OmniMonitor Essentials

Omnitech InfoSolutions Ltd
Omnitech House
A-13, Cross Road No. 5,
Marol MIDC, Anderi (E),
Mumbai 400 093

White Paper

Infrastructure Monitoring Through OmniMonitor

Table of Contents

Annexure – Monitors

Monitors

White Paper on OmniMonitor

Introduction

Infrastructure Availability and Performance Monitoring

OmniMonitor is an agentless monitoring solution designed to ensure the availability and performance of distributed IT infrastructures — e.g., servers, operating systems, network devices, network services, applications, and application components. This proactive, Web-based infrastructure monitoring appliance is lightweight, highly customizable, and doesn't require high overhead agents on your production systems. OmniMonitor is powered by Windows 2003 operating system and Mercury Sitescope. With OmniMonitor, you gain the real-time information you need to verify infrastructure operations, stay apprised of problems, and solve bottlenecks before they become critical.

Agentless Architecture Reduces Total Cost of Ownership

Unlike agent-based monitoring approaches — which require higher overhead, maintenance, and labor costs — OmniMonitor reduces total cost of ownership by:

- Gathering detailed performance data for infrastructure components
- Eliminating the need for extra memory or CPU power on production systems to run an agent
- Reducing the time and cost of maintenance by consolidating all maintenance to one central server
- Removing any requirement to take a production system offline in order to update its agent
- Eliminating the wasted time of tuning monitoring agents to coexist with other agents
- Reducing installation time by removing the need to physically visit production servers or wait for software distribution operations
- Abolishing any possibility that an unstable agent can cause system downtime and subsequent loss of business.

Enterprise Service Level Management

Managing service levels from a business perspective has become a critical requirement for most companies. With this requirement comes the challenge of translating the "bits and bytes" of infrastructure performance data into meaningful business impact metrics. OmniMonitor helps you make the move to service level management (SLM) by:

- Providing built-in best practices that specify the most important metrics to monitor
- Capturing an accurate and timely picture of infrastructure performance
- Integrating with Service Level Management to enable real-time views and reports that correlate infrastructure performance with business processes.
- By starting with OmniMonitor and adding Service Level Management, you can be assured of solid infrastructure monitoring that enables you to manage service levels from a business point of view.

How It Works

Agentless Architecture

OmniMonitor's "agentless" architecture gives administrators a centralized view of systems and Web monitoring without the need to install agents or software on production systems. OmniMonitor is implemented as a Java server application. It runs on the server as a daemon process, monitors system parameters, sends alerts, and generates summary reports. A user connects to OmniMonitor using a Web browser to view status information and make configuration changes. The architecture consists of a set of objects that perform various functions:

- **Webpage Objects** handle requests for Web pages from a browser (using the HTTP protocol). These objects display the current status information, present forms for editing the configuration and update the configuration based on form requests. These objects also enforce any access controls, such as username and password that restrict who is allowed to access the OmniMonitor Web pages.
- **Scheduler Objects** coordinate when monitors are run, alerts are created, and reports are generated by means of the configuration file.
- **Monitor Objects** collect information about the system being monitored. There are objects for monitoring application logs, CPU usage, disk space, processes, Web server throughput, DNS servers, mail servers, access to Web pages, and network response. The Monitor API allows custom monitor objects to be added to handle application specific monitoring needs.
- **Alert Objects** send alerts about exceptional events. There are objects for sending email, pager, and SNMP trap messages. The Script Alert Object allows application specific scripts to be run.
- **Report Objects** generate reports summarizing monitoring activity. The objects read the history information from the log files, summarize, filter, and generate HTML reports in graph and table format.

Remote Monitoring

OmniMonitor allows system administrators to remotely monitor multiple servers from a central installation without the need for agents on the remotely monitored machines. OmniMonitor accomplishes remote monitoring by logging into systems as a user from its central server which can run on Windows, UNIX and Linux platforms. Several login formats are supported including TELNET, RLOGIN, HTTP, SSH and NETBIOS

OmniMonitor Specifications	
Dimension	(W x D x H) , 146 x 254 x 72 mm
Weight	2.7 Kg., w/o power supply
External Ports	Ethernet Port (10/100 Mbps)
	Two USB (Version 2) Ports
	25 Pin (D-Type) Printer Port
	9 Pin (D-Type , Female) Serial Port
	External Monitor, Keyboard & Mouse Port
External Power Supply	230V ~ Input , 12V DC O/P
Processor	Intel Pentium IV with onboard cache
Memory	512 MB expandable upto 2 GB
Disk	40 GB IDE
Operating System	Windows 2003 Server, Service Pack 2 OR RedHat Linux 8.0, RedHat Linux 9.0 with Native POSIX Threading Library (NPTL) and RedHat Enterprise Linux 3.0
Sitescope	Mercury Site Scope 8.0 . (min 100 monitor).
Optional	Rack Kit, SMS Mobile , Monitor , Keyboard, Muse

The above system requirements are based on a standard mixture of monitor types. Some monitors are more resource intensive than others. In addition, the OmniMonitor server should be tuned, especially for higher total monitor and monitors per minute scenarios.

