

Server Monitoring: Centralize and Win





Table of Contents

Introduction.....	2
Event & Performance Management.....	2
Troubleshooting.....	3
Health Reporting & Notification.....	3
Security Posture & Compliance Fulfillment.....	4
TNT Software's Solutions Augment Native Controls with Critically Necessary Capabilities.....	5

Microsoft develops robust operating systems for networked systems, but their OSs store locally all of the event and systems management information. This local storage works well when a business' computing environment is comprised of a handful of systems. But most environments require many more than a handful of systems to host their computing needs. Along with this expansion of network services comes an equivalent expansion in IT responsibilities for monitoring, managing, and securing those systems.

The daily "care and feeding" of desktop and server systems can be a tedious and time consuming process. Native tools onboard OSs like Microsoft Windows have the capability of logging dozens or hundreds of events everyday. Adding security, auditing, and performance information to these logs adds an additional order of magnitude to the volume of captured data. All of this must be managed and reviewed on a regular basis to look for anomalies and trend performance and health on each managed system.

One of the biggest problems with native tools is that they're limited in what they can do. Windows event logs, performance logs, and resource utilization information all store data locally on the individual system. The native user interface to view and manipulate this data often allows for the analysis of only a single system at a time. Thus, having ten servers in an environment means ten places to watch. Fifty servers mean fifty touch points. Problem analysis using native tools alone is challenging. Longitudinal or time-based analysis across multiple systems involves the use of multiple consoles in multiple windows.

Because of these limitations to native interfaces, the daily management and maintenance on individual systems grows unwieldy as the number of systems increases. With this increase, the need for third party tools grows from merely necessary to absolutely critical for efficient system management. These third party tools assist in the gathering, visualization, reporting and monitoring of system data. They also enable a number of additional benefits such as:

- Enhancing event and performance management
- Improving troubleshooting
- Centralizing reporting and simplifying report generation
- Improving the overall environment's security posture
- Enabling the environment to properly meet governmental and industry-specific compliance regulations.

As your computing environment grows to support your business needs, you may already be experiencing the challenge of relying solely on native tools. By not implementing solutions to monitor event logs, application logs, performance metrics, and network device status, you are putting your environment at risk. Let's take a look at several of these necessary capabilities and discuss how **not incorporating** effective centralized management and monitoring tools are a liability for the computing environment as well as your business as a whole.

Event & Performance Management

Native to the Windows operating system are two tools commonly used for collecting, visualizing, and alerting based on event and performance-based conditions. The first, *Event Viewer*, provides a unified interface for many of the events that occur on an individual Windows system. Event Viewer's interface allows for the viewing of individual events based on category. It includes limited tools for creating reports as well as sorting and filtering based on event characteristics. With Window Vista and Windows Server 2008, the tool gains new capabilities, including savable views and a limited ability to forward specific event data from one system to another.

Even with its new capabilities, Event Viewer is still a tool designed with the individual system in mind. Its rollup reports, custom views, and enhanced sorting and filtering capabilities work exclusively with event data stored on-system. Notification alerts through one of three interfaces - email, pop-up window, and the invocation of an external program - can be configured for individual events. However, the configuration of these notification parameters is specific to the event as well as the individual system. Event-based triggers can be challenging to configure, and some types of notification alerts require coding and/or scripting skills to be fully realized. Scaling the use of Event Viewer across more than a small number of machines involves added administrative overhead to manage the independent system settings.

The tool used for collecting and viewing performance data in Microsoft Windows is *Performance Monitor*, also called *PerfMon* or *System Monitor*. PerfMon includes similar capabilities for performance visualization and reporting as with Event Viewer. Similar to Event Viewer, PerfMon can be configured for administrator alerting based on specific conditions. A single instance of PerfMon can monitor performance counters for multiple machines; however the total number of machines that can be managed by PerfMon is

limited. PerfMon data is stored in a file-based format that adds administrative overhead for their management and storage.

These two native tools suffer from a singular problem. In order to monitor their event-based data, the administrator is required to manually start the interface for each machine of interest and manually review their log data. No central console that aggregates information across multiple machines is provided, and separate administrative consoles are required for the viewing of event-based versus performance-based data. Due to this challenge, the result with many overworked administrators is that they often neglect performing the critical and proactive task of monitoring their logs.

Third party tools exist that can alleviate many of the limitations of PerfMon and Event Viewer. These tools can centralize event and performance data across multiple machines into a searchable database. The information in that database can be viewed and manipulated via a single, unified interface. Comparing performance data alongside event-based data enables an easier correlation between the triggering of an event and its associated performance loss. Most importantly, when the interface for monitoring systems is unobtrusive to an administrator's daily workflow, they are much more likely to properly make use of that interface.

