

ELM ENTERPRISE MANAGER™

---

WHITE PAPER SERIES

REAL-TIME MONITORING  
FOR EXCHANGE 2000

- ELM ENTERPRISE MANAGER™ -  
REAL-TIME MONITORING  
FOR EXCHANGE 2000

---



Software

[www.tntsoftware.com](http://www.tntsoftware.com)

**2001 Main Street**

**Vancouver, WA 98660 USA**

**Phone 360.546.0878 • Fax 360.546.5017**

**[info@tntsoftware.com](mailto:info@tntsoftware.com)**

# TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>ABOUT TNT SOFTWARE.....</b>                                      | <b>1</b>  |
| <b>INTRODUCTION .....</b>   | <b>2</b>  |
| <i>How ELM Enterprise Manager Works .....</i>                       | 3         |
| <i>Exchange Monitor.....</i>  | 7         |
| <i>ELM Enterprise Manager Server .....</i>                          | 10        |
| <i>Monitoring your Exchange infrastructure .....</i>                | 11        |
| <i>Summary of Benefits .....</i>                                    | 12        |
| <i>Using this Paper.....</i>  | 13        |
| <b>EXCHANGE SERVER EVENTS.....</b>                                  | <b>14</b> |
| Collecting, Archiving And Monitoring Events.....                    | 14        |
| <b>PREPARING YOUR EXCHANGE SERVER .....</b>                         | <b>16</b> |
| Installing Agents .....   | 16        |
| <b>MONITORING EXCHANGE SERVER PERFORMANCE.....</b>                  | <b>19</b> |
| Establishing Performance Baselines .....                            | 19        |
| Trending .....  | 23        |
| Tuning Exchange 2000.....   | 23        |
| Capacity Planning.....  | 24        |
| Performance Alarms .....  | 24        |
| Practical Applications for Performance alarms .....                 | 26        |
| <b>COLLECTING, ANALYZING AND RESPONDING TO SYSTEM EVENTS .....</b>  | <b>27</b> |
| The Windows 2000 Event Structure .....                              | 27        |
| Interpreting Events .....   | 28        |
| Diagnostics Logging.....  | 28        |
| What to look for .....  | 29        |
| <b>CONCLUSIONS AND SUMMARY .....</b>                                | <b>30</b> |
| <b>EXPLANATION OF RECOMMENDED PERFORMANCE COUNTER OBJECTS .....</b> | <b>31</b> |
| Active Directory (NTDS).....  | 31        |
| Database (Active Directory & Exchange IS) .....                     | 32        |
| DNS .....   | 33        |
| Epoxy .....   | 33        |
| Internet Information Services Global .....                          | 33        |
| Logical Disk.....   | 34        |
| Memory .....  | 34        |
| Microsoft Exchange DS Access Cache.....                             | 36        |
| Microsoft Exchange Information Store.....                           | 37        |
| Microsoft Exchange IS Mailbox/Public .....                          | 37        |
| Microsoft Exchange MTA (MSEExchangeMTA) .....                       | 38        |
| Microsoft Exchange MTA Connections.....                             | 38        |
| Microsoft Gatherer.....   | 38        |
| Microsoft Gatherer Projects .....                                   | 39        |
| Microsoft Search .....  | 39        |
| Microsoft Search Catalogs .....                                     | 39        |
| Microsoft Search Indexer Catalogs.....                              | 40        |
| Network Interface .....   | 40        |
| Paging File .....   | 40        |
| Physical Disk .....   | 40        |
| Process.....  | 41        |
| Processor .....   | 42        |
| Redirector .....  | 42        |
| Server .....  | 42        |
| SMTP Server.....  | 44        |
| SMTP NTFS Store Driver.....   | 46        |
| System .....  | 46        |

---

## FOREWORD

## 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.NET, Windows XP, Windows 2000, Windows NT 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 on TNT Software, or any of our products, please visit our web site at <http://www.tntsoftware.com>, or send email to [info@tntsoftware.com](mailto:info@tntsoftware.com).

## CHAPTER

## 1

## INTRODUCTION

*Why ELM Enterprise Manager?*

**E**xchange 2000 Server represents a milestone for Microsoft. With more than 100 million seats sold worldwide, Microsoft Exchange Server has become the world's leading Windows-based electronic messaging application. With the release of Exchange 2000, Microsoft has taken messaging and collaboration to a whole new level. Because Exchange 2000 relies on Windows 2000's Active Directory, Internet Information Services and Domain Name Servers, administrators need to be more proactive than ever to ensure that critical messaging and collaboration services remain available to customers and end-users.

Architecturally, Exchange 2000 is radically different from its predecessors. Components that were built into Exchange 5.5 are now separate from Exchange with many of them now Windows 2000 components. Exchange 2000's performance and health relies heavily on external influences and components that include Internet protocols, name resolution and directory services. Exchange 2000 administrators need to take a wholistic approach when it comes to maintaining the reliability and availability of messaging and collaboration services.

The most common way of measuring Exchange 2000 performance is the "user-initiated alert system." This system works by users calling you to complain that the system is slow or not working. While it is a very simple and effective system, this is obviously not the best way to manage your systems.

Using ELM Enterprise Manager, you can implement proactive management techniques by monitoring your entire Exchange infrastructure. ELM Enterprise Manager enables you to monitor all aspects of your Exchange infrastructure, without requiring any add-on modules or the learning of any scripting languages. This type of wholistic management ensures that technical support resources are utilized in the most effective manner possible.

The goal of a responsible Exchange administrator is to maximize proactive management and minimize reactive management. With proactive management, you are in control; with reactive management you have no control. ELM Enterprise Manager allows you to proactively manage your Exchange infrastructure, permitting you to stay in control.

## HOW ELM ENTERPRISE MANAGER WORKS

---

ELM Enterprise Manager has two main modules, the ELM Server and the Agent. Agents are configured, installed and uninstalled directly from the ELM Console, which is the primary user interface for ELM Enterprise Manager. Agents can be physical agents that are installed as a service on the monitored system, or logical (remote) agents where the agent computer is monitored without installing any agent software on it.

You can configure the Agent to ignore certain events at a very granular level, reducing the network overhead. For example, if the Agent is on a busy logon domain controller, the administrator may choose not to send Success Audit messages over the network. In addition, you do not need to collect events to monitor event logs; you can use an Event Monitor to look for certain events and alert you when those events do or do not occur.

A separate thread in the Agent monitors 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 ELM Enterprise Manager thread processes the message. When a previously stopped service or driver starts, an informational event is created and again the ELM Enterprise Manager thread processes the message.

Another thread in the Agent monitors active processes. If a 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.

The ELM Console displays a variety of Event Views for events collected from the monitored systems. ELM Enterprise Manager also includes pre-defined security views, as well. You can also create additional filters and event views as needed.

### **ELM Enterprise Manager Agent**

ELM uses a **Service Agent** to monitor Windows NT, Windows 2000, and Windows XP computers in real-time.

A Service Agent can run on the following computers:

- Windows NT Workstation 4.0 w/SP4 or greater
- Windows NT Server 4.0 w/SP4 or greater
- Windows NT Server 4.0 - Enterprise Edition w/SP4 or greater
- Windows NT Server 4.0 - Terminal Server Edition w/SP4 or greater
- Windows 2000 Professional
- Windows 2000 Server/Advanced Server
- Windows XP Professional

Agents and Servers can have one-to-one or one-to-many relationships. A Server can monitor multiple Service Agents and a Service Agent can be monitored by multiple Servers. Each Service Agent maintains separate configuration, collection set and cache files for each Server that is monitoring it.

The Service Agent is comprised of a 32-bit, multithreaded executable called TNTAGENT.EXE, its companion DLLs and its configuration data. These files exist in the %systemroot%\TNTAgent and \Program Files\Common Files\TNT Software\ELM 3.0 folders on the monitored system. It runs as service under the LocalSystem account on each monitored system. Service Agents typically consume about 7-15MB of memory, and less than .1% of the overall CPU time on the monitored system. The amount of resources actually consumed depend on the number of Monitor Items applied to the Agent, the frequency at which those

Monitor Items are executed, and the amount of data generated by or being collected from the monitored system.

Service Agents create files in the TNTAgent folder that contain configuration data and cache information:


| File Name    | Description              |
|--------------|--------------------------|
| TNTAgent.dat | Agent configuration data |

If the communication or connectivity between the Service Agent and the ELM Server are interrupted, or if the ELM Server is down for any reason, the Agent will go into Cache Mode. In Cache Mode, the Agent will cache up to 10MB of data.

| File Name        | Description   |
|------------------|---|
| SERVER-XXX.CACHE | Event cache file (where SERVER is the name of the ELM Server(s) monitoring the Agent and XXX represents the three character product edition: ELM Enterprise Manager - EEM, ELM Log Manager - ELM, and ELM Performance Manager - EPM). |

This cache file will be located in one of two places:

- If the Agent has an ELM Server or an ELM Console installed, the cache file will reside in the \Program Files\Common Files\TNT Software\ELM 3.0 folder.
- If the Agent does not have an ELM Server or an ELM Console installed, the cache file will reside in the TNTAgent folder.

 **Note**

Agents monitored by multiple ELM Servers may not go into cache mode for all Servers at the same time. The cache mode behavior on an Agent monitored by multiple Servers will depend on the nature of the communications disruption. For example, if one of the ELM Servers is down for maintenance, the Agent will go into cache mode for that ELM Server only. The Agent will only go into cache mode for those ELM Servers with which it cannot communicate.

Also note that disabling an Agent or its Monitor Items does not put it into cache mode. This means that data transmitted by the Agent after re-enabling an Agent is not cached data and will therefore trigger applicable Rules and Notification Methods.

Windows-based Monitor Items that can be used for Service Agents include:

### **Event Collector/Alarm**

Event Collectors are used to monitor and collect events from the event logs on Windows NT, Windows 2000, and Windows XP. You can specify the events you want to collect based on a variety of event criteria, including event log, type, source, event ID, and event details. Event Alarms are used to trigger action and/or notification when an event does or does not occur. Event Alarms can be configured for Windows NT, Windows 2000 and Windows XP Agents.

### **Cluster Monitor**

Event Log Monitor used a combination of a Server Agent and ELM Cluster Monitor (a cluster-aware resource DLL) to provide enhanced monitoring and performance data collection in Windows NT and Windows 2000 clusters. The Cluster Monitor in ELM Enterprise Manager 3.0 replaces these two components with an integrated cluster-aware Service Agent. You can use a Cluster Monitor to monitor your Exchange and other clusters, and watch cluster system and cluster registry events. The Cluster Monitor thread can monitor any or all of the seven Cluster APIs: cluster events, quorum events, network events, node events, group events, resource events and registry events.

### **File Monitor**

Event Log Monitor used a separate executable called FileMon.exe to monitor ASCII flat files. FileMon has been integrated into this version of ELM to provide you easier monitoring of flat files. You can monitor individual files, an entire directory of files, or an entire directory tree of files. File Monitors can be used to monitor IIS log files, as well as any other non-circular log file.

### **Performance Alarm**

Performance Alarms monitor performance objects, counters and instances and can generate a variety of Notification Methods when a counter or instance of a counter is greater than, less than or equal to your specified threshold for your specified duration. You can use this monitor item on Windows NT, Windows 2000, and Windows XP. You can use Performance Alarms to keep tabs on any published performance object and counter, including those published for Exchange Server email queues.