OmniMonitor Server Monitoring Services

Infrastructure	Services	Parameters	Activities
Server Monitoring	Capacity Management	CPU Utilisation	CPU Utilization Monitor reports the percentage of CPU time that is currently being used on the server
		Memory Utilisation	Memory Monitor provides an easy way for you to track how much virtual memory is currently in use on your server.
		Disk Space Management	Disk Space Monitor provides an easy way for you to track how much disk space is currently in use on your server.
	Availability	Ping Response	Ping Monitor checks the availability of a host via the network. Use this monitor to ensure that your connection to the Internet is alive and well
	Mail Services monitoring	Mail Traffic Monitoring	Mail Monitor checks a Mail Server via the network. It verifies that the mail server is accepting requests, and also verifies that a message can be sent and retrieved.
		MAPI Services Monitoring for MS Exchange	MAPI Monitor checks a Messaging Application Program Interface (MAPI) server to confirm that e-mail operations can be executed.
	Citrix Server Monitoring	Availability and Performance Management	The Citrix Server Monitor makes use of Performance Counters to measure application server performance. Citrix Server Monitor allows you to monitor the availability of an Citrix MetaFrame servers
	SAP Server Monitoring	Availability and Performance Management	SAP Monitor allows you to monitor the availability and performance statistics of a SAP Application Server

Server Monitoring Continued

Server Monitoring	Webserver Monitoring	FTP Access Monitoring	FTP Monitor attempts to log into an FTP server and retrieve a specified file. A successful file retrieval assures you that your FTP server is functioning properly
		Web Server Availability (Link Check)	This monitor can be set to check every link on your site, internal and external, every day, letting you know immediately which links have a problem.
		Web Server Availability (URL Check)	URL Monitors provide you with end-to-end verification that your Web server is running, serving pages correctly, and doing so in a timely manner.
		URL Content Monitoring	URL Content Monitor is a specialized variation of the URL Monitor that can match up to ten different values from the content of a specified URL
		URL List Monitor	URL List Monitor to check a list of URLs without having to create a separate URL monitor for each one. For example, this is useful if you host several Web sites and simply want to ensure that they are each serving pages properly
		Load Statistics- No of Hits / Min.	Web Server Monitor runs, it writes the current hits per minute and bytes per minute
		Performance Monitoring	The Web Service Monitor is used to check Simple Object Access Protocol (SOAP) enabled Web services for availability and stability.
	Windows Server Operating System Monitoring	Windows Performance Monitoring	The Windows Performance Counter Monitor tracks the values of any Window NT performance statistic
		Windows Resource Management	Windows Resources Monitor allows you to monitor system performance
		Proactive Fault Management	The Log File Monitor watches for specific entries added to a log file by looking for entries containing a text phrase indicating the fault
		Lightweight Directory Monitoring	LDAP Monitor verifies that a Lightweight Directory Access Protocol (LDAP) server is working correctly

Network Monitoring through OmniMonitor			
Infrastructure	**Services**	**Parameters**	**Activities**
Network Monitoring	Performance Monitoring	Network Statistics	Network Monitor provides an easy way for you to track network statistics for server. Information provided by this monitor can help you track down performance problems related to the network
	Capacity Management	Bandwidth Management	Network Bandwidth Monitor to monitor SNMP-enabled network appliances such as routers and switches.
	Availability	Port Availability and Performance Monitoring	Port Monitor verifies that a connection can be made to a network port and measures the length of time it takes to make the connection
	SNMP Support	SNMP framework based Reporting	Connecting the network information to SNMP devices for the network reporting through different SNMP framework tools
			SNMP Trap Monitor watches for SNMP Traps received to report network problems
			SNMP by MIB Monitor allows you to monitor objects on any SNMP agent

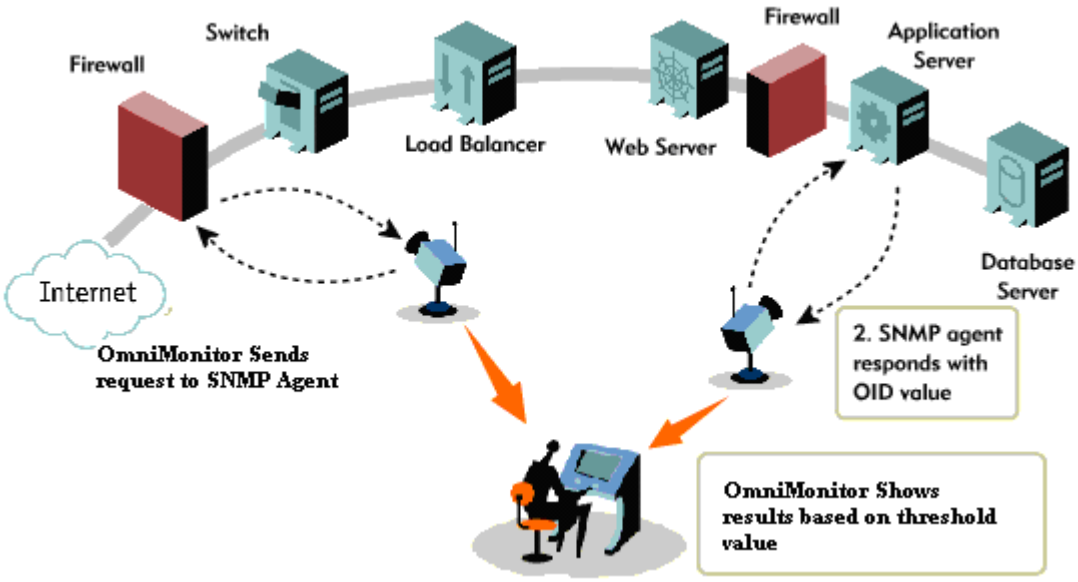
Database Monitoring through OmniMonitor			
Infrastructure	**Services**	**Parameters**	**Activities**
Database Monitor	Availability	Database Availability Monitor	Oracle Database Monitor allows you to monitor the availability of an Oracle database server (versions 8i and 9i plus some earlier versions).
	Oracle Server	Availability and Performance	Oracle9i Application Server Monitor allows you to monitor the availability and performance statistics of an Oracle9i Application Server.
	SQL Server	Availability and Performance	SQL Server Monitor allows you to monitor the availability and performance of an Microsoft SQL Server (versions 6.5, 7.1, 2000) on Windows NT systems