Troubleshooting

Think for a moment about a typical vehicle. Much of the internal workings of that car or truck are relatively opaque to its driver. Turning the key starts the vehicle and pressing on the accelerator makes it go. A well running engine doesn't require much attention by its driver. But in some cases situations occur where driver awareness is critical. For those situations, that notification is realized through the lights and gauges on the dashboard. These indicators provide the driver with awareness about the vehicle's overall health and status. If the vehicle didn't have these notification components, it would be difficult for the driver to recognize a failure or warning that might otherwise be brought to the driver's attention and potentially avoided.

Compare this situation with the typical Windows computing environment. Servers and workstations work with each other along with devices elsewhere on the network. They combine to provide a set of services that ultimately drives the computing needs of the business. The activities between those servers and workstations occur through a network where unmonitored traffic is opaque to the network's users. When the environment

appears to be operating properly, it doesn't get much attention from its users. But when problems occur, the environment's complexity makes it difficult to locate and resolve the errors. In those situations, much like our vehicle example earlier, the use of monitoring and measurement tools provide the same sense of awareness to its administrators.

When environments lack centralized event and systems management toolsets, the process to locate and resolve errors on the network is made much more difficult. Administrators must regularly view individual event logs on one or more devices. Depending on the complexity of the problem, this activity can take an extended period of time. With hundreds or thousands of event entries per system and multiple systems to manually check for problems, user error and missed events can further worsen the problem.

Furthermore, issues are often not limited to a single server or network device. In situations where more than one server or device in a service thread are suspect, the analysis of event and log information across multiple systems can be challenging with native tools. Inexact time between devices can cause skewed event timing across individual events. Even more challenging is when different device classes utilize different mechanisms for gathering and viewing logs. Aligning events between multiple types of systems can involve multiple, segregated consoles viewed in separate windows.

By centralizing device and server event information from multiple systems and system classes into a single tool, the troubleshooting administrator can view events and logs in a time-aligned interface. This aids significantly in the troubleshooting process. When performance and service or process information is gathered and viewable within the integrated interface, it is possible to align event-based problems with changes in system performance.

Implementing third party tools into the environment can improve the visualization of problem information. The troubleshooting administrator can come to a resolution much faster, having used higher quality information and without resorting to band-aid solutions when a problem's root cause cannot be identified.

Health Reporting & Notification

Thus far the visualization of data within an interface has been addressed. But often times there are needs from administrators, managers, and security officers to be given a

view of data outside the interface. Also necessary is the ability for administrators to be notified through an alert mechanism when they are not using the interface. In either of these situations, the need for external report creation and notification is critical.

Many network elements, whether Microsoft Windows servers, network devices, or individual applications include the ability to configure notification parameters to alert individuals when preconfigured conditions occur. As discussed previously, the native Windows Event Log includes three options for notification: Running an external program, raising an alert window on the console, or sending an email message. In very small environments, these options may be sufficient for notification on specific events. But it is often the case where entire classes of events may be of interest to alert to the administrator. Even more important are the numerous types of devices that make up the computing environment. Syslog messages and SNMP traps from network components will also require some level of alerting. However each of these element types may have a completely different process for setting up and sending the alert. Managing these differences adds a significant administrative overhead.

Many native interfaces are also limited in the types of notification they can support. Email messages are the most often used, as they can be targeted towards administrator inboxes and mobile devices. But other options may be required depending on the needs of the environment. Options like Syslog messaging, paging, sound files, and posting to web pages can extend the standard email-based notification options into other more valuable mediums. As an example, third party tools that enable web posting alerting options can be used to populate web-based heads-up displays with alerting information. They can also be used to automatically generate help desk tickets in environments that use web-based help desk applications.

To the issue of reporting, notification options are excellent for administrators when problems occur. But IT also needs to report on environmental health and status to business leaders and outside auditors. With native tools, this process involves the exporting of information to a text file. This text file must then be manually reconfigured into charts, graphs, and aggregate rollup reports that make sense to executives and auditors. As a manual process, these steps can be difficult and time consuming to reproduce.

Even more critical - as we'll learn in the next section on compliance - is that manual processes are not necessarily guaranteed through a technical control to provide verifiably correct information to the recipient. Using a third party tool's automated process for generating reports guarantees standardized reports that contain verifiably correct information.

As computing environments and IT organizations mature, reporting grows ever more relevant for the fulfillment of Service Level Agreements. It also feeds data into other formal metrics-based agreements between IT and the business or its customers. Third party tools are necessary to formalize and automate the data gathering and formatting process. These tools can leverage standardized reports as well as built-in report generating tools to facilitate and ease the report creation process.