### **Performance Data Collection Set**

This Monitor Item is used to collect and store performance data from Windows NT, Windows 2000, and Windows XP computers. A Collection Set is a group of performance counters that are collected at the same time. You can use multiple Performance Data Collection Sets that contain different groups of counters, or a single Performance Data Collection Set that contains all of the counters you want to collect. Using Performance Data Collection Sets, you can regularly collect performance data for your Exchange servers for baselining, trending and capacity planning.

### **Process Monitor**

If you need to monitor individual processes, you can do so with a Process Monitor. The Process Monitor is multi-functional; it can let you know when a process has exceeded the threshold of CPU usage you specify, and it can track when processes are instantiated or terminated.

### **Service Monitor**

Service Monitor items are used to monitor individual services and devices on Windows NT, Windows 2000, or Windows XP. Service Monitors can generate notification when a service or device is stopped, started, paused or resumed. In addition, Service Monitors can alert you when it finds a service or device set to Automatic startup that is not running.

### **SQL Monitor**

Using SQL Monitors, you can periodically execute SQL queries against a database and generate a variety of notification options if the results returned are different from what is expected. SQL Monitors support both Windows NT and SQL Server authentication, making them easy to fit into your existing SQL security environment.

### **WMI Monitor**

If you are using Windows Management Instrumentation--the Microsoft implementation of Web-Based Enterprise Management (WBEM)--you can use WMI Monitors to query a WMI namespace and database. If the results of the query change, a variety of notification options can be executed.

In addition to the above Windows-based Monitor Items, you can also monitor TCP/IP-based applications. TCP/IP-based application monitoring is performed through the use of the following Monitor Items and Receivers:

### **FTP Monitor**

This monitor is used to monitor a specific FTP URL. The Service Agent periodically establishes an FTP connection to the URL and port specified. If the response is negative or slower than expected a variety of actions and notification options can be triggered.

### **TCP Port Monitor**

If you need to monitor a TCP port on any TCP/IP-based system or device, you can use a TCP Port Monitor to do so. Simply specify the port you wish to monitor and the expected response time in seconds.

### **Ping Monitor**

The Ping Monitor is used to send period ICMP echo requests to the Agent(s) being monitored. You can specify the size of the echo request packets and the number of packets that are sent.

### **POP3 Monitor**

POP3 Monitors are used to periodically check a POP3 mailbox for availability. The Service Agent periodically establishes a POP3 connection to the server and port specified using the mailbox credentials you enter. If the response is negative or slower than expected a variety of actions and notification options can be triggered.

### **SMTP Monitor**

SMTP Monitors are used to keep tabs on SMTP hosts, gateways and services. The Service Agent periodically establishes an SMTP connection to the server and port specified. If the response is negative or slower than expected a variety of actions and notification options can be triggered.

### Web Page Monitor

Web Page Monitors are used to monitor HTTP and HTTPS URLs. The ELM Enterprise Manager Server periodically establishes an HTTP connection to the server and port specified. If the response is negative, slower than expected, or if the content has been changed, a variety of actions and notification options can be triggered.

### SNMP OID

ELM includes an SNMP Monitor that enables you to query an SNMP Object ID (OID) and trigger notification if the value is greater than, less than or equal to a specified value. The SNMP Monitor includes an object browser that enables you to query the objects on an SNMP-capable computer, and select specific objects for monitoring.

### SNMP Receiver

SNMP systems are monitored using IP Agents. The ELM Server listens for SNMP Traps from registered IP Agents, and treats the incoming traps as events.

### Syslog Receiver

Unix, Linux and other Syslog clients are monitored using IP Agents. The ELM Server listens for Syslog messages from registered IP Agents, and treats the incoming messages as events.

## EXCHANGE MONITOR

---


ELM Enterprise Manager includes an **Exchange Monitor** that enables you to perform end-to-end monitoring of Microsoft Exchange 5.5 or Exchange 2000. This type of monitoring allows you to specify a custom quality of service (QoS) threshold for internal email delivery, and to be notified when that threshold is not met.

To do this, the ELM Enterprise Manager Server uses three mailboxes in your Exchange organization for each Exchange Monitor:

- **Administrator mailbox.** This mailbox is used to establish a MAPI session to an Exchange Server in your organization. No email is sent to/from this mailbox; it is used for session purposes only.
- **Source Mailbox.** This is one of the two end-points; in this case, this is the originating mailbox.
- **Target (Destination) Mailbox.** This is the other end-point; in this case, this is the destination mailbox.

Messages sent between these two mailboxes are automatically removed. You do not need to perform any mailbox maintenance on any of the end-point mailboxes you use.

When a message is sent from the source mailbox, an internal clock is started. If the message is not received by the target mailbox within your QoS threshold, the assigned actions will be executed.

 **Note**

Creating an Exchange Server Monitor requires the MAPI subsystem to be installed on your ELM Enterprise Manager Server. This is accomplished by installing one of the many Exchange Server clients on to your ELM Server. Due to its lightweight nature, the Exchange 5.0 client is the recommended client; however, any Exchange client, including any version of Microsoft Outlook will work.

In addition, you must also set the Exchange client as the default E-mail client in **Control Panel | Internet Options | Programs | E-mail**. If you do not do this, you will receive a message that no default mail client was configured.

You can create multiple Exchange Monitors, enabling you monitor end-to-end mail delivery between all Exchange servers in your organization.

» To create an Exchange Monitor:

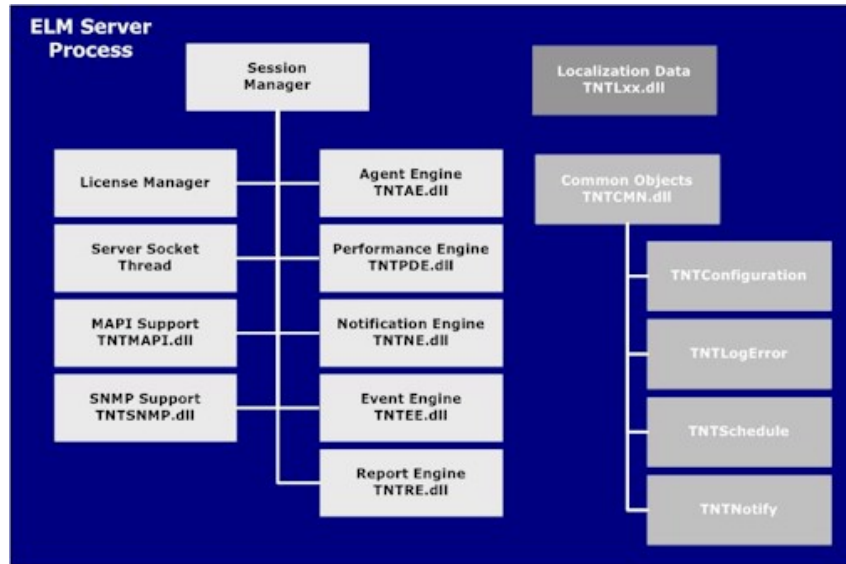
1. Right-click on the Monitor Items container in the ELM Console and select **New | Monitor Item**. The Create Monitor Wizard will appear. Click **Next** to continue.
2. Select **Exchange Monitor** from the Monitor Type dropdown and click **Next** to continue. ELM Enterprise Manager will verify the existence of the required MAPI subsystem files. If those files are not present, an error message will appear and you will not be able to continue.
3. The Exchange Monitor dialog will appear. Click the ellipses button (...) and select an Administrative mailbox. Click **Next** to continue.
4. Using the ellipses buttons, select a **source** and **destination** mailbox. Selecting source and destination mailboxes on different servers can be used to perform end-to-end monitoring of your Exchange infrastructure.
5. Set the **Warn if slower than \_\_\_ seconds** to the desired quality of service threshold.
6. To generate a warning each time the response is slower than the number of seconds specified, check the box that says **Continue to warn after first failure**.
7. The Monitor Action dialog will appear. There are four tabs you can configure: Failed, Quality of Service, Success and Warning. Each tab has several configuration options.
  - **Enable**. When this box is checked, the Action is enabled and will execute as configured. Uncheck this box to disable the Action.
  - **Create New Alert Entry**. Selecting this checkbox to cause an Alert to be generated and posted to the Alerts container within the Console snap-in.

- **Create Application Event Log Entry.** Selecting this enables you to cause a customizable event to be logged to the Application log on the ELM Enterprise Manager server. Use the Variables button to specify any additional variables you want the event to include.
  - **Net Send Message.** Using this option enables you to send a popup message over the network. The target system must have the Messenger Service (NT/2000/XP/.NET) or WinPopUp (Win9x/WinMe) running in order to receive the Net Send Message. Use the Browse button to browse for a target computer, or manually enter the name of the computer to which you want to send messages in the Computer Name field.
  - **Run Command.** This option provides a mechanism for executing a batch file, running a command line application or launching a script (CScript or WScript). Use the Edit button to create and edit any scripts or command line parameters you want to execute.
8. Configure each tab as desired and click **Next** to continue.
  9. Click **Next** to continue. The Schedule dialog box will appear.
    - On the **Scheduled Interval** tab, set the frequency at which you want to execute this Exchange Monitor. Leaving the interval at the default of 1 second will enable you to monitor your event logs in real-time.
    - On the **Scheduled Hours** tab, configure the hours and days you want this Exchange Monitor active.
  10. Click **Next** to continue. Enter a Name and Description and make sure Make this a default selection for new items is not checked.
  11. Click **Finish** to save the Monitor Item.
  12. Click **OK** to acknowledge the item creation. When prompted to create another item, click **No**.

**ELM ENTERPRISE MANAGER SERVER**

---

The ELM Server process is a 32-bit, multi-threaded, COM-based application that runs as a service on Windows NT, Windows 2000 and Windows XP computers. This process is implemented as a set of engines and other components that provide a variety of functions, as shown in the following illustration:



**Session Manager**

The Session Manager manages the state of the ELM Server. In addition, when an ELM Console is communicating with an ELM Server, it is communicating via the Session Manager. This is all COM-based communication.

**Agent Engine**

The Agent Engine contains the non-performance related Monitor Items. In addition, the Agent Engine manages the COM objects that represent each Service Agent.

**Performance Engine**

The Performance Engine contains the performance-related Monitor Item. In addition, it includes the list of all performance objects and counters that are displayed in the Performance Data container within the ELM Console.

**Notification Engine**

The Notification Engine executes all notification methods. It manages both the COM objects that represent the Notification Methods, and the Notification Method job queue.

**Event Engine**

The Event Engine manages all of COM objects that represent Event Filters, all Views (Events, Alerts, and Personal), and Rules. In addition, it also executes and maintains both the SNMP and Syslog receiver threads.

**Report Engine**

The Report Engine manages all of the COM objects that represent all Reports. In addition, it manages a Report Job Queue (for scheduled reports). The Report Engine also creates,

manages and maintains the database used by the ELM Server. It also manages report editing, and handles both running and previewing reports.

### **License Manager**

The License Manager maintains current licensing state. The License Manager provides license information to the Agent Engine to ensure that the proper types of Agents are licensed, and that the proper Monitor Items are available to each Agent. It also provides information to the Event Engine to facilitate the licensing of IP Agents that transmit SNMP Traps or Syslog messages to the ELM Server.

