

TNT SOFTWARE

White Paper Series

Event Log Monitor™ White Paper:
Architecture

TNT SOFTWARE

Event Log MonitorTM Architecture

© 2000 TNT Software – All Rights Reserved
1308 NE 134th Street • Suite F
Vancouver, WA 98685 USA
Phone 360.546.0878 • Fax 360.546.5017
<http://www.tntsoftware.com>
info@tntsoftware.com
support@tntsoftware.com

Table of Contents

About TNT Software.....	0
Abstract.....	1
Introduction	3
Summary	12

About TNT Software

TNT Software is a Microsoft ISV that develops solutions for Microsoft's Windows operating systems. Our products help automate and simplify the administration of Windows 2000, Windows NT, Windows 95/98 systems and TCP/IP devices and services.

We specialize in building administration tools for Microsoft's Windows operating systems. Drawing from years of experience, we have a unique understanding of the importance of having solid tools available to support the administration of today's complex networks.

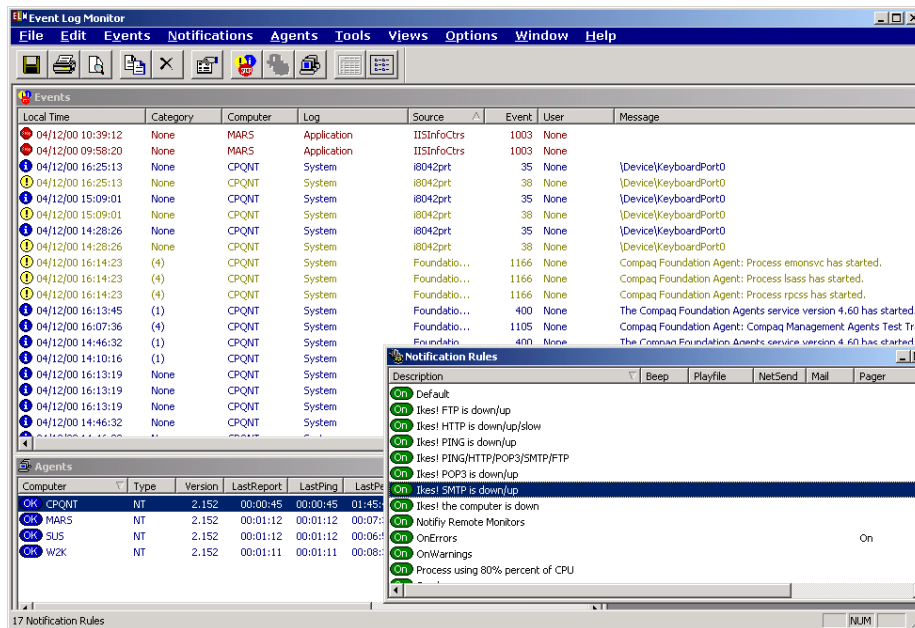
TNT Software is a private company located in Vancouver, Washington. Our clients include companies of all types, from manufacturing to universities, from news agencies to communications companies, and from aerospace companies to government agencies.

If you would like more information about TNT Software or any of our products, please visit our web site at <http://www.tntsoftware.com>, or send email to <mailto:info@tntsoftware.com>.

Abstract

Executive Overview

Event Log Monitor™ provides automated event log monitoring, service monitoring, notification, configuration data collection, performance data collection, and remote administration tools. *Agents* monitor Windows NT and Windows 2000 event logs, system services, and active processes and forward information to one or more central *Consoles*.



The *Console* displays a consolidated view of all the event logs and provides the capability to create custom views of groups of events. Each view is dynamically updated as new events occur in the network. For example, an Internet service administrator can view Internet service related events in one window and similarly, a security administrator may view audit entries in a separate window.

The *Console* monitors common Internet services such as HTTP, FTP, POP3, SMTP, and PING, providing the ability to monitor any TCP/IP based device. In addition, the *Console* can be configured to *listen* for SNMP traps sent from SNMP managed devices, or configured to forward event log entries to a customer owned SNMP management system by generating SNMP traps.

The *Console* has a sophisticated and extensible *Event Filter* feature used to select which events are displayed in each view. *Event Filters* can be applied to *Notification Rules* which associate user defined *Notification Methods* with *Event Filters*. This provides the ability for an administrator to be contacted when important events occur. Using *Event Filters*, the administrator defines which events are important without having to define each individual event. Event filters are created using wildcards and Boolean logic against any information in the event.

