WHITE PAPER

How to Say Goodbye to Windows Server 2003

By Don Jones
Principal Technologist
Concentrated Tech

FLEXERA® SOFTWARE

TABLE OF CONTENTS

FLEXERA® SOFTWARE

Windows Server 2003 remains in widespread use in organizations of all sizes. However, in the decade since its invention, an incredible array of changes has occurred in the IT industry.

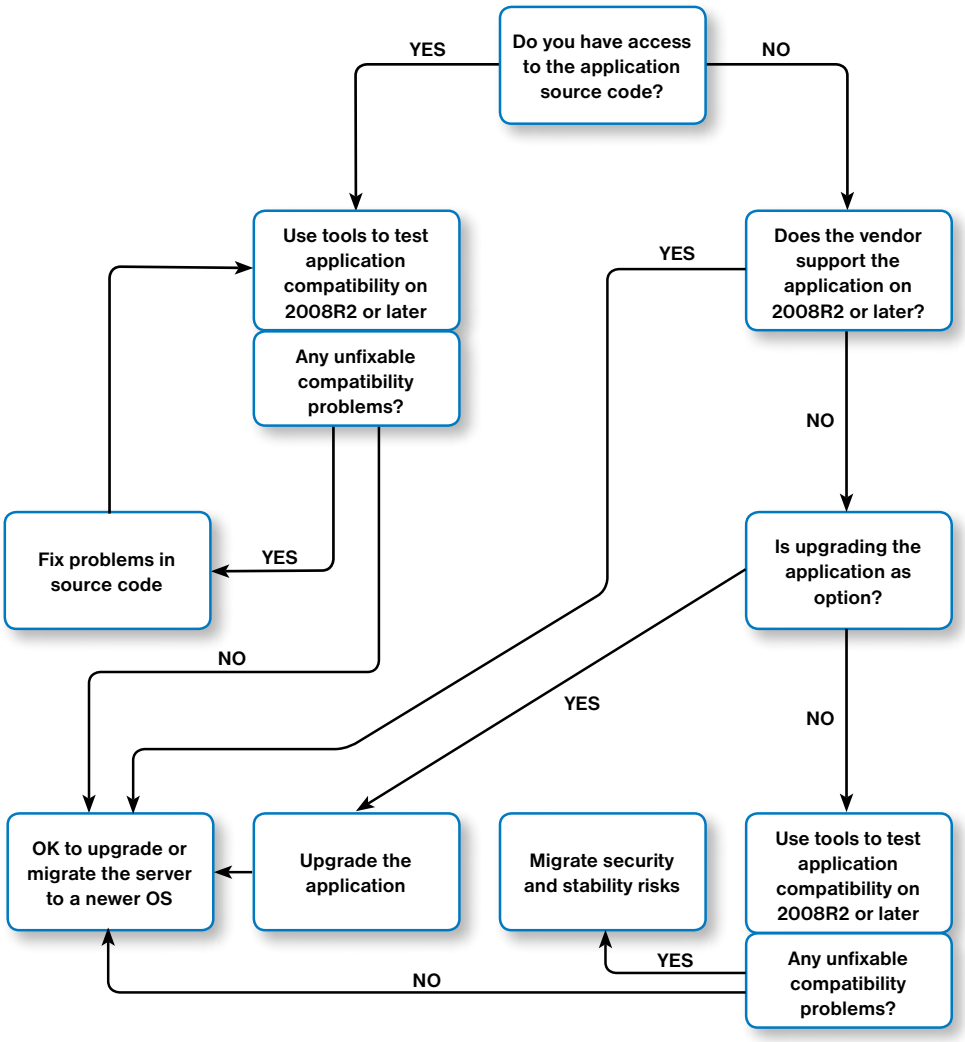
In July 2015, Microsoft's venerable Windows Server 2003 operating system will reach end-of-life, meaning the company will not provide regular support or updates for the operating system outside of an expensive custom support contract.

Windows Server 2003 remains in widespread use in organizations of all sizes. However, in the decade since its invention, an incredible array of changes has occurred in the IT industry. New forms of attacks, new perspectives on security, new approaches to management and automation, and more are all here. Windows Server 2003 will never be able to take advantage of these advances, or adequately protect itself against these risks.

Organizations need to adopt a repeatable, sustainable pattern for dealing with operating system obsolescence, on both client and server computers. In this paper, we will analyze the components of such a practice, and detail the capabilities that organizations will need to develop or acquire. We will present this process as something that organizations should conduct on a regular basis, and make an argument against treating operating system obsolescence as an interrupt-driven task that is only considered once every several years.

THE WIN2003 EOL DECISION TREE

In dealing with server operating system obsolescence, you have a straightforward chain of questions to ask and decisions to make. This decision tree illustrates those:



Applications should be prioritized, so that the applications most important to your organization are considered first.