### **Server Socket Thread**

This thread is responsible for TCP two-way socket-based communication with all Service Agents. In addition to establishing outbound communication with a Service Agent, this thread also listens for inbound communication from an ELM Agent. This thread is also responsible for reporting the current values of all of the ELM Server's published performance counters. The component of the thread that is responsible for reporting this data does not do anything unless and until something requests the data via the ELM Server performance library file.

### **MAPI Support**

This component provides support for the MAPI Email Notification Method. In addition, in ELM Enterprise Manager, this component supports the Exchange Monitor item.

### **SNMP Support**

This component provides support for the SNMP receiver, and the SNMP Trap and OID Notification Methods.

### **Common Objects**

This component provides support for updating, managing and saving server configuration data, ELM Server logging, scheduled items, and actions that are triggered from Monitor Items.

### **Localization Data**

This component provides support for non-U.S. English (localized) versions of Windows.

## **MONITORING YOUR EXCHANGE INFRASTRUCTURE**

---

At a minimum, you should monitor the following aspects of your Exchange 2000 infrastructure:

**Event logs:** Ideally, you will want to monitor the event logs on all Exchange 2000 servers and all DNS servers, Active Directory domain controllers and global catalog servers.

**Performance Data:** Regularly collecting performance data is the only way to baseline and trend a server's true performance over time. Without establishing a baseline and then watching performance trends you will not be able to accurately perform capacity planning. In addition to collecting performance data, you'll want to watch various performance counters for thresholds. For example, you should watch queue-based performance counters to make sure a message queue is backed up. Similarly, you should also watch important processor, memory, disk and network counters, as well.

**Quality of Service:** Using an Exchange Monitor, you can monitor MAPI availability as well as the quality of service (delivery time thresholds) between any two servers in your Exchange organization. Because you can create an unlimited number of Exchange Monitors, you can perform end-to-end and quality of service monitoring between all Exchange servers in your organization.

**Services:** Exchange 2000 and its related components run as services under Windows 2000. If one or more of these services terminates or otherwise fails, it could mean substantial downtime, stuck messages and general server unavailability.

**Processes:** Each of Exchange 2000's services represents a specific process. In addition to monitoring these processes for instantiation and termination, you'll want to watch them to make sure they do not monopolize the CPU and cause the system to become unresponsive.

**TCP/IP Services:** You should also regularly monitor TCP/IP-based services on your Exchange 2000 servers, such as Ping, HTTP/HTTPS (Outlook Web Access), SMTP and POP3.

**IIS Log Files:** With its tight interdependency on Internet Information Services, it is also important to monitor your IIS log files, such as the World Wide Web Publishing and SMTP services log files.

**DNS:** Healthy DNS name resolution is essential to the operations of Active Directory, Exchange 2000, IIS and Windows 2000.

**Active Directory:** Active Directory is the lynchpin of Exchange 2000.

## SUMMARY OF BENEFITS

---

ELM Enterprise Manager provides total monitoring for your Exchange 2000 infrastructure. By monitoring events, performance data, services and processes in real-time, ELM Enterprise Manager enables you to take a proactive approach to server management. With ELM Enterprise Manager you can:

- Monitor your servers' event logs in real-time, and store the events in a database for archiving and auditing;
- Collect and store all published performance counters, including Exchange 2000, Active Directory, DNS and IIS performance counters, for baselining, trending and capacity planning;
- Monitor end-to-end MAPI availability and quality of service;
- Collect and store system configuration data, including security accounts, shares and device drivers;
- Create top-level and drill-down reports of your event data and performance data;
- Send and receive alert notifications, including via email and pager, when events occur, services fail or performance exceeds thresholds;
- Automatically restart failed services;
- Stop and start services locally and remotely;
- Kill processes locally and remotely;
- Monitor the availability of SMTP gateways, Outlook Web Access, and POP3 servers;
- and more...

**USING THIS PAPER**

---

This paper is based on ELM Enterprise Manager 3.0, Windows 2000 with Service Pack 2, Internet Information Services, including the NNTP, SMTP and W3SVC services, Active Directory, Windows 2000 DNS, and Exchange 2000 Enterprise Server with Service Pack 2. We assume that you have already installed and configured ELM Enterprise Manager. For information on installing and configuring ELM Enterprise Manager, review the [Getting Started Guide](#), as well as the product Help file (EEMMC.CHM) which contains complete product documentation.



## EXCHANGE SERVER EVENTS

### *Collecting and Analyzing Events on a Microsoft Exchange 2000 Server*

Windows 2000, Internet Information Services and Exchange 2000 are capable of generating a variety of error and event messages. These messages, which contain important information about the completion of normal tasks, as well as error messages about problems that occur, should be reviewed daily. Microsoft has documented a lot of these events and made them available online at <http://www.microsoft.com/exchange/en/60/help/default.asp>. Microsoft also provides additional resources which are useful in deciphering these events. Links to those resources can be found at <http://www.microsoft.com/exchange/en/60/help/support.htm>.

But you can't research an event or correct any problems until you're aware of them. It's not practical to manually monitor the event logs on a single server, let alone multiple servers.

### **COLLECTING, ARCHIVING AND MONITORING EVENTS**

---

The ELM Server receives the events from each monitored Agent and displays them in the Events window, which can be sorted a number of ways by clicking the desired column once or twice with the left mouse button.

In addition, you can create customized views for displaying specific events. ELM Enterprise Manager ships with several pre-defined views, including views designated as security views. You can further modify the pre-defined views, or create your own from scratch. This is helpful if you want to isolate events from a specific computer, a specific type of computer, a group of computers, or a specific type of event from one or more computers.

The Console uses the defined Event Filters against events as they arrive. Event Filters are used to process, select, and group events. Event Filters are also used with Notification Rules, which in turn trigger Notification Methods. In addition, Event Filters are processed to create Views.

Depending upon what components are installed, Exchange 2000 registers some or all of the following event sources:

|                           |                           |                          |
|---------------------------|---------------------------|--------------------------|
| CCMailProxy               | MSEExchangeAdmin          | MSEExchangeISPublicStore |
| DAVEX                     | MSEExchangeAL             | MSEExchangeKMS           |
| EDB                       | MSEExchangeCCMC           | MSEExchangeMig           |
| ESEBACKUP                 | MSEExchangeChat           | MSEExchangeMigDS         |
| ESE98                     | MSEExchangeCoCo           | MSEExchangeMSMI          |
| ESENT                     | MSEExchangeCONF           | MSEExchangeMTA           |
| EXCDO                     | MSEExchangeDCS            | MSEExchangeMU            |
| EXIFS                     | MSEExchangeDirExp         | MSEExchangeNOTES         |
| EXODBPC                   | MSEExchangeDirImp         | MSEExchangeNText         |
| EXOLEDB                   | MSEExchangeDSAccess       | MSEExchangeNWExt         |
| EXPROX                    | MSEExchangeDSExp          | MSEExchangeSA            |
| Exchsync                  | MSEExchangeDSImp          | MSEExchangeSetup         |
| ExSMTP                    | MSEExchangeDS             | MSEExchangeSRS           |
| ExWin32                   | MSEExchangeDX             | MSEExchangeT120          |
| GWISEProxy                | MSEExchangeES             | MSEExchangeTransport     |
| IBMProxy                  | MSEExchangeFB             | MSEExchangeWEB           |
| IMAP4SVC                  | MSEExchangeFBPublish      | MSPProxy                 |
| InternetProxy             | MSEExchangeGWISE          | NOTESProxy               |
| MAPITCP                   | MSEExchangeH323           | POP3SVC                  |
| MSADC                     | MSEExchangeIM             | WorkflowAuditTrail       |
| MSEExchangeIMAP4Interface | MSEExchangeIpConf         | WorkflowEventSink        |
| MSEExchangeNNTPInterface  | MSEExchangeIS             | X400Proxy                |
| MSEExchangePOP3Interface  | MSEExchangeISMailboxStore |                          |

Events from these sources appear in the Application log. Exchange 2000 is capable of logging thousands of different informational, warning and error events (in fact, there are over 1,100 different events for the source MSEExchangeISMailboxStore). It is critical to find out about some events immediately; however, given the large number of events Exchange can generate it is impossible to manually wade through them all.

ELM Enterprise Manager automates and simplifies the process of collecting and filtering events. Both Views and Notification Methods are based on (and generated by) filters. You can use several levels of filtering so views only show specific events, and so that notification or corrective action only take place for certain events.

## PREPARING YOUR EXCHANGE SERVER

### *Installing and Configuring the Agent on a Microsoft Exchange 2000 Server*

The ELM Enterprise Manager Agent is deployed remotely from within the ELM Enterprise Manager Console. You will need administrative privileges on the Exchange server to do this, but you do not need any special privileges within Exchange itself.

#### **INSTALLING AGENTS**

---

» To install a Service Agent:

1. Right-click on the Agents container in the ELM Console and select New | Agent. The Agent Installation Wizard will launch. If the Welcome dialog is displayed, click **Next** to continue.
2. From the dropdown, select **Service Agent** and click **Next** to continue.
3. Enter the **name** or **IP address** of the system you want to monitor. Click the **Browse** button to browse your network if you are unsure of the computer's name. Click **Next** to continue.
4. Using the dropdown, select **Service Agent**. Click **Next** to continue.
5. In the **Listen on TCP Port** field, enter the TCP port on which you want the Agent to listen. Service Agents communicate with the ELM Server over TCP/IP sockets. By default, Service Agents listen on TCP port 1253. You can change the port used by the Agent by selecting an alternative TCP port. Use the **Test** button to verify that the port is available.

#### **Note**

Once an Agent has been configured to listen on a specific port, you cannot change the port. If you want the Agent to listen on a different port, you will need to remove and re-add the Agent using the new port.

- Click **Next** to continue. The copy file process will begin. The Agent executable, companion DLL files and configuration data will be copied to the target computer.



### Important

In order to install a Service Agent, you must have administrative rights on the monitored system. ELM will attempt to install the Service Agent using your current credentials (e.g., the account you're logged on with); if this account does not have administrative rights on the Service Agent, you will be prompted to specify alternate credentials to perform the installation. Another alternative is to use the setup package you downloaded to [install the Service Agent remotely](#).

- The Agent Categories dialog will appear. Modify the **Categories** field as desired, or leave the default entries. Click **Next** to continue.
- If there are no monitor items configured, click **Finish** to complete the Wizard. If there are monitor items, the **Select Monitors** dialog will appear. In this event, click on each Monitor Item you want applied to this Service Agent. To create a new Monitor Item, right-click in the white space in this dialog and select **New Monitor Item**. Click **Next** to continue. Click **Finish**. A service called the TNT Agent will be installed and started, and real-time monitoring of the Service Agent will commence.
- Click **OK** to acknowledge the Agent installation. When prompted to install another Service Agent, click **Yes** or **No**, depending on your needs.

You can also monitor Exchange Servers without having to install a Service Agent. This is done by using a Remote Agent.