The *Console* can store the collected events and performance data in Microsoft® Access, Microsoft SQL Server, or Oracle database. The database store engine automatically manages the databases, creating tables and indexes as necessary, and optionally aggregates performance data to periodically roll-up detailed performance data records into summary records.

Introduction

Introducing Event Log Monitor™

In a network of multiple servers, it is impractical to manually monitor each server's event log. Event Log Monitor automates and simplifies the task of monitoring multiple server event logs. Event Log Monitor is a 32-bit multi-threaded application compatible with Windows 2000, Windows NT 4.0 and Windows NT 3.51. Network communication between the agent and console uses TCP/IP.

Notification methods include beeps, multimedia sound files, network pop-up messages, SMTP/MAPI mail, posting information to web forms, alpha-numeric/numeric pagers, SNMP traps, Short Message System, and user written batch files or programs.

The agent is ~500KB executable with a small memory requirement (less than 4MB) that typically operates using less than 1% of the CPU. Configuration settings are stored in the HKEY_LOCAL_MACHINE\Software registry hive.

HOW EVENT LOGS WORK

Microsoft Windows NT and Windows 2000 event logs are designed for consistency and efficiency. The event logs contain the most important information for diagnosing application and operating system failures, determining the health and status of a system, and verifying that system and applications are operating properly. Message definitions are stored in dynamic link libraries, which are registered with the Event Log service through the registry. Only the event parameters are stored in the event log. This reduces the redundant message text associated with messages. The event log WIN32 API provides the application an interface to store event parameters in the system event logs. An event message and its' parameters are displayed by looking up the appropriate message in the application's message DLL and formatting the message definition with the event parameters.

There are three basic event logs, the Application log, System log, and Security log. There are five types of event log entries, Informational, Warning, Failure, Audit Success, and Audit Failure. In most cases, Audit Success and Audit Failure events are reserved for the Security log, while Informational, Warning, and Failure events are common to the Application and System logs although Audit entries can be entered into any log.

HOW EVENT LOG MONITOR WORKS

Event Log Monitor has two main modules, the Console and the Agent. Agents are configured, installed, and uninstalled directly from the Console, although if need be, they can be installed manually at the Agent. For example, if you are installing Event Log Monitor in a firewall environment that, for security reasons, has some ports blocked, you can manually install an Event Log Monitor Agent on the un-secure side of the firewall and safely send the collected data to the Console on the secure side of the firewall.

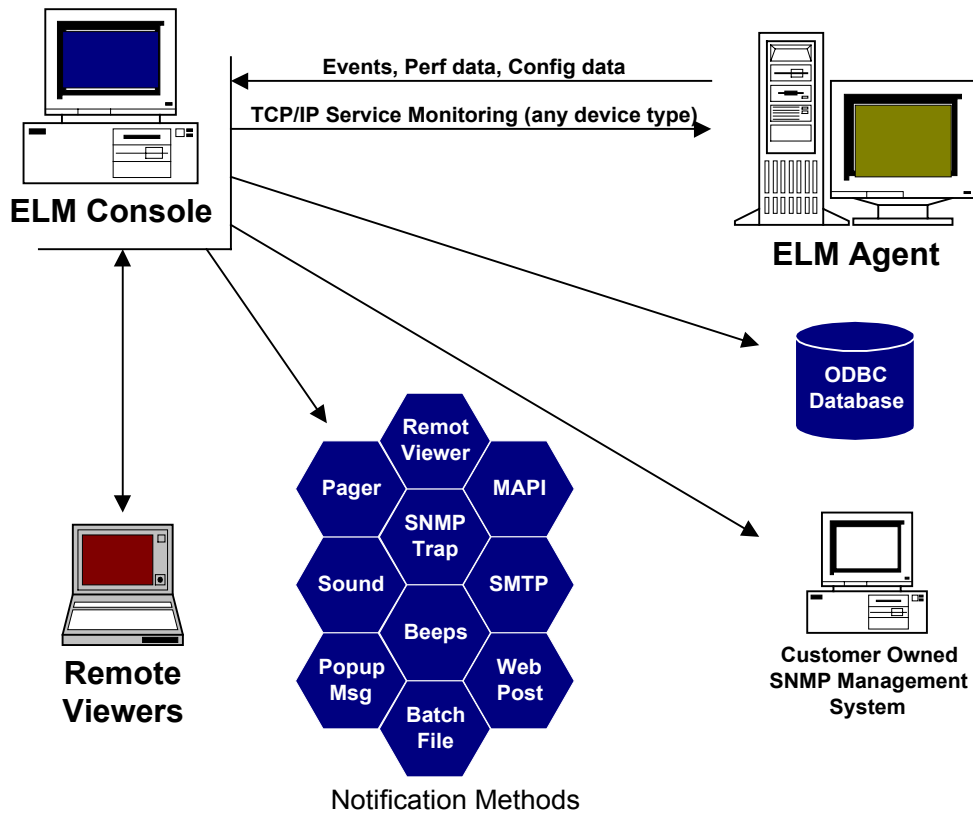
When new events are put into a system event log, the Agent is notified by the operating system through the WIN32 API. The Agent waits three seconds, and then reads all new events, opening the appropriate DLL and formatting the message text. The Agent then sends these new events to the Console.