This decision tree should be used to examine each server-based application in your environment. Applications should be prioritized, so that the applications most important to your organization are considered first. Also prioritize applications that face, or are accessed by, the public or by external customers or partners. Public-facing applications face significantly higher security risks, and should be considered primary candidates for server OS upgrades or migrations.

CONSIDERING IN-HOUSE APPLICATIONS

For applications where you have access to the source code, you have the most flexibility. Begin by assessing the applications' compatibility with a newer server OS, such as Windows Server 2008R2, Windows Server 2012, or later. Ideally, try to evaluate the application against the newest possible OS version, so that the resulting server will have the longest possible life after the upgrade or migration is complete.

Actually testing applications for OS compatibility can be an arduous process. While it is conceptually straightforward to deploy a virtual test environment and run the application through a gauntlet of functional tests, it's easy to miss specific functional scenarios, or to incompletely test the application. Instead, you should consider acquiring tools and solutions designed specifically for application compatibility testing and packaging needs.

Compatibility testing solutions can also markedly reduce the amount of time needed to evaluate an application.

These solutions are designed to statically and dynamically evaluate applications, and rather than simply testing them for functionality, the solution analyzes the applications for known compatibility problems. In some cases, these problems can be automatically mitigated through the use of OS-level compatibility adjustments or "shims;" in other cases, the solution may present an incompatibility report. These reports make it easy to assess the actual effort involved in recoding the application to be upwardly compatible; with such a report, you can make a more informed decision about what to do with the application.

Compatibility testing solutions can also markedly reduce the amount of time needed to evaluate an application. Rather than having to test the entire application's range of functionality, you can simply use the tool to look at known problem areas for compatibility. Many applications, particularly those developed in recent years, will likely be able to move to a newer OS with few or no problems. In some cases, organizations may simply need to repackage applications into a more modern installer in order to obtain forward compatibility.

Endpoint: Move the Application Forward

If you have no compatibility problems, or are able to resolve them using an acceptable amount of effort, then the application can either be deployed to a server that runs a newer OS (and the old server

decommissioned) or, when available, the server where the application is currently deployed can be upgraded.

Many environments prefer a new deployment of the application to a new server. This approach lets you take a slower, "side-by-side" approach to the migration, providing more options for rolling back, pre-deployment testing, and so on.

Endpoint: Do Not Move the Application Forward

In cases where you cannot, or do not have the resources, to address compatibility issues, the you may elect to not move the application forward to a newer server OS. This decision can also be driven by a need for the application to interoperate with specialized hardware, which itself is not compatible with a newer OS.

Many environments prefer a new deployment of the application to a new server.

In this case, you will need to mitigate the risks to the server. We will discuss server mitigation later in this paper.

CONSIDERING VENDOR-SUPPLIED APPLICATIONS

Your choices are obviously more limited when you do not have the ability to modify an application. When the vendor explicitly supports the application on a newer OS version, then your choice is easy: you can move the application forward to that newer OS version, either by deploying the application to a new server or by upgrading the existing server (when that option exists).

In some cases, the vendor may not support your version of the application on a newer server OS, but instead makes an upgraded version of the application available. You will obviously need to asses that upgrade, as it may involve a deployment process, additional expenses or license acquisitions, and so on. It is generally preferable to upgrade when possible, but there are also circumstances when it may not be possible. For example, upgrading an application may also require an expensive upgrade of specialized hardware, which may exceed your available resources.

When an upgrade is not available or is not possible, or when the vendor simply does not offer a version that can run on a newer server OS, then

you are not necessarily "stuck." You can still use compatibility testing solutions, as discussed in the previous section of this paper, to analyze the application and see if any compatibility problems exist. In some cases, existing problems may be subject to mitigation by the compatibility testing solution, and you can decide whether or not to move the application to a newer server OS without the vendor's blessing. In those situations, discuss with the vendor what "not supported on that OS version" actually means.

For example, you always have the option of maintaining a deployment of the application on an isolated virtual machine running the older, "supported" server OS version. If problems arise in production, you can try to duplicate those problems on the older, isolated virtual machine. If you can reproduce the problem there, then the problem may not be due specifically to the server OS version, and the application vendor may still provide corrective support.

If you find yourself unable to move an application forward, then you will need to continue running the application on Windows Server 2003.

Once you have completed your analysis, you essentially have two choices: move the application forward to a newer server OS, or decide to not move it forward.

MITIGATING SERVER RISKS

If you find yourself unable to move an application forward, then you will need to continue running the application on Windows Server 2003. In that case, you will need to make a careful risk assessment, and mitigate what risks you can, as completely as possible.

When possible, run the server in a virtual machine. Today's live-migration and other high-availability options of your hypervisor will provide a level of stability that Windows Server 2003 alone cannot usually match. Virtualization also provides more flexibility in terms of protecting the network communications to and from the server, to help mitigate against specific kinds of attacks.

