

# “Packing Tips” for Supporting Legacy Applications Secretary of State Office Systems

---

## The Hiking Trip Story

Imagine going out on a hiking trip in an area infiltrated by poisonous snakes and slippery slopes next to steep cliffs. If you’re prepared with well-constructed hiking boots, know how to stay on the trail, and use an experienced guide or accurate map, you might be fine and even enjoy the scenery. But if you’re cutting corners and wishfully thinking you’ll see the trail markers in advance, you might be in a world of hurt. Serious hurt.

The analogy parlays perfectly to supporting enterprise applications known for their complexity and pitfalls. Some applications are like the neighborhood street: new, perfectly lit, well-marked and potentially safe from disastrous risks. And for these, you can navigate the excursion at a low cost and with a smart, less experienced team.

But for those legacy applications that are full of hidden “snake pits,” prudence is recommended for the sake of system stability, security and functionality. This is especially true of enterprise-level applications, such as some of those currently in use in several secretary of state offices around the

country. Often, these applications have evolved over a decade and are based on competing interests and lack of long-term funding or planning. Due to their antiquated code, along with their suspect development, test and deployment processes used along the way, managing these applications without skill and experience is like hiking through the slippery snake pit blindfolded and barefooted. The question is not whether one will get struck, but when, how and for how long.

### ***Real Life Snake Pit Examples***

- ❖ **Hardcoded passwords.** These are often embedded throughout the application and/or registry, which significantly increases the risk of a security penetration unless the support approach is meticulous, pervasive and based on experience
- ❖ **Code duplication.** This may become an issue if a change is made in only one location, forcing the developer and tester to have prior knowledge of the other locations. Not knowing these in advance leads to inconsistent code and errors, sometimes very subtle but potentially costly errors, when any changes are made.
- ❖ **Incomplete or outdated documentation.** Sometimes a bad map is worse than no map at all. Some documentation might exist that refers to executables and other files that may not actually exist. This may lead to false hope if the system crashes or when an error surfaces.

## **Six “Packing Tips” for the Enterprise Support Mission**

Before signing up for support for your mission-critical application, make sure the vendor of choice has the following six items fully covered:

- ☑ **Knowledge of the Code Base** – There is no substitute for experience in the code. Only by implementing and supporting the code in multiple scenarios do you start to understand the pitfalls and dangers within the application. Complete end-to-end experience with some applications is a must for

functioning deployments, as not knowing all the places to change settings or hardcoded text in places like encrypted client registries, database tables and web services will lead to non-working software. Further, because the development and QA environments are usually separated from development tools, troubleshooting errors can be quite challenging, even for seasoned professionals.

- ☑ **Experience in the Domain** – By understanding best practices and emerging issues within the domain, such as redaction and potential legislation (e.g. §.569), one will be able to proactively ask the right questions and implement intelligent solutions that add value to the team needing the application.
- ☑ **Repeatable QA Experience** – An established quality assurance practice will help ensure that code and configuration changes do not result in undesirable side-effects—such as new bugs, or breaking existing functionality.
- ☑ **Deployment Expertise** – There is art, science and discipline related to supporting enterprise applications, especially as it relates configuration management and effectively deploying an application from development, to test or QA, and then to production. This is especially critical when multiple technologies—including VB6, ASP, ASP.Net, VB.Net and C#, some of which have more than one technology framework—have all been used and are intertwined.
- ☑ **Effective Documentation** – Enterprise-level providers are adept at creating and maintaining the right amount and type of documentation to properly support and sustain an enterprise-level application, especially one with complexities. They're used to having others read and rely on their guides, which have to be easy to read, precise and useful.
- ☑ **Comprehensive Support Model** – Issues will develop with any application, and the key to resolution is quickly troubleshooting where the issue manifests itself. Only by having a team in place that understands the entire enterprise stack—running from the network, through the hardware, to the operating system, and into the code and database, and how they all integrate—can you troubleshoot issues and fix them in an efficient and timely way.

### Moral of the Story

Nobody wants to be bitten by a snake or fall off a slippery slope onto the jagged rocks below. Unfortunately, sometimes it takes an actual snake bite to fully appreciate the particular pain that occurs, and the same is definitely true with managing enterprise applications in a Secretary of State office. Hopefully, by following the “packing tip” steps outlined above, you can minimize the pain and ensure the system is fully supported, maintained and continuing to add value to the busy team in the SOS office.