Security Monitoring through OmniMonitor			
Infrastructure	**Services**	**Parameters**	**Activities**
Security Monitoring	Checkpoint Management	Performance Monitoring	Check Point Firewall-1 Monitor allows you to monitor the statistics of a Check Point Firewall-1 using SNMP
	Content Monitoring	URL Content Management	URL Content Monitor is a specialized variation of the URL Monitor that can match up to ten different values from the content of a specified URL

SNMP Monitor

The increasingly diverse technologies used in today's network environments present a challenge to integrating and consolidating management solutions. The range of protocols, applications, and file formats it supports makes OmniMonitor the ideal choice for monitoring the availability of increasingly complex system environments. The versatility of OmniMonitor was enhanced further with the addition of the SNMP Trap Monitor (version 6.0) and Web Service Monitor (version 5.6). These not only position OmniMonitor well for the current trend toward Web services but further enhance its capabilities to monitor systems that employ the simple network management protocol (SNMP) such as firewall appliances, load balancers, and some Web application servers.

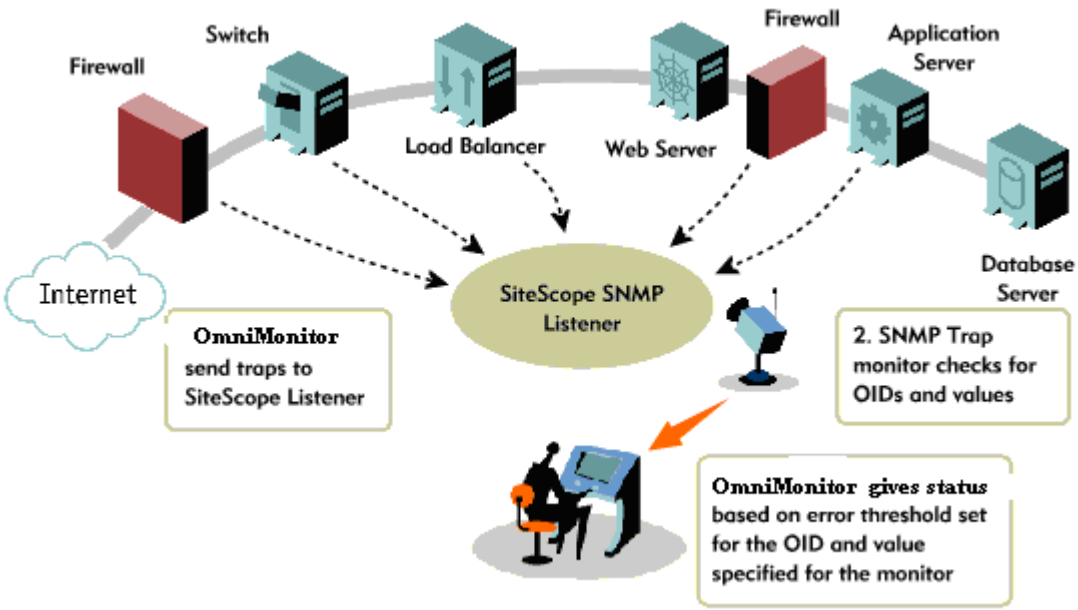
OmniMonitor's SNMP capabilities offer system administrators an alternative to other enterprise management systems or a means of forwarding data from non-SNMP enabled systems to an SNMP-based enterprise management console. System administrators will need to be familiar with SNMP concepts to make the most of these features. This includes having access to the MIB OID tables/data for the SNMP enabled devices on the network.



Monitoring SNMP Traps with OmniMonitor

Applications and solutions for monitoring and managing SNMP enabled network applications can be complicated and costly. Yet monitoring SNMP-enabled systems may be needed when implementing a complete operations monitoring solution. For example, performance parameters for many router and firewall appliances are accessible only through SNMP.

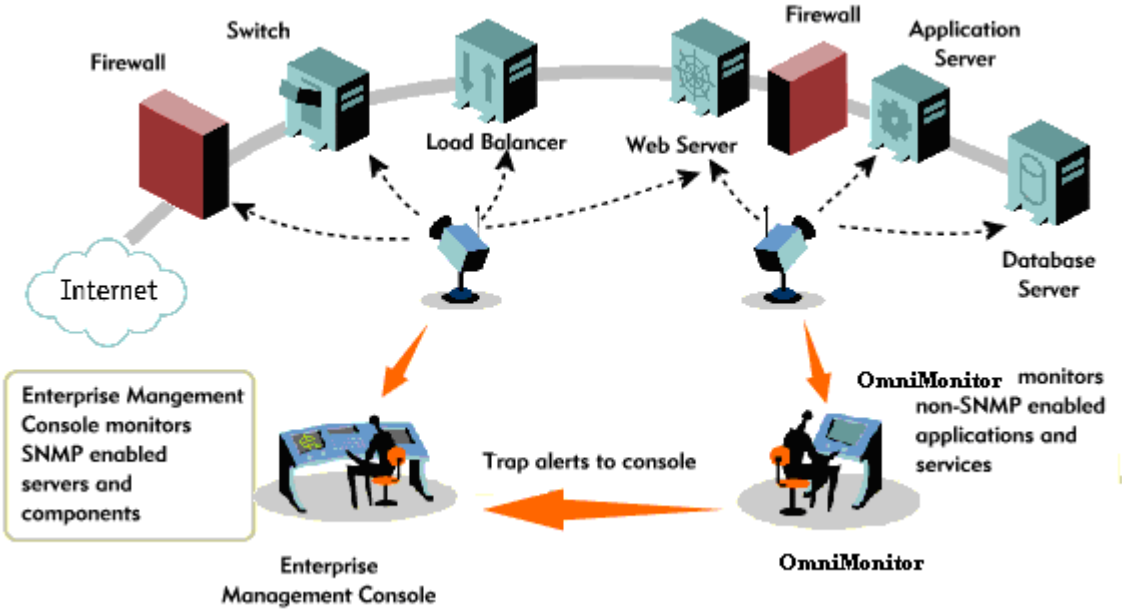
The OmniMonitor SNMP Monitor performs an SNMP get command to a specific SNMP device



Or agent. The agent returns the value associated with the requested OID. This allows a system administrator to use OmniMonitor to watch specific values through the use of a single OID such as processor usage or rejected requests.