Work with your anti-malware software vendor to ensure that the server will continue to be supported by their product. Continue to keep the anti-malware software on the server updated, to help guard against malware-based attacks.

Begin to severely restrict communications with the server. Use either an external firewall, or a local software firewall that continues to be updated, patched, and supported – meaning Windows Firewall is not a good option. Tightly control both incoming and outgoing communications. When possible, restrict not only the TCP or UDP ports used, but also the IP addresses with which the server may communicate. If possible, implement IPSec security so that the server can communicate only with known hosts, and only over secured, mutually-authenticated connections.

The biggest risk of running an end-of-life operating system is that attackers will have discovered exploits that have not yet been seen in the wild. These "held" exploits enable attackers to wait until the operating system vendor (Microsoft, in this case) will no longer patch these exploits, giving the attackers more time and flexibility to use those exploits. One way to combat this situation is to tightly regulate communications with the server.

The biggest risk of running an end-of-life operating system is that attackers will have discovered exploits that have not yet been seen in the wild.

An additional approach is to move all applications and services off the server, except those which you have decided to not move forward to a newer OS. Fewer "moving parts," or software components, means fewer potential exploitable elements for attackers to target.

Continue to re-evaluate your position on applications on a regular basis. As the server OS gets older and older, you may come to a point where running the application "unsupported" on a newer OS version is preferable to the risks inherent in running the older OS well past its end-of-life. Again, compatibility testing solutions can help you assess the actual functional risks of moving an application forward, regardless of the application vendor's stance on the subject.

DEVELOPING AN APPLICATION READINESS PROCESS

We have come to an age of more-frequent upgrades to operating systems. Microsoft has begun releasing smaller new versions of Windows on an almost annual basis, meaning it will become easier to fall further behind. In most ways, however, these regular releases are each less impactful than the operating system releases we have been accustomed to.

Most organizations deal with operating system rollouts on an interrupt-driven, once-every-several-years basis. That approach has worked well when Microsoft was releasing a new OS once every few years. Most organizations have also leaned toward mass upgrades, in an attempt to have every computer running the same operating system version, as much as possible. Neither of these organizational approaches are suitable for the current state of the IT industry.

Server operating systems should be considered to have a shorter "shelf life" than in the past – three years is likely a good maximum number. That means organizations need to adopt a number of practices, and acquire new capabilities, that make server upgrading less interruptive, and more of a regular, ongoing maintenance or "application lifecycle" activity. Specifically:

Server operating systems should be considered to have a shorter "shelf life" than in the past – three years is likely a good maximum number.

- Your organization needs the capability to continually analyze applications for compatibility with the newest OS versions. When applications exhibit no compatibility concerns, the default action should be to move them forward to the newest server OS. This capability, referred to as an application readiness process, helps ensure that the ability to analyze applications and move them forward is an ever-present part of the environment.
- Your OS licensing scheme should lend itself toward both constant upgrades, and the ability to run as many virtual servers as needed, so that applications can each be separated into their own virtual machines. This approach creates less cross-application dependency, and lets you deal with each application individually.
- Your goal should be to deploy new server OS versions as you have deployed service packs in the past. Allow a brief period after a version release for the industry to "settle in" and find any obvious problems, and then quickly deploy. Application compatibility analysis must be easy and quick, in order to facilitate this perspective.
- New software acquisitions should be conditional upon the vendor providing continual support for new operating system versions. Pressure vendors to specify that support in their maintenance agreements. Vendors must help you facilitate a constantly-moving-forward approach; use competitors and other purchasing pressures to guide vendors to a position that helps your organization.

CONCLUSION

Organizations should be careful not to let the end-of-life of Windows Server 2003 come as a surprise. A careful inventory of servers and their applications is necessary, and a plan should be developed to move applications forward to a newer, supported OS whenever possible. The process you will go through for Windows Server 2003 should serve as a model for making the "new OS move-forward" process a continual, repeatable part of your IT organization.

Organizations should be careful not to let the end-of-life of Windows Server 2003 come as a surprise.

Solutions that can help automate application compatibility assessments can become a valuable part of that continual process. OS end-of-life is going to be a more frequent occurrence going forward, and having the ability to quickly assess applications – and in some cases automate compatibility problem mitigation – will become a standard part of IT life.

Working with software developers and software vendors can also help make new applications easier to move forward. Ensuring that everyone understands, "OS versions will change, and will do so frequently," helps create a culture of moving forward, with all stakeholders understanding that simply leaving applications on old OS versions is not an option. ■

Don Jones is a 12-year industry veteran, author of more than 45 technology books and an in-demand speaker at industry events worldwide. His broad technological background, combined with his years of managerial-level business experience, make him a sought-after consultant by companies that want to better align their technology resources to their business direction. Jones is a contributor to TechNet Magazine and Redmond, and writes a blog at ConcentratedTech.com.

FLEXERA SOFTWARE