Tivoli

TME 10 Distributed Monitoring Collection Reference

September 1998

TME 10 Distributed Monitoring Collection Reference (September 1998)

Copyright Notice

Trademarks

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to Tivoli Systems’ or IBM’s valid intellectual property or other legally protectable right, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user.

Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594.

Contents

Chapter 1—TME 10 Distributed Monitoring Collection Installation

Chapter 2—ARM Monitoring Collection

Chapter 3—Compaq Insight Manager Monitoring Collection Introduction

Chapter 4—Compaq Insight Manager Monitoring Collection Sources

Chapter 7—OS/2 Monitoring Collection

Chapter 8—SNMP Monitoring Collection Introduction

Chapter 9—SNMP Compaq Insight Manager Monitoring Sources

Chapter 10—SNMP MIB-II Monitoring Sources

Chapter 11—User-Defined SNMP Monitoring Sources

Chapter 12—TME 10 Monitoring Sources

Chapter 13—TME 10 Enterprise Console Monitoring Sources

Chapter 14—UNIX Monitoring Collection

Chapter 15—Universal Monitoring Collection

Appendix A—TME 10 Distributed Monitoring Operator Groups

Preface

The *TME 10 Distributed Monitoring Collection Reference* describes how to install and use the monitoring collections for TME 10 Distributed Monitoring.

Note: This release of TME 10 Software Distribution marks the introduction of this product to the TME 10 product line. As products join TME 10, the product names are changing. See the *TME 10 Software Distribution Release Notes* for a list of old and new product names.

Who Should Read This Guide

This guide explains concepts you should know to effectively use TME 10 Distributed Monitoring. Readers of this guide should have a knowledge of TME 10 and TME 10 Distributed Monitoring.

Prerequisite and Related Documents

The *TME 10 Distributed Monitoring User's Guide* explains how to set up and use the TME 10 Distributed Monitoring application. You must be familiar with TME 10 Distributed Monitoring before you can install a monitoring collection.

The *TME 10 Framework User's Guide* contains more detailed information about profile management, including profile managers, profile databases, and profiles.

What This Guide Contains

The *TME 10 Distributed Monitoring Collection Reference* contains the following sections:

- Chapter 1, “TME 10 Distributed Monitoring Collection Installation”
- Chapter 2, “ARM Monitoring Collection”
- Chapter 3, “Compaq Insight Manager Monitoring Collection Introduction”
- Chapter 4, “Compaq Insight Manager Monitoring Collection Sources”

- Chapter 5, “Netware Monitoring Collection”
- Chapter 6, “NT Monitoring Collections”
- Chapter 7, “OS/2 Monitoring Collection”
- Chapter 8, “SNMP Monitoring Collection Introduction”
- Chapter 9, “SNMP Compaq Insight Manager Monitoring Sources”
- Chapter 10, “SNMP MIB-II Monitoring Sources”
- Chapter 11, “User-Defined SNMP Monitoring Sources”
- Chapter 12, “TME 10 Monitoring Sources”
- Chapter 13, “TME 10 Enterprise Console Monitoring Sources”
- Chapter 14, “UNIX Monitoring Collection”
- Chapter 15, “Universal Monitoring Collection”
- Appendix A, “TME 10 Distributed Monitoring Operator Groups”

Conventions Used in This Guide

The guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold Commands, keywords, file names, or other information that you must use literally appear in **bold**. Names of windows, dialogs, and other controls also appear in **bold**.

Italics Variables and values that you must provide appear in *italics*.

Bold Italics New terms appear in ***bold italics*** when they are defined in the text.

`Monospace` Code examples appear in a `monospace` font.

Many procedures in this guide include icons in the left margin. These icons provide context for performing a step within a procedure. For example, if you start a procedure by double-clicking on a policy region icon, that icon appears in the left margin next to the first step. If the fourth step of the procedure instructs you to open another icon, that icon appears in the left margin next to the fourth step.

Platform-Specific Information

The following table identifies text used to indicate platform-specific information or procedures:

Text	Supported Platform
AIX 4.*x*	IBM RS/6000 series running AIX 4.1 or 4.2
HP-UX 10.*x*	HP9000/700 and 800 series running HP-UX 10.01 or HP-UX 10.10
NetWare	IBM-compatible PCs 486 or higher running Novell NetWare 3.11, 3.12, or 4.0
Solaris	Sun SPARC series running Solaris 2.3,2.4, 2.5, or 2.5.1
SunOS	Sun SPARC series running SunOS 4.1.2 or 4.1.3
Windows NT	Client or server: IBM-compatible PCs 486 or higher running Microsoft Windows NT 3.51 or 4.0 PC agent: IBM-compatible PCs 486 or higher running Microsoft Windows NT 3.1 or 3.5

Contacting Customer Support

If you encounter difficulties with any Tivoli products, you can enter http://www.support.tivoli.com to view the Tivoli Support home page. After you link to and submit the customer registration form, you will be able to access many customer support services on the Web.