The Console can configure the agent to ignore certain type of events. This can reduce the network overhead. For example, if the Agent is on a busy domain controller the administrator may choose not to send Audit Success messages over the network.

A separate thread in the agent monitors Windows NT/2000 services and reports changes in the state of the services. When a service or driver stops, an error event log message is generated and the event log monitor thread processes the message. When a previously stopped service or driver starts, an informational event is created and again the event log monitor thread processes the message.

Another thread in the agent monitors active processes. If any process utilizes more than 30% of the CPU during the interval, a Warning event is generated. If the process uses more than 50% of the CPU, an Error event is generated. And if the process appears to be leaking resources such as handles or memory, an error event is generated. The process monitor thread identifies the process in the event log message.

CONSOLIDATED EVENT LOGS

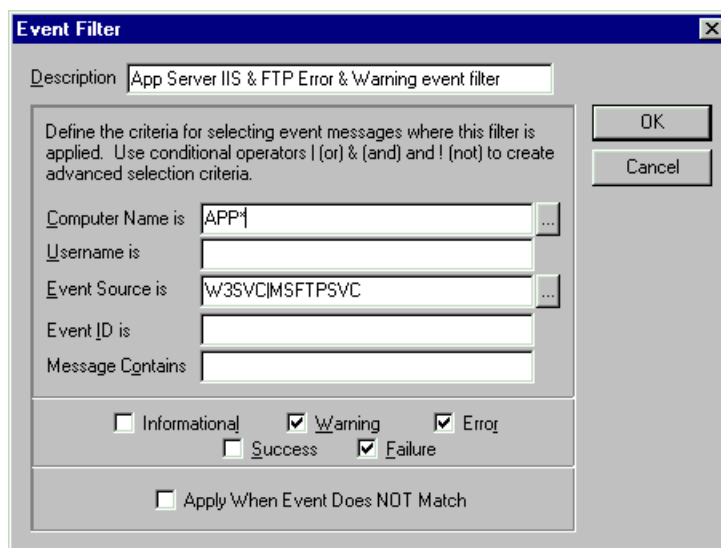


The Event Log Monitor Console displays a consolidated view of the events collected from the monitored systems. This consolidated view can be sorted, searched, and printed, and provides a powerful single source of health, status, and diagnostic information for all the monitored servers. And because it is consolidated, you see events generated because of domain relationships or application interdependencies. A good example of this is the relationship between a domain controller and the applications installed on member servers. If the application uses integrated security, it fails when the domain controller is down. Users may report a problem with the application when, in fact, the problem is with the domain controller. Administrators can waste time diagnosing the application server when, in fact, the problem is on another server. The consolidated view of the event logs makes this and similar scenarios very easy to detect.

The event logs on one busy server can contain hundreds or thousands of events making it difficult to diagnose problems. To reduce the amount of “noise” events such as common Informational or Success events, the administrator has two choices. Your first choice is to configure the Agent to not forward Informational or Success events. This is an all or nothing setting, which is not practical when there is a subset of Informational or Success messages you need to see. So the second choice is to create event filters that exclude the events you do not want to see.

EVENT FILTERING

Event Filters provide a mechanism for selecting a subset of all events. Using wild cards and Boolean logic, the filter will identify with an event or group of events. For example, to deal with messages from SQL Server you can specify SQL* as the event source to select any source that begins with SQL. And to deal with the SQL messages from servers A, B, and C you would specify A|B|C in the computer name field. Leading and trailing wildcards (*) and character position wildcards (%) are supported, as are the Boolean operators Or (|) and And (&).



Any number of Event Filters can be combined to create a complex set of events. By using wildcards and boolean operators, the Administrator does not have to be familiar with every event log message.