➤ To install a Remote Agent:

- Right-click on the Agent container in the ELM Console snap-in and select New | Agent. The Agent Installation Wizard will launch. If necessary, click **Next** to continue.
- From the dropdown, select **Remote Agent** and make sure the **IP Agent** checkbox is not checked. Click **Next** to continue.
- Enter the **name** or **IP address** of the system you want to monitor. Click the **Browse** button to browse your network if you are unsure of the computer's name. Click **Next** to continue.
- Modify the **Categories** field as desired, or leave the default entries. Click **Next** to continue.
- Click **Next** to continue. If there are no monitor items configured, click **Finish** to complete the Wizard. If there are monitor items, the **Select Monitors** dialog will appear. In this event, click on each Monitor Item you want applied to this Service Agent. To create a new Monitor Item,

right-click in the white space in this dialog and select **New Monitor Item**. Click **Next** to continue. Click **Finish**.

6. Click **OK** to acknowledge the Agent installation. When prompted to install another Remote Agent, click **No**.

The Remote Agent will be added to the list of monitored systems, and the selected Monitor Items for this Remote Agent will be executed according to their settings.

## MONITORING EXCHANGE SERVER PERFORMANCE

### *Baseline and Trend your Microsoft Exchange Servers*

Before you can properly analyze growth – and its effect on performance – you must have a starting point for your analysis. This starting point is known as a **baseline**. A baseline is the initial performance snapshot; performance data that is collected immediately prior to putting the server into production. Without a baseline, it is impossible to determine performance trends over time, or perform any useful capacity planning.

### **ESTABLISHING PERFORMANCE BASELINES**

---

Baselining a Windows 2000 system is fairly easy, especially with a tool like ELM Enterprise Manager at your disposal. Windows 2000 publishes an extensive list of performance counters; this list is further extended to include Exchange-specific performance counters when Exchange is installed.

Each published object has one or more counters that provide information on the object's utilization. We recommend that you collect data for both Windows 2000 and Exchange 2000 objects. In an Exchange environment, there are several types of objects that need to be monitored. While the list of available objects is long, you don't need to collect all of them to get an accurate picture of your server's health.

The areas that need to be monitored are CPU, memory, disk, network, the Information Store, IIS, Active Directory and DNS. Specifically, we recommend monitoring the following objects and counters:

REAL-TIME MONITORING FOR EXCHANGE 2000

| <b>Object</b>                                | <b>Counter</b>   | <b>Object</b>                        | <b>Counter</b>  |
|--|--|--------------------------------------|---|
| Processor                                    | Interrupts/sec<br>% Processor Time   | <b>Server</b>                        | Bytes Total/sec<br>Errors Access Permissions<br>Errors Granted Access<br>Errors Logon<br>Errors System<br>Pool Nonpaged Bytes<br>Pool Nonpaged Failures<br>Pool Nonpaged Peak<br>Pool Paged Bytes<br>Pool Paged Failures<br>Pool Paged Peak<br>Server Sessions<br>Sessions Errored Out<br>Sessions Timed Out<br>Work Item Shortages |
| Process                                      | Handle Count<br>Pages Faults/sec<br>Page File Bytes<br>% Processor Time<br>Private Bytes<br>Thread Count<br>Virtual Bytes<br>Working Set             |                                      |   |
| Paging File                                  | % Usage<br>% Usage Peak  |                                      |   |
| System                                       | % Registry Quota in Use<br>Processor Queue Length<br>System Calls/sec<br>System Up Time  | <b>Memory</b>                        | Available MBytes<br>Cache Faults/sec<br>Committed Bytes<br>Commit Limit<br>Page Reads/sec<br>Page Writes/sec<br>Pages/sec<br>Page Faults/sec<br>% Committed Bytes in Use<br>Pool Nonpaged Bytes   |
| LogicalDisk                                  | Avg. Disk Queue Length<br>Avg. Disk Sec/Transfer<br>Current Disk Queue Length<br>Disk Reads/sec<br>Disk Writes/sec<br>Free Megabytes<br>% Free Space | <b>Network Interface</b>             | Bytes Total/sec<br>Packets Outbound Errors<br>Packets Received Errors   |
| Database<br>(Exchange &<br>Active Directory) | Cache % Hit<br>Cache Size<br>Log Record Stalls/sec<br>Log Threads Waiting<br>Table Open Cache % Hit  | <b>PhysicalDisk</b>                  | Avg. Disk Sec/Read<br>Avg. Disk Sec/Transfer<br>Avg. Disk Sec/Write<br>Current Disk Queue Length  |
| Epoxy  | Client Out Que Len   | <b>MSExchangeMTA<br/>Connections</b> | Queue Length  |

REAL-TIME MONITORING FOR EXCHANGE 2000

| <b>Object</b>              | <b>Counter</b>  | <b>Object</b>                                       | <b>Counter</b>  |
|----------------------------|---|---|---|
| MSExchangeIS<br>Public     | Avg Delivery Time<br>Avg Local Delivery Time<br>Folder Opens/sec<br>Message Opens/sec<br>Messages Submitted<br>Receive Queue Size<br>Send Queue Size<br>Single Instance Ratio   | <b>MSExchangeMTA</b>                                | Messages/sec<br>Work Queue Length   |
| MSExchangeIS               | RPC Operations/sec<br>User Count<br>VM Total 16MB Free Blocks   | <b>MSExchangeIS<br/>Mailbox</b>                     | Avg Delivery Time<br>Avg Local Delivery Time<br>Folder Opens/sec<br>Message Opens/sec<br>Messages Submitted<br>Receive Queue Size<br>Send Queue Size<br>Single Instance Ratio   |
| Active Directory<br>(NTDS) | DRA Inbound Bytes Total<br>DRA Inbound Bytes Not Compressed<br>DRA Inbound Bytes Compressed (Before)<br>DRA Inbound Bytes Compressed (After)<br>DRA Inbound Obj. Updates Rem. in Packet<br>DRA Outbound Bytes Total<br>DRA Outbound Bytes Not Compressed<br>DRA Outbound Bytes Compressed (Before)<br>DRA Outbound Bytes Compressed (After)<br>DRA Pending Replication Synchronizations<br>DS Directory Searches/sec<br>DS Directory Reads/sec<br>DS Directory Writes/sec<br>DS Threads in Use<br>Kerberos Authentications/sec<br>LDAP Bind Time<br>LDAP Client Sessions<br>LDAP Searches/sec<br>NTLM Authentications/sec | <b>MSExchangeDS<br/>Access Caches</b>               | Cache Expiries Total<br>Cache Hits/sec<br>Cache Misses/sec<br>Cache Hits Total<br>Cache Inserts Total<br>Cache Inserts/Sec<br>Cache Misses Total<br>Cache Misses/Sec<br>Cache Timeout<br>DN Entries<br>DN Entries Memory<br>LDAP Searches Total<br>LDAP Searches/Sec<br>LRU Entries<br>Max Entries<br>Max Memory Size<br>Not Found DN Entries<br>Not Found DN Entries Memory<br>Not Found GUID Entries<br>Not Found GUID Entries Memory<br>Outstanding Async Notifies<br>Outstanding Async Reads<br>Outstanding Async Searches<br>Search Entries<br>Search Entries Memory<br>Total Entries<br>Total Entries Memory<br>Total Memory Size |
| DNS                        | Dynamic Update Received<br>Dynamic Update Rejected<br>Dynamic Update TimeOuts<br>Recursive Queries<br>Recursive Query Failure<br>Recursive Send TimeOuts<br>Secure Update Failure<br>Zone Transfer Failure  | <b>Internet<br/>Information<br/>Services Global</b> | BLOB Cache Hits<br>BLOB Cache Misses<br>Current File Cache Memory Usage<br>File Cache Hits<br>File Cache Misses<br>Total Blocked Async I/O Requests<br>URI Cache Hits<br>URI Cache Misses   |

REAL-TIME MONITORING FOR EXCHANGE 2000

| Object                    | Counter   | Object                                   | Counter  |
|---------------------------|---|--|--|
| Microsoft Gatherer        | Documents Filtered<br>Performance Level<br>System IO Traffic Rate<br>Reason to back off   | <b>MS Gatherer Projects</b>              | Crawl in Progress Flag<br>Current Crawl is Incremental<br>Gatherer Paused Flag<br>URLs in History<br>Waiting Documents |
| Microsoft Search Catalogs | Failed Queries<br>Queries Results<br>Successful Queries   | <b>Microsoft Search Indexer Catalogs</b> | Merge Progress   |
| Microsoft Search          | Active Threads<br>Failed Queries<br>Succeeded Queries<br>Threads  | <b>SMTP NTFS Store Driver</b>            | Messages in the queue directory  |
| SMTP Server               | Badmailed Messages (All)<br>Bytes Received Total<br>Bytes Received/sec<br>Bytes Sent Total<br>Bytes Sent/sec<br>Cat: Address lookups not found<br>Cat: Categorizations failed (All)<br>Cat: LDAP Bind Failures<br>Cat: LDAP Connection Failures<br>Cat: LDAP Paged Search Failures<br>Cat: LDAP Search Failures<br>Cat: LDAP Searches Pending Completion<br>Cat: Mailmsg duplication collisions<br>Cat: Recipients NDRd (All)<br>Cat: Senders Unresolved<br>Inbound Connections Current<br>Inbound Connections Total<br>Local Queue Length<br>Local Retry Queue Length<br>Messages Refused for Address Objects<br>Messages Refused for Mail Objects<br>Messages Refused for Size<br>Messages Received Total<br>Messages Sent Total<br>NDRs Generated<br>Outbound Connections Current<br>Outbound Connections Refused<br>Outbound Connections Total<br>Remote Queue Length<br>Remote Retry Queue Length<br>Total Connection Errors<br>Total DSN Failures<br>Total Messages Submitted | <b>Redirector</b>                        | Bytes Total/sec<br>Current Commands<br>Network Errors/sec<br>Reads Denied/sec<br>Writes Denied/sec                     |

Note: Physical disk counters are only available after **diskperf -y** is run to enable them. This command is run from the command line, and it requires a reboot to take effect. On systems using software RAID, the command used should be **diskperf -ye**. There is a space between diskperf and the option specified (-y or -ye). Physical disk counters can be turned off by using **diskperf -n** and rebooting. Collecting physical disk counters does add additional overhead to the system, but this overhead is well worth the benefits of having the physical disk counters available.

These counters should be collected continuously at regular intervals throughout the life of your Exchange server. Regular analysis of this data allows you to trend the growth of your server and perform capacity planning.

### T R E N D I N G

---

**Trending** is the analysis of data collected over time. Analyzing trends is important for two reasons. First, it allows an administrator to assess usage patterns. Second, it provides a mechanism for capacity planning. Usage of Exchange resources always has highs and lows, especially within 24x7 environments. For this reason, it is important to collect performance data at regular intervals over a 24-hour period. This will provide you with an accurate picture of usage, without adding the overhead associated with continuous real-time monitoring (e.g., collecting every second). We recommend collecting performance data at 5-15 minute intervals; by doing so, you'll learn a great deal about your Exchange environment. You'll also start to recognize performance patterns. By understanding these patterns, you'll be able to quickly spot out-of-bounds activity.