Security Posture & Compliance Fulfillment

Information and systems security is a growing requirement for virtually every connected network. As most business networks connect to the Internet, the threats that can arrive from the Internet mandate extra vigilance on the part of administrators to ensure the security of their networks. Network devices like firewalls and intrusion detection systems are put into place to provide a measure of protection against external attack. But their use is only valuable when administrators are kept aware of the type and magnitude of attacks they are repelling. When a business network is attacked by an external entity, these devices must be able to notify and alert administrators that an attack is underway.

The problem, as discussed earlier, is that each individual device can have its own individual mechanism for event processing, storage, and alerting. For environments that use more than a very few of these devices, a centralized database where security event information can be stored is critically necessary for administrators to have the necessary situational awareness.

Even more critical, though, can be the threat from inside. Both the Microsoft Windows operating system as well as Microsoft's Active Directory retains security logs that document the activities and authorizations completed within the environment. These security logs are part of the Event Viewer discussed earlier in this paper, and they similarly suffer from the same limitations as the Event Viewer's other log types. Depending on the level of auditing turned on for an individual system or a Windows domain, security logs can grow to include

many hundreds of thousands of events over a short period of time. Built-in limitations to the Windows event log makes it cumbersome to store these events and later search through them for specific activity. In addition, with the native setting, events can be overwritten and lost or consume excess resources on the local machine.

In environments where the storage of security events is necessary, third party tools are needed to offload security-based events out of individual system Event Logs. These tools can import security log data into searchable databases as well as archive database formats for long term storage.

Depending on the industry, government or industry-specific compliance regulations may also impose additional requirements on the computing environment for the storage of these security events. Sarbanes-Oxley, HIPAA, PCI DSS, and other compliance regulations typically require the incorporation of a *technical internal control* that facilitates the logging of security-related events. They also often require the secure offloading of security information into storage locations where they can be later used for forensic activities. As an example, regulations within the Sarbanes-Oxley Act require that policies and procedures are incorporated that ensure electronic records and transactions are stored securely and cannot be accessed by unauthorized persons. Security logs within Microsoft Windows can provide for the logging of security related events. However their built-in limitations may be insufficient for successfully passing a security audit.

In these cases, the implementation of a third party tool can offload event information to a centralized database, store it securely, and provide searchable access to select administrators. Aligning these features with our previous section on reporting, the integration of rich reporting within the tool provides the business with an easy-to-create mechanism for providing reports to outside auditors upon request.

TNT Software's Solutions Augment Native Controls with Critically Necessary Capabilities

Throughout the past few pages, this paper has illustrated the challenge of relying on native controls alone for the management and monitoring of distributed systems. As discussed, these issues involve all the devices and applications that make up the business network and are not strictly limited to just the Microsoft Windows operating system. TNT Software's ELM Enterprise Manager provides a comprehensive

solution that integrates with Microsoft Windows, Syslog, and SNMP-capable devices, as well as applications with log files to integrate status and performance information into a single, centralized database.

The ELM feature set extends beyond monitoring, alerting and reporting and provides resilient data management. Local caching and an efficient database failover mechanism support the reliability of the event data. To reduce storage costs and accelerate report generation, custom data pruning and archiving offload the Primary database. It is critical to have safeguards against data loss and storage procedures that differentiate the data for short term analytic and long term forensic objectives.

In addition, ELM integrates the management and monitoring information from each device that makes up the business network to provide a unified console for administrators. Windows systems can be configured with an agent that handles event and data collection and transfer to the central ELM Server. Alternatively, under most conditions, systems can be monitored without an installed agent. Administrators can monitor all device events and performance metrics from the ELM Console. They can watch for errors and setup alerting and notification using a large set of alerting options that go beyond the standard email message notification.

Using ELM Enterprise Manager, administrators can review logs across multiple devices within the interface to assist with troubleshooting. They can trend performance over an extended period of time to see where performance issues relate to event-based errors or hardware bottlenecks. They can also configure levels of access to this data that supports the security requirements associated with regulatory compliance. Throughout all of this, the implementation of ELM Enterprise Manager enhances the security posture of the computing environment by consolidating all security-related information to a single, secure, searchable location.

All in all, centralization of event, health and status data from enterprise systems will support proactive system management objectives, enhance productivity and fortify security policies. It's a winning technology for any IT Manager.

For more information about ELM Enterprise Manager or to download a 30-day full feature trial copy visit TNT Software's website at www.tntsoftware.com.



2001 Main Street
Vancouver, WA 98660
360-546-0878 phone
360-546-5017 fax