The OmniMonitor SNMP Trap Monitor listens for SNMP traps. This lets OmniMonitor act as a SNMP management console.

Using the match content feature a system administrator can search the list of SNMP data returned for a particular OID or content match. It is necessary to configure SNMP agents to send trap to the OmniMonitor server



Sending SNMP Traps with OmniMonitor

Interoperability with SNMP systems works both ways with OmniMonitor. Not only can OmniMonitor act as a management console that listens for and receives SNMP traps, it can also act as an SNMP agent. The OmniMonitor SNMP Trap Alert allows a system administrator to send SNMP traps to an enterprise management console. This is particularly useful because it allows system administrators to use OmniMonitor to forward traps for systems that are not SNMP enabled. Any OmniMonitor monitor, such as a File or URL monitor, can be associated with a SNMP Trap Alert. When the monitor detects an error, OmniMonitor will send a SNMP trap to the management console.

Alerts

An important measure of the return on investment (ROI) for technology is increased productivity. In the area of distributed system management this includes reduced downtime and fewer hours spent managing systems. OmniMonitor provides system administrators with the tools to ensure uptime by monitoring the availability of systems and server performance metrics. Administrators can use OmniMonitor alerts to automate system administration tasks and integrate with other management tools.

Script Alert

OmniMonitor's Script Alert is a versatile feature that allows system administrators to automate routine and special system administration tasks. In cases where temperamental servers go down on a regular basis, system administrators can create a Script Alert to run a batch or error recovery script to restart a service or even reboot the server automatically as soon as a problem is detected. For more routine tasks, such as moving or deleting files, a company can set up OmniMonitor monitors like the Directory and File monitors to watch for file counts, ages or content. Script alerts can run a batch file that copies or moves files. It is also possible to use Script Alerts to launch another application.

Database Alert and Post Alert

Interoperability is the key to a successful IT system. OmniMonitor is designed to be interoperable with other tools and applications. The OmniMonitor Database Alert and OmniMonitor Post Alert are two examples of this interoperability.

If your company is using a trouble ticket system to track problem remediation in your systems, you may be able to use OmniMonitor alerts to automatically create entries in your trouble tracking system. This saves time compared to making entries manually. If your trouble ticket system is based on an SQL database, you can set up Database Alerts to enter records into your tracking database. OmniMonitor's highly customizable alert templates allow you create automated alerts tailored to the fields used in your database records. Alternately, if your trouble tracking system is a Web-enabled application using the common gateway interface (CGI), you can use the Post Alert to have OmniMonitor automatically submit alert entries to the system. Like the Database Alert, the Post Alert template can be customized to include the name-value pairs required by your tracking system.

Alert Types

Along with the Script, Database and Post Alerts, OmniMonitor can use a variety of other media to send alerts when problems are detected. One or more monitors can trigger alerts. They also have flexible trigger thresholds that can filter transient false alarms or create escalation schemes.

The following table lists the all of the alert types available in OmniMonitor.

Alert Types	Description
E-Mail	Sends a problem notification and description as an e-mail message. The message content can be customized to include custom text and specific monitoring results.
Pager	Sends a notification to a pager. Alerts can be sent as alphanumeric pages that include specific monitoring information.
Script	Initiates the execution of a script or other program. Provides the capability to run automated recovery actions based on particular monitoring results. This might include automatically rebooting a service or moving files.
SNMP Trap	Send an SNMP trap to an enterprise management console. Provides interoperability with network management applications.
Sound	Plays an audio file alert on the server where OmniMonitor is running.
Database	Logs problem notifications and descriptions to a database.
Disable/Enable Monitor	Automatically disables or enables an individual monitor or group of monitors. This is useful to suppress redundant alerting from monitors watching elements that are dependent on a single service that may have gone down.
Log Event	Logs problem notifications as events into the Windows NT Event Log.
Post	Sends problem notification and descriptions to another server application as a CGI POST.
SMS	The SMS Alert form allows you to add or edit an Alert that send a message using the Short Message Service. This can be used to send a short message to a SMS-enabled mobile phone or other wireless device. The SMS Alert is designed to transmit only the name of the OmniMonitor monitor that has reported an alert condition and the status of that monitor as the content of the message.

OmniMonitor Network Security Compliances

The e-business transformation is not only changing the competitive landscape; it is changing the very nature of how enterprises view security. Data and transaction security is of paramount importance in this age of rapidly expanding commercial and public computer networks and the emerging Internet economy. For an e-business transformation to be successful, security has to become a top priority in every company that makes use of IT. Not long ago, many security discussions centered on a need to keep information proprietary, i.e., "keeping the bad guys out." This is no longer the case with today's evolving Internet economy. Major aspects of IT security are: confidentiality, integrity, and availability. As the impact of e-business and Internet economy on modern enterprise increases, companies see additional challenges in the way they do business. Companies must ensure that their web and intranet sites support their brand image and ability to deliver products and services seamlessly to customers, suppliers, partners, and employees. An unavailable or slow-performing website can affect a company's reputation, image, and revenues.