Any number of Event Filters can be applied to *Views* in order to display in one window a specific group of events.

Any number of Event Filters can be applied to *Notification Rules* so the Administrator or designate can be notified when an event occurs.

NOTIFICATION METHODS

Notification methods are defined each of the ways you would want to be notified. You may have separate methods for various event categories, or separate methods for various application events. For example, you could have one method that describes how to notify a database administrator about important database related events, and another method for notifying a security administrator about important security related events.

Notification methods use “pass the full event” information to the notification engine, which in turn forwards that information depending on the methods selected. The following describes each of the available methods of notification:

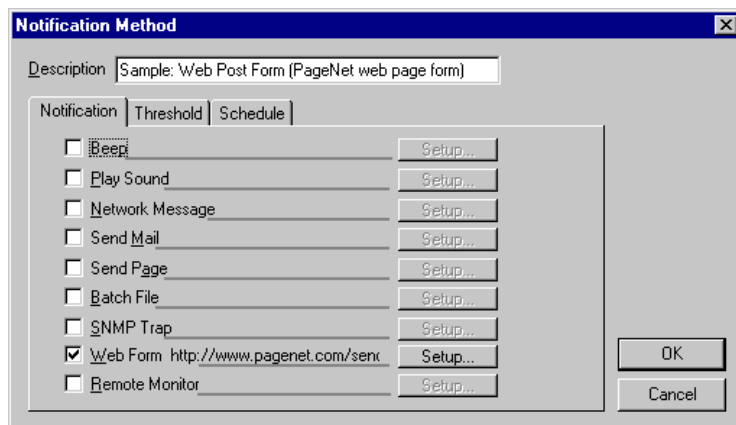
Beep sounds the computer speaker. The frequency and duration are customizable allowing you to designate different sounds to different types of events.

Play Sound plays a WAV file. You can record custom sounds or voices to be played for different types of events.

Network Message sends a pop-up message to another computer. The message can be customized to display any portion of the event details.

Send Mail sends e-mail using either SMTP or MAPI. The message can be customized to send any portion of the event details or user defined text.

Send Page sends pager messages to numeric or alphanumeric pagers, including pagers, which support the Short Message Service. Sending pages requires a modem attached to the Console computer. The message can be customized to display any portion of the event details.



Batch File executes a user defined batch program or executable. The event details are passed to the batch file in the environment space. The user written program can retrieve the event details from the environment variables and process the event accordingly.

SNMP Trap generates an SNMP trap. The Event Log Monitor MIB contains information about the event log message. SNMP trap requires the SNMP service be installed and configured on the Console. Normally the SNMP notification method trap is used in conjunction with a 3rd party SNMP management system.

Web Form posts event information to a user selected web page. This method is useful, for example, to send pager messages through the pager service's web interface.

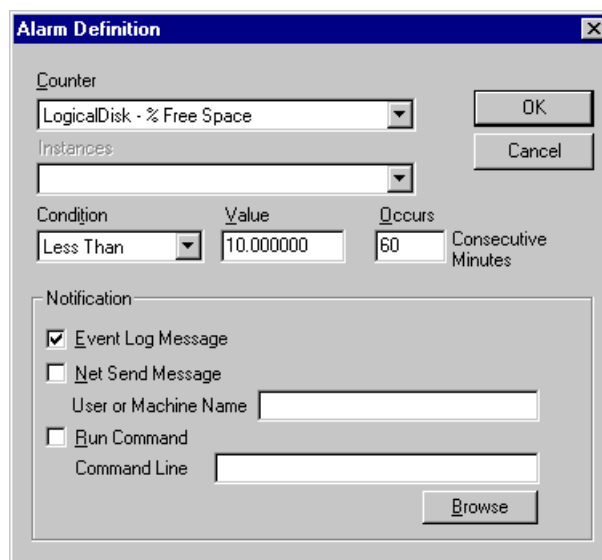
Remote Monitor notifies an Event Log Monitor Remote Viewer about the event. Event Log Monitor Remote Viewer runs on Windows 2000, Windows NT, Windows 95, Windows 98, Windows ME, and Windows CE.

NT PERFORMANCE DATA COLLECTION & ALARMS

Windows NT and Windows 2000 include extensive performance monitoring capabilities. These operating systems publish an extensive list of performance counters, which can be extended by add-on applications.

