

The semi-annual maintenance patch bundle is almost more of a preparation for the annual patching. With the exception of the application CPU patches, this patch bundle does not require much user testing and provides bug fixes, performance improvements, and reduce downtime and complexity for the annual patching. Though I recommend application CPU patches every 6 months, CPU patches should be evaluated at every release; if a CPU patch is determined to be critical to your environment, apply it asap. If your core components are up-to-date this should not be a big deal.

Annual patching is where the users get involved. Because extensive user testing is required and users are busy doing their regular jobs, we need to make sure that their time is not wasted. The ATG family pack is large and will affect core components like user management and workflow, which means business processes must undergo regression test (right?!). Likewise, the iAS version (e.g. Discoverer 4i to Discoverer 10g) will require extensive user testing. The Java plug-in update is a big pain for large, distributed environments and should be somewhat infrequent unless there is a critical security requirement.

This is, of course, only a baseline suggestion. Each organization will have different downtime tolerances and availability of testing resources. Outsourcing this maintenance to a remote DBA firm or a trusted, experienced consultant is always an option; those who have done it before perform core maintenance work most efficiently. The point is to find a plan that works for your organization and stick with it. Adjusting over time is of course necessary, but establishing the habit of regular core maintenance will improve E-Business Suite’s long-term health.