OmniMonitor is an agentless monitoring solution designed to ensure the availability and Performance of distributed IT infrastructures, including servers, operating systems, network devices, network services, applications, and application components. This proactive, web-based infrastructure monitoring solution is lightweight, highly customizable, and doesn't require high overhead agents on your production systems.

OmniMonitor Security features are:

• Scheduled secured aggregation of all systems and web-management and monitoring data into a single view.
• Centralized monitoring of large and small web-server farms.
• Complete suite of monitors that watch critical web-environment components.
• A published API for the creation of custom monitors and integration of homegrown tools.

Access Control

Access Control provides information confidentiality. The Access Control concept warrants that information is unavailable to those who are unauthorized to access it. Information access is controlled and granted only to those who need it by a predetermined security policy. When OmniMonitor is first installed on a system, a default administrator is created with a super authorization role. The "super" administrator can create other administrators and delegate any level of authorization and roles desired.

All application and monitoring information is stored in a local OmniMonitor database on an OmniMonitor server. Only administrators with appropriate valid login names/passwords and appropriate permissions can access the data. All data access is done via HTTP/HTTPS protocols, so that OmniMonitor's own web server can grant or deny access. In other words, OmniMonitor has its own ACL software to make sure only authorized administrators have access to the application and its data.

If even more secure access control methods are desired, OmniMonitor can use Netscape, Microsoft IIS, or Apache web servers instead of its own server. This will allow connecting to OmniMonitor using advanced security features of these servers like SSL and Client Certificates.

Authorization and Authentication

Authorization and authentication refers to the concept of granting and validating certain rights for entitled users to perform actions, access systems, data, applications, or networks that would otherwise be disallowed. As discussed above, only OmniMonitor administrators can gain access to OmniMonitor. The basic method of authentication requires a user ID and password. To further restrict access, OmniMonitor lets the user specify a range of IP addresses or IP networks entitled to connect and gain access to the server.

In addition, different administrators can have different authorization roles. Administrative privileges can be delegated with a wide variety of roles. Once a user is authenticated, his privileges and roles are determined according to permissions. Based on that, access rights can be limited starting from read-only access to specific groups, to specific groups in OmniMonitor configuration, with or without editing capabilities to different groups.

Privacy and Integrity

Privacy and integrity ensure that other people on the network cannot see the contents of messages being transmitted. It also means that the information is not modified in unexpected ways. In other words, privacy and integrity policies assure that data, system, or application files have not been tampered with. Privacy and integrity usually go hand in hand. If communications are secure, then no one should be able to alter these concepts. Communications with OmniMonitor and between OmniMonitor servers are done via standards based HTTP or HTTPS protocol encapsulation. This proves to be an effective method since these protocols are widely accepted standards. However, web environments usually expand beyond enterprise boundaries to the Internet and span multiple firewalls. OmniMonitor achieves advanced security for its data communications via encryption by supporting HTTPS protocol. Consequently, all data transfers between OmniMonitor servers or between OmniMonitor and users are encrypted to reduce

Security risks. Furthermore, the data traveling across networks only contains OmniMonitor-related information as opposed to general-purpose data. This monitoring data is stored on an OmniMonitor server, and can't be retrieved by an unauthorized user. The fact that OmniMonitor includes its own web server does not present a security threat. Because this is not a general purpose HTTP server, it can only serve OmniMonitor specific data and only to authorized administrators.

Availability

Availability prevents application and resources from becoming unavailable. As indicated before, OmniMonitor collection and reporting of e-business data is not intrusive on managed systems. In fact, most of the management is agentless (i.e., done remotely). For this reason, management functionality does not interfere with monitored servers and applications. Moreover, OmniMonitor collects most of the web-management information by using application and operating system native protocols and means.

For example, Windows NT Server and OS monitoring is achieved through remote calls to NT Performance Monitor. Various HTTP/HTTPS requests are transmitted to web servers to collect management information and track transactions from start to finish; the same applies to other applications. The table below outlines a few other server and application types, and corresponding protocols, which OmniMonitor employs when monitoring 98:

Web Servers	HTTP/HTTPS protocol
Mail Servers	POP, IMAP protocol
FTP/Telnet Servers	FTP, TCP/IP protocol
DNS Servers	TCP/IP, DNS requests

By using industry standards-based architecture and protocols native to each managed server and application, the risks affecting the resources and performance or availability of these systems are virtually eliminated.

Another benefit of utilizing standards-based architectures and protocols is seen in the use of HTTP protocol encapsulation. HTTP traffic is widely accepted and its flow through corporate firewalls is well controlled by IT organizations. Using TCP/IP port standards proves to be advantageous: IT organizations do not need to open any ports in corporate firewalls in order to allow management and monitoring traffic through. For that reason, opening additional ports does not compromise network security policies in firewalls.

Monitor Locally, Access Globally

There are, however, certain situations with certain monitoring configurations that may present potential security issues. These configurations can be easily avoided. A common situation occurs when remotely monitoring Windows NT servers inside corporate firewalls with an OmniMonitor server that sits outside the firewall. Since NT remote monitoring is accomplished by using the Microsoft Perfmo API (requires TCP/IP over Netbios services), special Netbios ports must be opened in the firewall. Doing so creates a common security problem. Once Netbios traffic is allowed to pass through the firewall, this hole can be exploited further to gain access to information published by Netbios services.

The previous configuration is generally not recommended and can be easily avoided by monitoring NT servers with a local OmniMonitor server installation (i.e., by using two connected OmniMonitor servers inside and outside the firewall). The server inside the firewall can perform NT remote monitoring and make the data available to the server outside the firewall.