As demands for Exchange resources increase – due to an increase in the number of users, the addition of Exchange-based applications, and so on – a historical record of performance data helps to identify bottlenecks. The system component that causes the most delay in the execution of a process or a task is known as a **bottleneck**. The primary goal of system optimization is to eliminate all bottlenecks. The challenge is that, eliminating one bottleneck frequently produces another bottleneck somewhere else. For example, adding a more powerful CPU might result in a memory bottleneck. Adding more memory to compensate might then over-task your disk subsystem, and so on, and so on.

Fortunately, with proper analysis, operating system bottlenecks are generally easy to spot. For example, if the value of Processor - % Processor Time consistently exceeds an average of 80%, Microsoft recommends upgrading your processor. If your paging file is correctly sized and the value for Paging File - % Usage is consistently more than 80%, you should add memory to your system. If the number of disk I/Os per second is greater than the rated value for your disk subsystem, then you need to upgrade your disk subsystem.

Bottlenecks within Exchange are also easy to spot, but not always easy to diagnose. Microsoft has published some general guidelines for determining bottlenecks within NT and Exchange. You can find this information in [Microsoft TechNet](#) or the [Windows 2000 Resource Kit](#). The [Exchange 2000 Resource Kit](#) and the [BackOffice Resource Kit](#) also have extensive information on the Exchange performance counters that Microsoft recommends you collect.

### T U N I N G E X C H A N G E 2 0 0 0

---

In prior versions of Exchange, administrators could tune performance using Performance Optimizer utility (PerfWiz). PerfWiz stepped the administrator through a series of questions to determine how the server was configured, whether Exchange should limit its memory usage, and how many users were being served. Based on the administrator's answers, Exchange would tune itself accordingly.

Exchange 2000 does not include PerfWiz or a replacement tool. Exchange 2000 is now more self-tuning, and is better able to dynamically change its operational parameters as needed. To tune Exchange 2000 further, administrators need to perform tasks such as manually optimizing disk subsystems and modifying Registry entries.

The greatest variable in tuning Exchange is environmental variations. Every Exchange 2000 organization is different in many respects. Companies may have very similar needs but they choose solutions that can differ significantly. Given the number of choices for processors, memory, and disk configurations, you'll definitely want to do some tuning beyond what Exchange does for itself.

According to Microsoft, the following Exchange 2000 components can be optimized in one or more respects: disks, message transfer agent, SMTP services, the Web Storage System, the Extensible Storage Engine (ESE) cache and log buffers, Active Directory and the Active Directory Connector, Exchange Installable File System (ExIFS) handle cache, credentials cache, mailbox cache, DSAccess cache, DSProxy, POP3, and IMAP4 settings.

The health and performance of Exchange 2000 depends in large part on the choice of server specifications, storage hardware, and topology. These factors should be based on expected levels of usage. Further tuning can be accomplished by modifying the Registry on the Exchange server. There are three main types of tuning parameters: those that are tuned dynamically by Exchange; those that are set to optimal values; and those that can be manually tuned.

### CAPACITY PLANNING

---

Capacity planning is a little more difficult than baselining or trending. The answers aren't going to jump out at you like a dramatic change in performance will. Capacity planning is the determination of future server needs to the extent possible. The primary goal of capacity planning is to ensure that you have room for growth, and to let you know when additional resources are needed.

Capacity planning for an Exchange infrastructure involves determining current and future needs, and then selecting the hardware resources that meet estimated needs. However, it can also mean upgraded hardware to meet new needs as they arise. Because there are so many variables, and because needs change so rapidly, capacity planning typically requires an iterative approach.

The important thing to remember is that, without a baseline and some trend analysis, capacity planning is impossible. By using ELM Enterprise Manager to collect and store your servers' performance data on a regular basis, you'll be able to baseline and trend your Exchange infrastructure, and accurately determine when demand has exceeded available resources.

### PERFORMANCE ALARMS

---

Even if you choose not to collect performance data, you can use **Performance Alarms** to monitor performance and generate alert messages when a performance counter object exceeds the threshold you specify. An Alarm is a condition where a selected performance counter is less than, greater than, or equal to a specific value. Alarms, which are configured globally (e.g., all alarms apply to all monitored systems), specify what action will be taken when a performance counter meets the specified criteria. By configuring Alarms, you'll be able to take action at the first sign of trouble. You can configure Alarms to send email messages and network messages, and even execute commands, batch files and applications.

➤ To create a Performance Alarm:

1. Right-click on the **Monitor Items** container in the ELM Console and select **New | Monitor Item**. The Create Monitor Wizard will appear. Click **Next** to continue.
2. Select **Performance Alarm** and click **Next** to continue.
3. Enter a **Name** and **Description** and click **Next** to continue.
4. The Performance Alarm Watch dialog box will appear.
  - a. In the **Object** drop-down, select the Performance Object you want to monitor. If the object you want to monitor is not listed, you can add additional objects and counters by clicking the **Add** button.
  - b. In the **Counter** drop-down, select the Counter you want to monitor.
  - c. To monitor all instances of a particular counter, leave an asterisk in the Instances field. To monitor a single instance, or a specific set of instances, click the **Add/Remove** button, enter the instance(s) you want to monitor and click **Close**. Alternatively, you can enter multiple instances directly in the **Instances** field; be sure to separate each instance with a semi-colon.
  - d. In the **Condition** field, enter the condition you want to match (e.g., less than, greater than, equal to, and so forth).
  - e. Enter the value you want to match in the **Value** field.
  - f. In the **Occurs** field, enter the number of consecutive times this condition can be met before triggering notification.
5. Click **Next** to continue. The Performance Alarm Action dialog box will appear. Configure the following options for each tab, and click **Next** to continue.
  - **Enable**. When this box is checked, the Action is enabled and will execute as configured. Uncheck this box to disable the Action.
  - **Create New Alert Entry**. Selecting this checkbox to cause an Alert to be generated and posted to the Alerts container within the ELM Console.
  - **Create Application Event Log Entry**. Selecting this enables you to cause a customizable event to be logged to the Application log. Use the **Variables** button to specify any additional variables you want the event to include.
  - **Net Send Message**. Using this option enables you to send a popup message over the network. The target system must have the Messenger Service (Windows NT, Windows 2000 or Windows XP) or WinPopUp (Win9x/WinMe) running in order to receive the Net Send Message. Use the **Browse** button to browse for a target computer, or manually enter the name of the computer to which you want to send messages in the Computer Name field.

- **Run Command.** This option provides a mechanism for executing a batch file, running a command line application or launching a script (CScript or WScript). Use the Edit button to create and edit any scripts or command line parameters you want to execute.
6. The Test Monitor Item dialog box will appear enabling you to test the Performance Alarm before using it. To do this, select an Agent on from the Agent dropdown and click the **Start Test** button.
  7. Click **Next** to continue. The Agents dialog box will appear. Check the checkbox for each Service Agent to which you want to apply the Performance Alarm.
  8. Click **Next** to continue. The Schedule dialog box will appear.
    - On the Scheduled Interval tab, set the frequency at which you want to execute the Alarm. Leaving the interval at the default of 1 second will enable you to monitor the performance counter or object in real-time.
    - On the Scheduled Hours tab, configure the hours and days you want this monitor active.
  9. Click **Finish** to save the Performance Alarm.

## PRACTICAL APPLICATIONS FOR PERFORMANCE ALARMS

---

Performance Alarms are used to watch performance counter objects for pre-determined thresholds. The message queues are perfect examples of this. The SMTP NTFS Store Driver, SMTP Server, MExchangeIS Virus Scan, MExchangeIS Mailbox, MExchangeIS Public, and other queues are always going to be in flux. Still, you'll want to know immediately if these queues begin filling up without emptying out.

On a very busy Exchange server, having 20 or so messages in the SMTP Server queue at any given time might be considered normal. However, if the queue suddenly jumps to 40, 50 or even more messages, or if the number in the queue continues to grow without ever shrinking, a problem probably exists. Perhaps there's a large message in the queue causing a delay in delivery. Perhaps a mail gateway, router or other device is down. Or perhaps your system is under attack by spammers or hackers. In any event, this problem needs to be identified and corrected as soon as possible. To monitor for these conditions, simply create Alarms for the queues you want to watch.

The same principle holds true for other dynamic performance counters, such as Active Connections, Current Sessions, etc.



## COLLECTING, ANALYZING AND RESPONDING TO SYSTEM EVENTS

### *Capturing and analyzing System, Security and Application events*

Exchange Server, like many other applications, makes use of the Windows 2000 event subsystem for reporting critical and information messages regarding system events and activities. Careful monitoring of system events can help you identify – and sometimes even predict – system problems. For example, if you see a low disk space event on a heavily used file server, it is likely you're about to run out of disk space. Events can also be used to confirm problems with applications. For example, application events can provide a record of activity leading up to a catastrophic event, such as an application crashing, or the like.

### **THE WINDOWS 2000 EVENT STRUCTURE**

---

Before you can use event data to diagnose problems, it is essential to have a basic understanding of interpreting the event that is logged. Events are made up of three distinct parts: the Header, the Description, and the Data.

The **Header** contains important information such as the date and time of the event, the event type (e.g., Informational, Warning, Error, Audit Success and Audit Failure), the user and computer name, the event source, the event ID and the category.

The **Description** field contains details on the exact event that occurred. This will vary between event types and, quite frankly, the details aren't always helpful in determining the problem (although they typically will point you in the right direction).

The **Data** field contains binary data for the event. This data is typically used for advance troubleshooting purposes by Microsoft Product Support Services (PSS). Mere mortals generally aren't going to be able to make use of this data.

## INTERPRETING EVENTS

---

The three primary types of events (Informational, Warning and Error) need to be interpreted in a different manner. Informational events typically signify normal system or application operations. Informational events include service initialization or shutdown, background maintenance notification and backup success. Warning events indicate minor problems or inconsistencies that may cause a problem down the road. Error events are more critical and should be investigated upon discovery. They indicate potentially serious problems, such as a service failure or the impending shutdown of a service, the catastrophic failure of an application or hardware component, or some other urgent issue.

The remaining event types (Audit Success and Audit Failures) are security-related and only present when auditing has been enabled. These events are useful when troubleshooting authentication failures that prevent users from accessing mailboxes, public folders, Outlook Web Access, Instant Messaging or Conferencing services.

By default, Exchange only logs basic events. These include informational events such as backup and restore activity, service startup and shutdown and background maintenance. In addition, Exchange will log events such as low disk space warnings, and database errors. Generally speaking, Exchange will log a warning or error for any event that causes a degradation of, or disruption in, service.

## DIAGNOSTICS LOGGING

---

When problems arise, diagnostics logging can be an Exchange administrator's best friend. Each Exchange component can be set to varying diagnostic reporting levels. In addition, components such as the Internet Mail Service can be configured to create an additional diagnostics log for more advanced troubleshooting. In almost all cases, diagnostic logging changes are immediate and do not require you to restart either the service or the server. There are four different diagnostic logging levels that can be enabled.