Event Log Monitor provides a simple to use mechanism for collecting performance data and storing the data in centralized databases. From the Event Log Monitor console an administrator can easily collect any combination of performance counters from any number of remote systems. The built-in scheduling feature permits data collection to occur during specified periods of the day and the built-in data aggregation settings provide a mechanism to automatically roll-up aged data into summary records. Data aggregation reduces the volume of data while maintaining valuable historical data.

Alarms can be defined to monitor for specific conditions, such as low disk space or high CPU utilization. A user-defined action can be taken when an alarm condition occurs.



SNMP & TCP/IP SUPPORT

Event Log Monitor can be both an SNMP client, generating SNMP traps as a notification method, and a SNMP trap receiver generating events that originate as SNMP traps on an SNMP managed device. This provides the ability to monitor any SNMP device such as hubs, routers, printers, etc. The events received as SNMP traps can be applied to notification rules and processed by the notification engine.

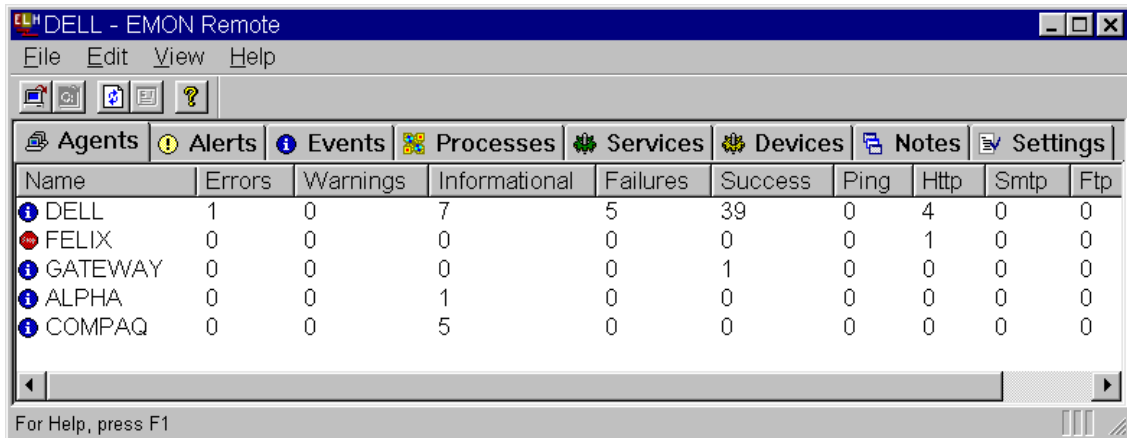
The Console can monitor TCP/IP services running on non-Windows NT computers such as Unix servers, FTP, POP3, SMTP, and ping other devices such as hubs, routers, and printers. The events received when these services fail can be applied to notification rules and processed by the notification engine.

REMOTE VIEWERS

Remote Viewers included with the Event Log Monitor can be used on Windows CE, Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, or any Web browser. The Remote

EVENT LOG MONITOR™ ARCHITECTURE

Viewers communicate with the Event Log Monitor Console and provide the administrator with an important diagnostic and information tool.



The screenshot shows a window titled "DELL - EMON Remote" with a menu bar (File, Edit, View, Help) and a toolbar. Below the toolbar is a tabbed interface with tabs for Agents, Alerts, Events, Processes, Services, Devices, Notes, and Settings. The "Agents" tab is active, displaying a table with the following data:

Name	Errors	Warnings	Informational	Failures	Success	Ping	Http	Smtp	Ftp
DELL	1	0	7	5	39	0	4	0	0
FELIX	0	0	0	0	0	0	1	0	0
GATEWAY	0	0	0	0	1	0	0	0	0
ALPHA	0	0	1	0	0	0	0	0	0
COMPAQ	0	0	5	0	0	0	0	0	0

At the bottom of the window, it says "For Help, press F1".

MESSAGE DEFINITIONS & USER DEFINED NOTES

Event Log Monitor has a unique feature that allows the organization to store notes and information related to event log entries. The information is stored in the centralized Events database and can be updated, searched, and displayed using the Event Log Monitor console or Remote Viewer software. This provides the capability for an organization to track important events and to keep information related to system administration in a common database.

SCALABILITY & LIMITATIONS

A Console can monitor any number of servers, workstations, or TCP/IP devices. The network architecture and domain model in use will be the limiting factor in deployment planning. While the typical event log message is 150 bytes or less, enabling file and process auditing on a monitored system can greatly increase the number of events generated, thereby increasing the network bandwidth demand and processing requirements of the console. Limiting the types of events forwarded to the console from the agent provides a level of manageability. The user of the console must have administrative rights on the monitored system to install the agent remotely, otherwise normal user rights will be sufficient to monitor a remote system.