Access to both servers is via a browser. The communication between OmniMonitor servers has the following characteristics:

- Communication is secure, via HTTP/HTTPS protocols.
- It uses standard TCP/IP ports that are usually open and well-controlled in firewall.

• There is no need to open Netbios services firewall ports and therefore there is no unnecessary security exposure.

The table below outlines TCP/IP and UDP ports used by OmniMonitor.

Features	Port
OmniMonitor Web server	8888 or any set custom value
URL Monitor80	443
SNMP Monitor	161 (UDP)
FTP Monitor	21
Mail Monitor	25 (SMTP), 110 (POP3), 143 (IMAP)
News Monitor	119
E-mail Alert	25
Post Alert	80,443
SNMP Trap Alert	162 (UDP)
Remote Unix	22 (ssh)
Remote Unix	23 (Telnet)
Remote Unix	513 (rlogin)
Remote NT Monitoring	139

Misconceptions of Infrastructure Monitoring System

Introduction

Today's enterprises depend on the availability and performance of their mission-critical business applications. If these applications suffer from degradations in performance or fail completely, companies are subject to lost revenue and decreased customer satisfaction. In order to avoid these undesirable outcomes, IT departments must adopt effective monitoring strategies without actually making problems worse or breaking the bank in terms of total cost of ownership (TCO).
Since the earliest days of business computing – with monolithic mainframe installations and very small client/server-based systems – IT departments have well understood the importance of monitoring availability and performance and have invested in various forms of monitoring solutions. But as technology has rapidly evolved, some of yesterday's best practices are no longer valid. In some cases, following outdated strategies can lead to ineffective monitoring, high overhead, and increased costs. In addition, today's complex, distributed applications on which many mission-critical business processes rely are forcing companies to revaluate their long-held beliefs about monitoring and to add new best practices to their existing strategies.
The purpose of this white paper is to draw attention to 10 aspects of monitoring that may have once been perfectly sound best practices, but now, due to new technologies and/or changes in IT infrastructures, may no longer apply in most availability and performance monitoring situations.

1. Monitoring basic infrastructure is enough.
Monitoring system metrics (such as CPU, memory, and disk) is important – but these metrics do not provide enough information to truly understand whether actual users or applications are experiencing performance problems. Trying to "add up" individual system performance metrics in order to understand actual application or end-user performance doesn't work either. True end-user-oriented monitoring is critical and should be the starting point of any monitoring strategy. In addition, due to advances in hardware reliability and performance, the causes of most performance problems today are usually problems with application components, as opposed to individual pieces of hardware. Thus, system monitoring alone – while still critical – will not provide an accurate or complete picture of true application performance.

2. The only way to monitor is with an agent.
Infrastructure availability and performance monitoring initially started with deploying heavyweight agents that added significant overhead to the production systems, and therefore introduced the possibility that the agents could either crash the system or become the cause of the very problems they were trying to help avoid. Companies lived with this possibility because few other options existed.
Today there is another way – agentless monitoring. Agentless monitoring enables operations groups to monitor complex, distributed systems without installing agents or software on the production systems. These solutions non-intrusively monitor any and all parts of the IT system remotely from a single host machine. Agentless monitoring solutions ensure rapid deployment; ease application maintenance and upgrades; reduce the risks of impacting production systems; facilitate system

infrastructure and expansion; and increase return on investment (ROI) by decreasing the TCO of the monitoring infrastructure.

Some companies have hesitated to implement agentless monitoring due to several common misconceptions. The most common agentless misconceptions are:

- **Agentless monitoring requires too much bandwidth.**

Agentless solutions connect remotely to the monitored production servers and do create some network traffic. However, in order to monitor most basic system and application metrics for the purposes of identifying sustained performance problems, very little bandwidth is needed. For example, monitoring basic CPU, memory, and disk information takes little more than 10KB total per sampling interval per system. With today's corporate networks regularly deploying 100MB bandwidth, more than enough capacity exists.

- **Agentless monitoring isn't secure.**

Agentless monitoring can be as secure as IT administrators want it to be. Since most agentless monitoring solutions simply login to a remote system just like any user would, login security can be maintained by using appropriate security steps and credentials for the user defined for monitoring. In many cases, access can be granted only for the system components or resources that need to be monitored.
In addition, many agentless solutions have the ability to further secure remote monitoring via technologies such as Secure Shell (SSH) and HTTPS. For example, OmnoMonitor's agentless monitoring solution, OmniMonitor, can be configured to establish remote connections via SSH for Windows, UNIX, and Linux remote systems, and can also transmit data back to the OmniMonitor server via HTTPS.

- **Agentless monitoring doesn't provide enough granularities.**

Another misconception is that agentless monitoring cannot collect enough data or granular enough data to effectively monitor mission-critical systems. While it is true that certain low-level metrics may not be easily accessed by agentless monitoring, metrics that provide the ability to identify the most important monitoring issue – sustained performance problems – are easily gathered.
It should also be noted that in the past few years, many technology vendors have embedded interfaces (such as JMX, XML, SOAP, etc.) into their products that enable a wide variety of data to be gathered using agentless monitoring. Finally, research indicates that collecting too much data can dramatically drive up the costs of monitoring and in most cases; data collected from metrics other than "key indicator" metrics is rarely used.
Many enterprises have already purchased an agent-based infrastructure monitoring framework from one of the "big four" management vendors (Hewlett-Packard, IBM Tivoli, Computer Associates, and BMC). Although these solutions may be providing some value to the enterprise, most rely on high cost, agent-based technologies. Therefore, whether it is the high cost of maintenance, high overall CO, or simply extremely long implementation times, enterprises that rely solely on these solutions are not monitoring everything they'd like to monitor.
In addition, the "big four" agent-based solutions tend to be much more infrastructure-, system-, or network-focused at a time when application monitoring and end-user

experience analysis is critical. Finally, solid integrations between the "big four" and other lower TCO monitoring solutions can enable an effective co-existence strategy that makes sure the entire enterprise is covered.