Once configured, the diagnostics logging settings are stored in the Registry under:

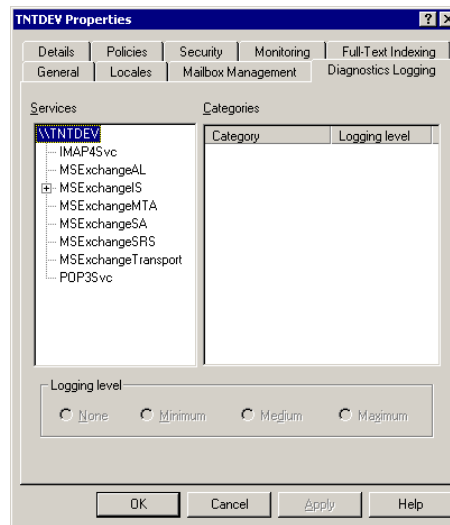
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\ServiceName

Each ServiceName will have a registry sub-key named "Diagnostics" under which the individual settings are stored. The standard logging levels including their registry settings are:

- 0 - None
- 1 - Minimum Logging
- 3 - Medium Logging
- 5 - Maximum Logging

Under normal circumstances these registry keys need never be modified manually. However, Microsoft PSS may instruct you to configure logging to a special setting of "6," which turns on advanced diagnostic logging.

Diagnostic logging can also be turned on and off through System Manager. Right-click on the Server object in Exchange System Manager and select Properties, then click the Diagnostics Logging tab, as shown below.



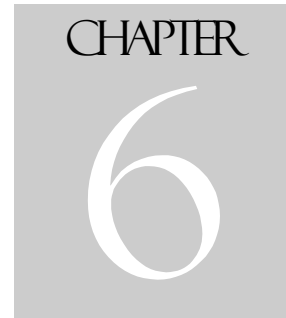
As the name implies, Diagnostic Logging is a troubleshooting tool. It should not be used during normal Exchange operating conditions, because if it is used improperly, it can cause system-wide problems (e.g., excessive disk usage and degraded system performance). Therefore, once you're done with Diagnostic Logging, it is important to change the logging level back to 'None.'

## WHAT TO LOOK FOR

---

Exchange is capable of logging thousands of different events. While some of these events can be safely ignored, others require some sort of administrative attention to either acknowledge them, or address them. The most common events will be service initialization and shutdown messages, background maintenance, backup-related events (if online backups are being taken), database free space messages, and so forth.

The easiest way to track and process events is to set up Rules within ELM Enterprise Manager that process incoming events and trigger the Notification Method(s) you specify.



## CONCLUSIONS AND SUMMARY

*ELM Enterprise Manager is the Best Solution for Total Management of your Exchange Servers*

No matter how good your management processes are, there will be problems with your systems. Problem management can be divided into three separate areas - problem detection, problem notification and problem resolution. All problem identification in an Exchange environment is ultimately based on the continuous monitoring of two data sources - the Windows 2000 event subsystem and the Windows/Exchange performance counters. Once a problem is detected, ELM Enterprise Manager provides a wide range of notification mechanisms to ensure that support personnel are appropriately alerted.

Exchange servers are expected to deliver a wide variety of functionality: simple text messaging, rich email, business forms, collaborative and discussion applications, workflow, customer communications and business-to-business e-commerce. No matter how big or small your Exchange infrastructure is, or how you use it, it requires careful and proactive monitoring.

ELM Enterprise Manager is the best solution for proactively monitoring all aspects of your Exchange infrastructure in real-time.



## EXPLANATION OF RECOMMENDED PERFORMANCE COUNTER OBJECTS

### ACTIVE DIRECTORY (NTDS)

---

**DRA Inbound Bytes Total** is the total number of bytes received from inbound replication. The value is the sum of the number of uncompressed bytes and compressed bytes (after compression).

**DRA Inbound Bytes Not Compressed** is the amount of replication data (in bytes) that was not compressed at the source. This represents inbound replication data from other directory service agents (DSAs) in the same site. This value is per second.

**DRA Inbound Bytes Not Compressed (Before)** is the uncompressed size of compressed replication data inbound from DSAs in other sites. This value is per second.

**DRA Inbound Bytes Not Compressed (After)** is the amount of compressed replication data inbound from DSAs in other sites. This value is per second.

**DRA Outbound Bytes Total** is the total number of bytes sent in outbound replication. The value is the sum of the number of uncompressed bytes and compressed bytes (after compression).

**DRA Outbound Bytes Not Compressed** is the amount of replication data (in bytes) that was not compressed at the source. This represents outbound replication data to other directory service agents (DSAs) in the same site. This value is per second.

**DRA Outbound Bytes Not Compressed (Before)** is the uncompressed size of compressed replication data outbound to DSAs in other sites. This value is per second.

**DRA Inbound Bytes Not Compressed (After)** is the amount of compressed replication data outbound to DSAs in other sites. This value is per second.

**DS Directory Searches/sec** is the number of directory searches per second.

**DS Directory Reads/sec** is the number of directory reads per second.

**DS Directory Writes/sec** is the number of directory writes per second.

**DS Threads in Use** is the current number of threads in use by the directory service (different than the number of threads in the directory service process). Threads in Use is the number of threads currently servicing client API calls. This counter is one indicator of whether additional processors could be of benefit.

**Kerberos Authentications/sec** is the number of times per second that clients use a ticket to this DC to authenticate to this DC.

**LDAP Bind Time** is the time (in milliseconds) taken for last successful LDAP bind.

**LDAP Client Sessions** is the number of connected LDAP client sessions.

**LDAP Searches/sec** is the rate at which LDAP clients perform search operations.

**NTLM Authentications** is the number of NTLM authentications per second serviced by this DC.

#### **DATABASE (ACTIVE DIRECTORY & EXCHANGE IS)**

---

**Cache % Hit** is the percentage of database file page requests that were fulfilled by the database cache without causing a file operation. You can monitor this counter for both the Directory and Information Store databases. If this percentage is too low (e.g., less than 85%), the database cache size may be too small.

**Cache Size** is the amount of system memory used by the database cache manager to hold commonly used information from the database file(s). This figure shows you how much memory is being used by the Extensible Storage Engine. If the database cache size seems to be too small for optimal performance and there is very little available memory on the system (see Memory/Available Bytes), adding more memory to the system may increase performance. If there is a lot of available memory on the system and the database cache size is not growing beyond a certain point, the database cache size may be capped at an artificially low limit.

**Log Record Stalls/sec** is the number of log records that cannot be added to the log buffers per second because they are full. If this counter is non-zero most of the time, the log buffer size may be a bottleneck.

**Log Threads Waiting** is the number of threads waiting for their data to be written to the log in order to complete an update of the database. If this number is too high, the log may be a bottleneck.

**Table Open Cache % Hit** is the percentage of database tables opened using cached schema information. If this percentage is too low (e.g., less than 75%), the table cache size may be too small.

## D N S

---

**Dynamic Update Received** is the total number of dynamic update requests received by the DNS server.

**Dynamic Update Rejected** is the total number of dynamic updates rejected by the DNS server.

**Dynamic Update TimeOuts** is the total number of dynamic update timeouts of the DNS server.

**Recursive Queries** is the total number of recursive queries received by DNS server.

**Recursive Query Failure** is the total number of recursive query failures.

**Recursive Send TimeOuts** is the total number of recursive query sending timeouts.

**Secure Update Failure** is the total number of secure updates failed of the DNS server.

**Zone Transfer Failure** is the total number of failed zone transfers of the master DNS server.

## E P O X Y

---

**Client Out Que Length** is the length of the queue from the client to the store. If this queue continues to grow and not return to zero then there may be a bottleneck between IIS and Exchange. In this case, you should closely examine Physical Disk and Logical Disk performance counters for similar bottlenecks.

## I N T E R N E T I N F O R M A T I O N S E R V I C E S G L O B A L

---

**BLOB Cache Hits** is the total number of successful lookups in the BLOB cache.

**BLOB Cache Misses** is the total number of unsuccessful lookups in the BLOB cache.

**Current File Cache Memory Usage** is current number of bytes used for file cache.

**File Cache Hits** is the total number of successful lookups in the file cache.

**File Cache Misses** is the total number of unsuccessful lookups in the file cache.

**Total Blocked Async I/O Requests** is the total number of requests since startup allowed by bandwidth throttling settings.

**URI Cache Hits** is the total number of successful lookups in the URI cache.

**URI Cache Misses** is the total number of unsuccessful lookups in the URI cache.

## LOGICAL DISK

---

**Avg. Disk Queue Length** is the average number of both read and write requests that were queued for the selected disk during the sample interval. Increases over the initial baseline typically indicate a disk bottleneck.

**Avg. Disk Sec/Transfer** is the time, in seconds, of the average disk transfer. Increases over the initial baseline typically indicate a disk bottleneck.

**Current Disk Queue Length** is the number of requests outstanding on the disk (including requests in service) at the time the performance data is collected. This is an instantaneous length, not an average over the time interval. Multi-spindle disk devices can have multiple requests active at one time, but other concurrent requests are awaiting service. This counter may reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests are experiencing delays proportional to the length of this queue minus the number of spindles on the disks. The queue length should always be less than half of the number of disks in a volume. So, if you have 8 disks in a volume, the queue length should always be less than 4.

**Disk Reads/Sec** is the rate of read operations on the disk. Most disk and disk controller manufacturers have formulas for measuring total disk I/O, which use this counter as part of that formula.

**Disk Writes/Sec** is the rate of write operations on the disk. Most disk and disk controller manufacturers have formulas for measuring total disk I/O, which also use this counter as part of that formula.

**Free Megabytes** is the amount of unallocated space on the disk in megabytes. This is a very important counter to monitor. Alarms should be configured so that alerts will be generated as the disks or disk volumes approach capacity. Exchange will self-terminate if its log files or databases have no more space to grow. For example, the MTA will terminate if there is less than 10MB available. Using Microsoft's definition, one megabyte = 1,048,576 bytes.

**% Free Space** is the ratio of the free space available on the logical disk to the total usable space on selected logical disk.

## MEMORY

---

**Available Mbytes** is the amount of physical memory available (in megabytes) to running processes. It is calculated by summing space on the Zeroed, Free, and Standby memory lists. Zeroed memory refers to pages of memory filled with zeros to prevent later processes from seeing data used by a previous process. Free memory is ready for use. Standby memory is memory removed from a process' working set on route to disk, but is still available to be recalled. This counter displays the last observed value only; it is not an average. We recommended having a minimum of 4MB always available on an Exchange Server that has less than 256MB of RAM. This number should be doubled for systems with more RAM (e.g., 8MB for less than 512MB, 16MB for less than 1GB, etc).

**Cache Faults/sec** is the number of faults that occur when a page sought in the file system cache is not found there and must be retrieved from elsewhere, either in memory (a soft fault) or from disk (a hard fault). The file system cache is an area of physical memory that stores recently used

pages of data for applications. Cache activity is generally a reliable indicator of most application I/O operations. This counter represents the number of faults, without regard for the number of pages faulted in each operation.

**Commit Limit** is the size (in bytes) of virtual memory that can be committed without having to increase the size of the paging file. If Committed Bytes approaches the Commit Limit, and the paging file cannot be extended, there are simply no more available pages in main memory or in the paging file. If this occurs you may see three errors in the System Log from source: SRV:

2001: The server was unable to perform an operation due to a shortage of available resources.

2016: The server was unable to allocate virtual memory.

2020: The server was unable to allocate from the system paged pool because the pool was empty.

This is generally related to a memory leak in another process. To determine the process at fault you can monitor each process's Page File bytes or Working Set.