A monitored system can report to any number of consoles. There can be, for example, consoles dedicated to monitoring specific applications such as a Security Administrator, Database Administrator, and Web Master might use. Each administrator may monitor a set of servers, workstations, or TCP/IP

EVENT LOG MONITOR™ ARCHITECTURE

devices regardless of whether or not the machine is being monitored by another console. Event Log Monitor supports a distributed monitoring environment for both centralized and distributed administration organizations. While deployment may be centralized with a single console monitoring all servers (one-to-many), Event Log Monitor can also be deployed in a decentralized fashion with several consoles monitoring several servers (any-to-any).

FIREWALL ENVIRONMENTS

The console and agents use TCP/IP to communicate. The TCP/IP port settings can be assigned specific to the firewall environment.

AUTOMATED TASKS

The console can automatically update agents when new releases of the software are deployed. There is no need for hands-on work on the monitored systems. The Event Log Monitor can take corrective action when specific events occur and can be used to automate recovery of service failures and runaway processes.

APPLIED APPLICATIONS

Event Log Monitor can be implemented for various purposes with intertwined goals. An individual's responsibilities would determine the level of notification assigned to event log messages. Messages deemed very important would typically require immediate notification via pager or e-mail, while messages with less importance would have a more subtle notification method such as a unique beep or sound file.

SECURITY MONITORING

A Security Administrator is primarily concerned with who is accessing what resources, and identifying break-in attempts. Event notification rules would log all security events to a file or database, while specific messages regarding sensitive data may warrant immediate notification. The Security Administrator is able to see repeated logon failures, as well as changes to the auditing configuration and accounts database, across multiple servers, which helps to detect break-in attempts and security breaches.

SECURITY AUDITING

Where security monitoring is primarily concerned with intrusion detection, security auditing is concerned with keeping track of who has done what, where, successful or not. By configuring Windows NT security to record audit information, the consolidated audit event log messages are available in the Event Log Monitor events database for generating system wide audit reports.

NETWORK ADMINISTRATION

The Network Administrator is primarily concerned that the network and its' components are functioning properly. Event notification rules would primarily filter for service and network oriented messages.

BACKUP MONITORING

The Backup Operator is primarily concerned with status of backups. Event notification rules would filter for backup related messages.

APPLICATION SUPPORT

An application support specialist is primarily concerned with the health and status of an application. In-house applications can use the event log to communicate status information back to the support specialist. Event notification rules would filter for application specific messages, with secondary consideration for service and network oriented messages.

HELP DESK

Help desk support is primarily concerned with health and status and providing a first level of support to end-users. A down system could result in a flood of calls to the help desk. The consolidated view of event logs provides a quick reference and potential for limited down time of a server.

Summary

Event Log Monitor allows you to be as proactive as possible

The first step in resolving every problem is being aware that there is a problem. The faster you become aware of the problem, the sooner it can be resolved. Event Log Monitor provides the first notification and diagnostic resource for understanding the scope of a problem. Many big problems resulting in system down time begin as small issues that go undetected. Event Log Monitor helps you react quickly when problems occur.

- The consolidated view of event log messages is essential to see the whole picture when diagnosing and resolving problems. By using Views to organize volumes of event log information the administrator can quickly diagnose problems.
- The notification engine and event filters are simple, intuitive, and provide the power and flexibility to enhance the overall availability and stability of network services. Notification scheduling provides the ability to direct notifications based on the time of day. Notification thresholds provide the ability to counteract event storms.
- SNMP integration and TCP/IP service monitoring provide integration for cross platform Unix environments.
- Performance data collection makes it simple to collect NT performance data in a centralized database. Collecting performance counters is essential to proactive capacity planning and system tuning activities.
- An extensible centralized database for event message reference, notes, and comments with the tools to search, display, edit, and copy the information provides a valuable resource for network administrators.
- Any-to-any configuration of consoles and agents provides a rich monitoring environment for business critical servers and applications.

EVENT LOG MONITOR™ ARCHITECTURE

- Remote Viewers that run on Microsoft platforms such as Windows CE, Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, and Web browsers provide easy access to the information administrators need when problems occur.
- Event Log Monitor has proven scalability, performance, and application in large Windows NT/2000 networks.