3. One end-user experience monitoring technology is better than another.

When it comes to monitoring the end-user perspective, much debate has raged about the accuracy and overhead of different end-user monitoring approaches. The fact is that several end-user monitoring technologies are required, depending on the situation. There are three categories of end-user monitoring technologies – business process monitors, real-user monitors, and client monitors.

- Business process monitors run synthetic transactions to capture the performance and availability as experienced by end users. Synthetic monitors use agents distributed throughout the Internet to *simulate* how a particular website or application will react to heavy loads and provide information about application performance from distributed geographies. Synthetic monitoring solutions enable enterprises to drive active transactions against applications from outside the company firewall to monitor business processes externally.
- Client monitors sit on the actual end user's machine and capture and report on the performance and availability of real transactions executed by the end user.
- Real-user monitors sit off of the network node to capture and report on the performance and availability of URLs and transactions. These monitors measure application performance for many existing end users by tracking actual user traffic. Real-user measurement tools often use appliances or software probes to passively monitor all client interactions by attaching to a mirror port on the edge router in a data center.

All three technologies are important and each one has a role to play when creating a complete end-user monitoring solution.

4. Monitoring processes or services for an application suffices.

Today's applications – whether they be packaged applications, J2EE-, or. Net-based – are complex and span multiple systems and various technologies. Simply monitoring a few key services or processes will not provide a complete picture of application health and certainly will not provide the level of detail needed to troubleshoot thorny performance problems. In order to thoroughly understand application health, component monitoring is required.

5. Monitoring processes or services for an application suffices.
While it is tempting to monitor everything that uses electricity in the IT enterprise, 100-percent coverage is not necessary. IT enterprises typically consist of several systems that don't support business-critical functions or applications. The trick is in knowing which systems relate to critical business functions and which ones don't. Application relationship mapping technology can help ensure that IT is monitoring the systems, applications, and application components that really matter.

6. Monitoring all of the available metrics for a system or application is the best approach.

Performance problems tend to follow Paretos Rule – "80 percent of problems are generally caused by 20 percent of the system's or application's components." The challenge is in knowing which metrics are the "key indicators." Otherwise, either too much data is collected or the wrong metrics are monitored. Instead of monitoring every possible metric, IT administrators should look for monitoring solutions with built-in expertise regarding the most important metrics to watch.

7. Second or sub-second sampling rates are necessary.

The most important alerts needed when monitoring infrastructure performance and availability are the ones for sustained performance problems. Monitoring with second or sub-second intervals is not necessary to identify sustained performance issues and usually results in massive amounts of data that is never used, or "alert storms" that trigger too many people getting involved in a situation that may not be an emergency.
Second or sub-second monitoring is also likely to uncover events that are temporary or transitional, and not necessarily good indicators of performance problems that really impact end-user experience. While it is true that some aspects of performance and availability may execute faster than a second and therefore require sub-second sampling, these are few and far between, especially when it comes to basic infrastructure.

8. the best monitoring solutions include built-in, automatic corrective actions.

While corrective actions can be useful in a very few application performance situations, rarely do automated corrective actions see the light of day in most company's' monitoring strategies. Few system administrators are willing to trust a software tool to take action on its own.
Secondly, hardly any corrective actions (besides server reboot, temporary file cleanup, etc.) can be applied as the remedy for the types of performance problems that are showing up in current distributed applications. Most performance and availability problems are a result of application component issues. A custom, scripted approach is generally the best strategy for taking corrective actions, especially if it provides the ability to control whether it runs automatically or not.

9. The monitoring software has to reside in-house.

Application management outsourcing has gone mainstream. The reason for this is generally due to its lower TCO, faster implementation time, and ability to deliver insight without end users or even administrators needing to become application monitoring gurus. In addition, an outsourced monitoring strategy offers the ability to "test" a company's infrastructure from outside of its firewalls, preferably from multiple locations around the world.

10. A company's infrastructure monitoring strategy can operate in its own, detached silo.

Today's enterprise monitoring strategies are becoming more and more tied to other strategies within the organization. Pre-production testing and development need feedback from the real-time monitoring teams when designing new applications or tweaking existing ones. Change management strategies and solutions must be factored in when determining the cause of performance issues.
Business people must be involved in helping to set thresholds and service-level agreements (SLAs).
The bottom line is that monitoring must be a good corporate citizen and integrate with a myriad of other solutions, strategies, and processes.
Today's enterprises need to have the ability to ensure that their applications and systems are meeting their established performance and availability requirements in both pre-production and production. IT organizations must therefore have the ability to monitor, diagnose, and resolve critical problems across the entire application lifecycle. Effective performance optimization and management solutions must be able to span the entire performance lifecycle from development to production. Specifically, these solutions should enable IT to:

- Performance-test applications prior to rolling them out to production – mitigating the risks of application downtime.
- Use capacity planning to create the best architecture in the production environment
- By optimizing across cost, performance, and utilization requirements.
- Monitor, measure, and manage enterprise applications and the underlying infrastructure in production.
- Proactively diagnose and resolve application problems in both test and production environments.