**Committed Bytes** is the amount of committed virtual memory, measured in bytes. Committed memory is physical memory for which space has been reserved on the paging file in case it needs to be written back to disk. This counter must not exceed the overall size of the paging file, as this would indicate that too much application data has been committed to virtual space.

**Pages/sec** is the number of pages written to or read from disk to resolve hard page faults. This counter is meant to be the primary indicator of the kinds of faults that cause system-wide delays. It is counted in numbers of pages, so it can be compared to other counts of pages, such as Memory – Page Faults/sec, without requiring any conversion.

**Page faults/sec** is the overall rate that faulted pages are handled by the CPU. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its space in physical memory. This counter includes both hard faults and soft faults. Most CPUs can handle large numbers of soft faults without consequence. Hard faults, however, can cause significant system delays.

**Page Reads/sec** is the number of times the disk was read to resolve hard page faults. A hard page fault occurs when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It includes reads to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of read operations, without regard to the numbers of pages retrieved by each operation.

**Page Writes/sec** is the number of times pages were written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This counter counts write operations, without regard to the number of pages written in each operation.

**% Committed Bytes in Use** is the ratio of the Committed Bytes to the Commit Limit. This is an instantaneous value, not an average, which represents the amount of available virtual memory in use. Note that the Commit Limit may change if the paging file is extended.

**Pool Nonpaged Bytes** is the number of bytes in the Nonpaged Pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks.

Nonpaged Pool pages cannot be paged out to the paging file, but instead remain in main memory as long as they are allocated. A slow rise in the value for this counter could indicate a memory leak.

## **M I C R O S O F T   E X C H A N G E   D S   A C C E S S   C A C H E**

---

**Cache Expiries Total** is the total number of objects expired from the cache since system initialization.

**Cache Hits/sec** is the number of "object found in cache" events per second.

**Cache Misses/sec** is the number of "object not found in cache" events per second.

**Cache Hits Total** is the total number of "object found in cache" events since system initialization.

**Cache Inserts Total** is the total number of objects inserted into the cache since system initialization.

**Cache Inserts/Sec** is the number of objects inserted into the cache per second.

**Cache Misses Total** is the total number of "object not found in cache" events since system initialization.

**Cache Misses/Sec** is the number of "object not found in cache" events per second.

**Cache Timeout** is the number of seconds a DN Entry can stay in the cache before being expired.

**DN Entries** is the total number of Distinguished Name objects in the cache.

**DN Entries Memory** is the memory occupied (in bytes) by all DN objects in the cache.

**LDAP Searches Total** is the total number of LDAP search requests issued since system initialization.

**LDAP Searches/Sec** is the number of LDAP search requests issued per second.

**LRU Entries** is the number of cached object entries on the LRU chain.

**Max Entries** is the maximum number of entries allowed in the cache.

**Max Memory Size** is the maximum allowed memory size (in bytes) of the cache.

**Not Found DN Entries** is the total number of "Not Found DN" type entries in the cache.

**Not Found DN Entries Memory** is the memory occupied (in bytes) by all "Not Found DN" Entries in the cache.

**Not Found GUID Entries** is the total number of "Not Found GUID" type entries in the cache.

**Not Found GUID Entries Memory** is the memory occupied (in bytes) by all "Not Found GUID" Entries in the cache.

**Outstanding Async Notices** is the number of outstanding LDAP notification requests.

**Outstanding Async Reads** is the number of outstanding LDAP read requests.

**Outstanding Async Searches** is the number of outstanding LDAP search requests.

**Search Entries** is the total number of Search type entries in the cache.

**Search Entries Memory** is the memory occupied (in bytes) by all Search Entries in the cache.

**Total Entries** is the total number of entries in the cache, including all DN, Search, Not Found DN and Not Found GUID type entries.

**Total Entries Memory** is the memory occupied (in bytes) by all entry objects in the cache excluding the cache tables overhead.

**Total Memory Size** is the total shared memory size of the cache (in bytes) including cache Entry objects and the cache table data structures.

## M I C R O S O F T   E X C H A N G E   I N F O R M A T I O N   S T O R E

---

**RPC Operations/sec** is the rate at which RPC operations occur. If this number is substantially higher than baseline values, it could indicate excessive or unauthorized use of your Exchange Server.

**User Count** is the number of users connected to the Information Store. This is the actual count of people (not connections) that are currently using the IS.

**VM Total 16MB Free Blocks** is the total number of free Virtual Memory blocks larger than or equal to 16MB.

## M I C R O S O F T   E X C H A N G E   I S   M A I L B O X / P U B L I C

---

**Average Delivery Time** is the average time between the submission of a message to the store and submission to other storage providers for the last 10 messages.

**Average Local Delivery Time** is the average time between the submission of a message to the store and the delivery to all local recipients (recipients on the same server) for the last 10 messages.

**Folder Opens/Sec** is another good indicator of user activity. This counter represents the rate at which requests to open folders are submitted to the store.

**Local Delivery Rate** is the rate at which messages are delivered locally. Under normal conditions this queue should not stay above zero for very long.

**Message Opens/Sec** is the rate that requests to open messages are submitted to the store. This counter helps in forming an overall picture of user activity.

**Messages Submitted** is the total number of messages submitted by clients since service startup. This should be compared against both baseline and trend data to detect spikes of activity.

**Receive Queue Size** is the number of messages in the store's receive queue. As with the Send Queue Size, this should not remain above zero for very long during normal operating conditions.

**Send Queue Size** is the number of messages in the store's send queue. Under normal conditions this queue should not stay above zero for very long.

**Single Instance Ratio** is the average number of references to each message in the store.

---

#### M I C R O S O F T   E X C H A N G E   M T A   ( M S E X C H A N G E M T A )

---

**Messages/sec** is the count of messages per second that the MTA is processing inbound and outbound. This counter can be used with the Work Queue Length counter to give an indication of how long messages are queued before delivery. By dividing the average Work Queue Length by the average Messages/sec, you can calculate how long a message will remain in the queue before being delivered.

**Work Queue Length** is the queue length for the entire MTA. It includes both inbound and outbound messages for the Information Store, the Directory and any connectors that route through the MTA. This single counter can be used to determine the overall health of the MTA. If this counter is above zero for a sustained amount of time it may indicate a communications problem with one of the Exchange components, a connector or a remote Exchange MTA.

---

#### M I C R O S O F T   E X C H A N G E   M T A   C O N N E C T I O N S

---

**Queue Length** is the number of outstanding messages queued for transfer to the entity. This counter takes the MTA Work Queue Length counter to a lower level and breaks it out into all the different work queues within the MTA. If a large queue is building in the MTA, this counter can be used to identify the responsible connection.

---

#### M I C R O S O F T   G A T H E R E R

---

**Documents Filtered** is the number of times a filter object was created. This value corresponds to the total number of documents filtered in the system since startup.

**Performance Level** shows you the amount of system resources that the Gatherer service is allowed to use. This value corresponds to the value set in Exchange System Manager:

- 1 – Minimum
- 2 – Low
- 3 – High
- 4 – Maximum

**System IO Traffic Rate** is the System I/O (disk) traffic rate in KB/s detected by back-off logic.

**Reason to back off** is a code describing why gathering service went into back off state. This code can have the following values:

- 0 - Up and running
- 1 - High IO rate
- 4 - Back off on user activity (by default this is disabled in server install)
- 5 - Battery low (currently, if running on battery, not on AC power)
- 6 - Memory low (less than 5 MB left in paging file)

## MICROSOFT GATHERER PROJECTS

---

**Crawl in Progress Flag** indicates if a crawl is currently in progress. A value of 0 means a crawl is running; a value of 1 means it is not.

**Current Crawl is Incremental** indicates if the current crawl is an incremental crawl. A value of 0 means it is an incremental crawl. A value of 1 means it is a full crawl.

**Gatherer Pause Flag** indicates whether the Gather has been paused. 0 means not paused and 1 means paused.

**URLs in History** is the number of files (URLs) in the history list. This is the number of files known to full-text indexing.

**Waiting Documents** indicates the number of documents waiting to be crawled. This value increases at the start of a crawl as new URLs are identified, then decreases as the crawl progresses.

## MICROSOFT SEARCH

---

**Active Threads** is the total number of threads currently servicing queries.

**Failed Queries** is the number of queries that have failed.

**Succeeded Queries** is the number of queries that have succeeded.

**Threads** is the number of threads available for servicing queries.

## MICROSOFT SEARCH CATALOGS

---

**Queries** is the cumulative number of queries posted to the catalog.

**Successful Queries** is the number of queries that produce successful searches.

**Failed Queries** is the number of queries that have failed.

**Results** is the cumulative number of results returned to clients.

## MICROSOFT SEARCH INDEXER CATALOGS

---

**Merge Process** is the percentage of merge complete for the current merge.

## NETWORK INTERFACE

---

**Bytes Total/sec** is the rate that bytes are sent and received on the interface, including framing characters. You can use this counter to monitor the overall network traffic to that Exchange server and compare it to the maximum available bandwidth for that segment. If the sum is close to the maximum available bandwidth, you will know that segment is reaching its maximum capacity. In this event, you should add a new segment to the network.

**Packets Outbound Errors** is the number of outbound packets that could not be transmitted because of errors. Some errors are to be expected, but if you see a large number of errors over your baseline, you probably need to replace your network cable or network adapter.

**Packets Received Errors** is the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. Some errors are to be expected, but if you see a large number of errors over your baseline, you probably need to replace your network cable or network adapter.

## PAGING FILE

---

**% Usage** is the amount of the paging file instance in use. This indicates how much of the page file is in use, which is in turn used to determine if a memory bottleneck exists.

**% Usage Peak** is the peak usage of the selected paging file instance given as a percentage.

## PHYSICAL DISK

---

**Avg. Disk sec/Read** is the average time in seconds of a read of data from the disk.

**Avg. Disk sec/Write** is the average time in seconds of a write of data to the disk.

**Avg. Disk sec/Transfer** is the time in seconds of the average disk transfer.

**Current Disk Queue Length** is the number of requests outstanding on the disk at the time the performance data is collected. It includes requests in service at the time of the snapshot. This is an instantaneous length, not an average over the time interval. Multi-spindle disk devices can have multiple requests active at one time, but other concurrent requests are awaiting service. This counter might reflect a transitory high or low queue length, but if there is a sustained load on the disk drive, it is likely that this will be consistently high. Requests are experiencing delays

proportional to the length of this queue minus the number of spindles on the disks. This value for this counter should generally be less than 2. If it is higher than 2 for a sustained period of time, your disks or disk subsystem may be a bottleneck.

## PROCESS

---

**Handle Count** is the total number of handles that are currently open by the process. This number is the sum of the handles currently open by each thread in this process.

**Page faults/sec** is the rate at which page faults occur in the threads executing in this process. A page fault occurs when a thread refers to a virtual memory page that is not in its working set. This will not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with which the page is shared.

**Page File Bytes** is the current number of bytes this process has used in the paging file(s). Because paging files are shared by all processes, the lack of available space within the paging file can prevent other processes from allocating memory. This counter can be used to isolate memory leak in a process.

**% Processor Time** is the percentage of elapsed processor time used by all threads of the selected process. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count. On SMP machines, the maximum value of the counter is 100% times the number of processors.