Annexure – Monitors

Operating System Supported

Operating System Monitors	Version
Windows	XP Pro, NT 4.0, 2000, 2003
UNIX	Any version of any UNIX OS that supports telnet or SSH.
Linux	RedHat 7.x, 8.x, 9.x, Redhat Enterprise Linux 3.x AS and ES. *Note:* *Any version of Linux OS that supports telnet or SSH should work although not all have been tested.*

Application Monitors

Application/Device Monitors	Platform (Platforms on which the Applications run on)	Version
Apache Web Server	All	1.3.9, 1.3.12, 2.0.x
ATG Dynamo	All	5.6.1
BEA TUXEDO	Windows	6.5
BEA WebLogic	All	6.0, 7.0, 8.x
BroadVision	All	5.5, 6.0
CheckPoint Firewall	All	4.1 NG
Cisco Works	All	2000
Citrix MetaFrame	Windows	1.8, XP
COM+	Windows	2000
IBM DB2	Windows	6.x, 7.x

Application/Device Monitors	Platform (Platforms on which the Applications run on)	Version
IBM WebSphere Application Server	All (Including AS400)	3.x, 4.x, 5.x
IBM WebSphere MQ	Windows NT 4.0 and above, Solaris 2.6 and above	5.2 and above
F5 Big-IP	Not Applicable (sits on its own hardware)	4.0
MacroMedia ColdFusion	Windows	4.5.1
Microsoft ASP Server	Windows	4.0, 2000
Microsoft IIS	Windows	4.0, 2000
Microsoft SQL Server	Windows	6.5, 7.1, 2000
Microsoft Windows Media Player	Windows	7.x, 9.x
Microsoft Windows Media Server	Windows	All via perfmon
NetScape/iPlanet Server	All	3.6, iPlanet 4.x, 6.x
Novell SilverStream	Windows	2.4, 3.7.3
Oracle 9iAS	All	9I
Oracle JDBC	All	8I, 9I
Real One /Real Media Player	Windows	RealOne 2.x Real Media Player 7.x, 8.x
Real One /Real Media Server	Windows	All via perfmon
SAP (GUI)	All	SAPGUI 6.1/6.2 + R/3 4.6 SAPGUI 6.1 + R/3 4.7
SAP CCMS	Windows NT 4.0, Windows 2000, Linux and Solaris	SAP R/3 4.5B and above
Siebel Server Manager	Windows and all Unix	7.03, 7.04, 7.5.3
Siebel Web Server	Windows and all Unix	7.03, 7.04, 7.5.3
SunOne Web Server	All	6.x
Sybase Database	All	11.0, 11.5, 11.92, 12.x

Network devices and services

Applications.	Versions supported	Platforms Supported
CheckPoint Firewall-1	4.1 NG	All
Load Balancers and Networking		
F5 BigIP	4.0	Not applicable
CiscoWorks	2000	All Platforms
Network Services		
SNMP	1.0, 2.0 3.0 md5 authentication	All Platforms
SNMP Trap	1.0, 2.0, 3.0 3.0 md5 authentication	All Platforms
Formula (Bandwidth) Composite	NA	All
Ping	NA	NA
Web Service (SOAP)	NA	NA
DNS	NA	NA
Port	NA	NA
FTP	NA	All Platforms
RTSP	NA	All Platforms
Email	NA	All Platforms

Advanced Monitors

Applications	Versions supported	Platforms Supported
LDAP	NA	All Platforms
File	NA	NA
Directory	NA	NA
Log File	NA	NA
News (NNTP server)	NA	NA
Script	NA	NA
NT Dial-up	NA	NA
NT Event Log	NA	NA
Radius	NA	NA
Link Check	NA	NA

Server Monitors

Applications	Versions supported	Platforms Supported
CPU	NA	Windows, Unix, Linux
Disk Space	NA	Windows, Unix, Linux
Memory	NA	Windows, Unix, Linux
Web Server	NA	Windows, Unix, Linux
DHCP	NA	Windows, Unix, Linux
Network	NA	Windows, Unix, Linux
Service	NA	Windows, Unix, Linux

URL Monitors

Applications	Versions supported	Platforms Supported
URL	NA	NA
URL Content	NA	NA
URL List	NA	NA
URL Sequence	NA	NA

Solution Templates*

Solution Name	Description	Platform
Active Directory	Monitors performance and efficiency of Microsoft domain controllers	Windows NT/2000 Only
Exchange Server 2000	Monitors application health, message flow, and usage statistics for Exchange Server 2000	Windows NT/2000 Only
Exchange Server 2003	Monitors application health, message flow, and usage statistics for Exchange Server 2003	Windows NT/2000 Only
Exchange Server 5.5	Monitors application health, message flow, and usage statistics for Exchange Server 5.5	Windows NT/2000 Only
Oracle Database	Monitors performance, availability, and usage statistics for Oracle 8i, 9i and 10g databases	Windows, UNIX, Linux
Siebel Application Server	Monitors the availability and server statistics for Siebel Application Servers	Windows NT/2000 Only
Siebel Gateway Server	Monitors the availability of a Siebel Gateway Server	Windows NT/2000 Only
Siebel Web Server	Monitors the availability and server statistics for Siebel Web Servers	Windows NT/2000 Only
WebLogic	Monitors the availability and server statistics for WebLogic Application Server versions 6.x, 7.x, and 8.x	Windows, UNIX, Linux
WebSphere	Monitors the availability, server statistics, and deployed J2EE components on a IBM WebSphere Application Server 5.x	Windows, UNIX, Linux

* *Requires separate purchase and license key to enable.*

Solution templates deploy a combination of standard OmniMonitor monitor types and solution specific monitors with settings that are optimized for monitoring the availability, performance, and health of the target application or system. For example, the solutions for Microsoft Exchange monitoring include performance counter, event log, MAPI, and Exchange application specific monitor types.