**Private Bytes** is the current number of bytes this process has allocated that cannot be shared with other processes.

**Thread Count** is the number of threads that are currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.

**Virtual Bytes** is the current size of the virtual address space (in bytes) that the process is using. This counter is critical in determining if the store cache size is large enough. If your Exchange 2000 server has the /3GB switch set in Boot.ini, the Virtual Bytes value for the STORE.EXE process should be below 2.8 GB. If your server does not use the /3GB switch, the value should be below 1.8 GB. Microsoft recommends using the /3GB switch for Exchange 2000 servers that have 1 GB of memory or more. If you see values that are higher than those above for either configuration, do not increase the size of your maximum cache size. If you see values that are lower for either configuration, you can safely increase the size of your database maximum cache size.

**Working Set** is the current amount of RAM used by this process and its data. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they will then be soft-faulted back into the Working Set before they leave main memory. See Microsoft Knowledge Base article [Q184699](#) for more information on this counter.

## PROCESSOR

---

**Interrupts/sec** is the number of device interrupts the processor experiences. A device interrupts the processor when it has completed a task or when it otherwise requires attention. Normal thread execution is suspended during interrupts. An interrupt may cause the processor to switch to another, higher priority thread. Clock interrupts are frequent and periodic and create a background of interrupt activity. This counter needs to be compared with baselines taken during normal operations to determine if the present value indicates a problem of some sort.

**% Processor Time** is the percentage of time the processor is executing a non-idle thread. This counter is the primary indicator of processor activity. It is calculated by measuring the time the processor spends executing the thread of the Idle process in each sample interval, and subtracting that value from 100%. Each processor has an Idle thread which consumes cycles when no other threads are ready to run. If this counter is consistently above 75%, you have a CPU bottleneck.

## REDIRECTOR

---

**Bytes Total/sec** is the rate at which the Redirector processes data bytes. This includes all application and file data, and protocol information such as packet headers.

**Current Commands** is the number of requests to the Redirector that are currently queued for service. If this number is much larger than the number of network adapter cards installed in the Exchange Server, then the network and/or the Exchange Server are seriously bottlenecked.

**Network Errors/sec** is the count of network errors encountered by the Redirector. A few errors are to be expected, so this counter should be continuously monitored and compared to baselines recorded during normal operating conditions. Higher than normal values generally mean that the Redirector is having serious communication problems. Some of these errors will result in events being generated, which provides additional troubleshooting information.

**Reads Denied/sec** is the rate at which the server is unable to accommodate requests for Raw Reads. When a read is much larger than the server's negotiated buffer size, the Redirector requests a Raw Read which, if granted, permits the transfer of the data without lots of protocol overhead on each packet. To accomplish this the server must lock out other requests, so the request is denied if the server is really busy.

**Writes Denied/sec** is the rate at which the server is unable to accommodate requests for Raw Writes. When a write is much larger than the server's negotiated buffer size, the Redirector requests a Raw Write which, if granted, permits the transfer of the data without lots of protocol overhead on each packet. To accomplish this, the server must lock out other requests, so the request is denied if the server is really busy.

## SERVER

---

**Bytes Total/sec** is the number of bytes the server has sent to and received from the network. This value provides an overall indication of how busy the server is. If this counter is close to the maximum transfer rate of your network, you may need to segment your network.

**Errors Access Permissions** is the number of times opens on behalf of clients have failed with Access Denied. A high number could indicate that somebody is randomly attempting to access files in hopes of getting at something that was not properly protected. If you have auditing enabled, each failed attempt will generate a Failure Audit event. A network sniffer, such as Network Monitor, can be used to identify the source of these attempts.

**Errors Granted Access** is the number of times accesses to files opened successfully were denied. This could also indicate attempts to access files without proper access authorization. If you have auditing enabled, each failed attempt will generate a Failure Audit event. A network sniffer would help here, too.

**Errors Logon** is the number of failed logon attempts to the server. This could indicate that a password cracking program is being used to hack into the server. If you have auditing enabled, each failed attempt will generate a Failure Audit event. A network sniffer would help here, too.

**Errors System** is the number of times an internal Server Error was detected. Unexpected errors usually indicate a problem with the server. Check to see whether the server is running out of memory and check the collected system events to see if you have a hardware problem. If neither is indicated, you might consider contacting Microsoft Product Support Services.

**Pool Nonpaged Bytes** is the number of bytes of non-pageable memory the server is currently using. A slow rise in the value of Pool Nonpaged Bytes could indicate a memory leak. You should also make sure the Server service is set to "Maximize Throughput for Network Applications."

**Pool Nonpaged Failures** is the number of times allocations from nonpaged pool have failed. Any value above zero could indicate that more memory is needed.

**Pool Nonpaged Peak** is the maximum number of bytes of nonpaged pool the server has had in use at any one point. This counter provides a good indication of how much physical memory the server should have.

**Pool Paged Bytes** is the number of bytes of pageable memory the server is currently using.

**Pool Paged Failures** is the number of times allocations from paged pool have failed. This could indicate that the server doesn't have enough physical memory or that the paging file is too small.

**Pool Paged Peak** is the maximum number of bytes of paged pool the server has had allocated. This counter can be used to determine the proper sizes of the paging file and proper amounts of physical memory.

**Server Sessions** is the number of sessions currently active in the server.

**Sessions Errored Out** is the number of sessions that have been closed due to unexpected error conditions. This counter indicates how frequently network problems are causing dropped sessions on the server.

**Sessions Timed Out** is the number of sessions that have been closed due to their idle time exceeding the autodisconnect parameter for the server. This counter will indicate whether or not the autodisconnect setting is helping to conserve resources.

**Work Item Shortages** is the number of times STATUS\_DATA\_NOT\_ACCEPTED was returned at receive indication time. This occurs when no work item is available or can be allocated to service the incoming request. This indicates whether the InitWorkItems or MaxWorkItems parameters need to be adjusted.

**S M T P S E R V E R**

---

**Badmailed Messages (Bad Pickup File)** is the number of malformed pickup messages sent to badmail.

**Badmailed Messages (General Failure)** is the number of messages sent to badmail for reasons not associated with a specific counter.

**Badmailed Messages (Hop Count Exceeded)** is the number of messages sent to badmail because they had exceeded the maximum hop count.

**Badmail Messages (NDR of DSN)** is the number of Delivery Status Notifications sent to badmail because they could not be delivered.

**Badmail Messages (No Recipients)** is the number of messages sent to badmail because they had no recipients.

**Badmail Messages (Triggered via Event)** is the number of messages sent to badmail at the request of a server event sink.

**Bytes Received Total** is the total number of bytes received by the SMTP server.

**Bytes Received/sec** is the rate at which bytes are received.

**Bytes Sent Total** is the total number of bytes sent by the SMTP server.

**Bytes Sent/sec** is the rate at which bytes are sent.

**Cat: Address lookups not found** is the number of address lookups that did not find any directory object.

**Cat: Categorizations Failed (DS Connection Failure)** is the number of categorizations that failed due to a directory server connection failure.

**Cat: Categorizations Failed (DS Logon Failure)** is the number of categorizations that failed due to a directory server logon failure.

**Cat: Categorizations Failed (Non-Retryable Error)** is the number of categorizations that failed with a hard error and could not be retried.

**Cat: Categorizations Failed (Out of Memory)** is the number of categorizations that failed due to lack of available memory.

**Cat: Categorizations Failed (Retryable Error)** is the number of categorizations that failed with an error but could be retried.

**Cat: Categorizations Failed (Sink Retryable Error)** is the number of categorizations that failed with a generic error that could be retried.

**Cat: LDAP Bind Failures** is total number of LDAP bind failures.

**Cat: LDAP Connection Failures** is the total number of failures encountered connection to LDAP servers.

**Cat: LDAP Paged Search Failures** is the number of failures to dispatch an async paged LDAP search.

**Cat: LDAP Search Failures** is the number of failures to dispatch an async LDAP search.

**Cat: LDAP Searches Pending Completion** is the number of LDAP searches pending async completion.

**Cat: Mailmsg Duplicate Collisions** is the number of times a duplicate recipient address was detected by mailmsg/categorizer.

**Cat: Recipients NDRd (Ambiguous Address)** is the number of recipients with addresses that match multiple directory objects.

**Cat: Recipients NDRd (Forwarding Loop)** is the number of recipients NDRd by the categorizer due to a forwarding loop detection.

**Cat: Recipients NDRd (Illegal Address)** is the number of recipients with illegal addresses detected by the categorizer.

**Cat: Recipients NDRd (Sink Recip Errors)** is the number of recipients NDRd by the categorizer due to a generic recipient failure.

**Cat: Recipients NDRd (Unresolved)** is the number of unresolved recipients (local addresses not found).

**Cat: Recipients NDRd by Categorizer** is the number of recipients set to be NDRd by the categorizer.

**Cat: Senders Unresolved** is the number of senders not found in the directory.

**Inbound Connections Current** is the total number of current inbound connections.

**Inbound Connections Total** is the total number of inbound connections received since the SMTP server was started.

**Local Queue Length** is the number of messages in the queue.

**Local Retry Queue Length** is the number of messages in the retry queue.

**Messages Refused for Address Object** is the total number of messages refused due to lack of address objects.

**Messages Refused for Mail Objects** is the total number of messages refused due to lack of mail objects.

**Messages Refused for Size** is the number of messages rejected because they were too big.

**Messages Received Total** is the total number of inbound messages accepted.

**Messages Sent Total** is the number of outbound messages sent.

**NDRs Generated** is the number of non-delivery reports that have been generated.

**Outbound Connections Current** is the number of current outbound connections.

**Outbound Connections Refused** is the number of outbound connection attempts refused by remote sites.

**Outbound Connections Total** is the total number of outbound connections attempted.

**Remote Queue Length** is the number of messages in the remote queue.

**Remote Retry Queue Length** is the number of messages in the retry queue for remote delivery.

**Total Connection Errors** is the total number of connection errors that have occurred since the SMTP server was started.

**Total DSN Failures** is the total number of failed DSN generation attempts.

**Total Messages Submitted** is the total messages submitted to queuing for delivery.

#### S M T P N T F S S T O R E D R I V E R

---

**Messages in the Queue Directory** is the current number of messages in the queue directory. If this value continues to grow without decreasing, then items in the queue are not leaving the queue, or not leaving the quicker than they are entering the queue.

#### S Y S T E M

---

**% Registry Quota in Use** indicates the percentage of the Total Registry Quota Allowed currently in use by the system. This is especially important to monitor if your Exchange server is also a domain controller, because user accounts, system policies, and related information can cause a registry quota to become exhausted, especially on large networks. If this value begins to approach 100%, you should increase the total registry size. If this happens and your Exchange server is not a domain controller, then you should examine the Registry to determine why it has grown so large.

**Processor Queue Length** is the instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. If this counter is greater than two, something is causing congestion. This could also indicate a processor bottleneck.

**System Calls/Sec** is the frequency of calls to system service routines. These routines perform all of the basic scheduling and synchronization of activities on the computer, and provide access to non-graphical devices, memory management, and name space management. If there are many more processor interrupts per second than system calls, it could indicate that a hardware device is generating an excessive number of interrupts.

**System Up Time** is the elapsed time (in seconds) that the computer has been running since it was last started. This counter displays the difference between the start time and the current time.