Use the following phone numbers to contact customer support in the United States: the Tivoli number is 1-800-848-6548 (1-800-TIVOLI8) and the IBM number is 1-800-235-5511 (press or say 8 after you reach this number). Both of these numbers direct your call to the Tivoli Customer Support Call Center.

We are very interested in hearing from you about your experience with

Tivoli products and documentation. We welcome your suggestions for improvements. If you have comments or suggestions about this documentation, please send e-mail to pubs@tivoli.com.

1

TME 10 Distributed Monitoring Collection Installation

This chapter provides the information you need to install a monitoring collection for TME 10 Distributed Monitoring in TME 10.

Before attempting to install this application, make certain you review the following sections.

- Software Requirements on page 1-2
- Hardware Requirements on page 1-2

Installing with TME 10 Software Installation Service

TME 10 Software Installation Service (SIS) is a new product that can install multiple TME 10 products on multiple systems in parallel. This Java-based product can, therefore, install more products on more systems in much less time than the Framework's install facility. SIS performs product prerequisite checks and, if defined, user-specified prerequisite checks, ensuring as few install failures as possible. In most cases, failures now occur only when machines are turned off or removed from the network.

SIS also creates an ***install repository*** (IR) into which you can import the installation image of one or more TME 10 products. You can import only those interpreter types needed in your environment, which saves you disk space and import time. The IR is then the source of all your TME 10 installations. You can even share a single IR across multiple

TMRs.

Tivoli recommends you upgrade the Framework install facility in your current TME 10 installation by installing SIS. If you are installing TME 10 for the first time, install SIS on the first managed node running an SIS-supported operating system. Once installed, you should use SIS to install other TME 10 products.

See the *TME 10 Software Installation Service User's Guide* for instructions on how to install SIS in your TME 10 installation and how to install products using SIS.

Software Requirements

You must have previously installed TME 10 Framework 3.6 and TME 10 Distributed Monitoring 3.6. For information on installing TME 10 Framework 3.6, see the *TME 10 Framework User's Guide*.

Note: You must install the **bos.acct** package as part of the AIX operating system in order to use the **Page Outs** monitor in the UNIX-NT monitoring collection. The **vmstat** command, which is used by the **Page Outs** monitor, is part of the **bos.acct** package.

Hardware Requirements

Refer to the *TME 10 Distributed Monitoring 3.6 Release Notes* for client and server disk space requirements for the monitoring collections that you will be installing.

Installing a Monitoring Collection

You can install a monitoring collection from the desktop or command line.

The following table provides the context and authorization role required for this task.

Activity	Context	Required Role
Install a monitor collection	TME	**senior or product_install**

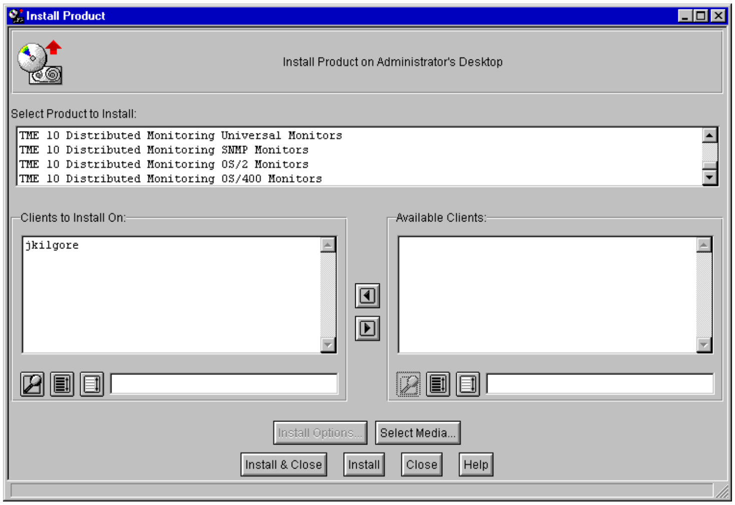
Note: If you are installing the NetWare monitoring collection, you must install it on the LCF gateway rather than on the managed node.

Desktop

Use the following steps to install the monitoring collection from the TME 10 desktop. You must have the **senior** or **product_install** authorization role to install this application.

Note: You can use TME 10 Distributed Monitoring to monitor interconnected TMRs. You must install monitoring collections in each TMR in which the monitoring collections will be use

1. Select the **Install** -> **Install Product...** option from the TME 10 **Desktop** menu. TME 10 displays the **Install Product** dialog:

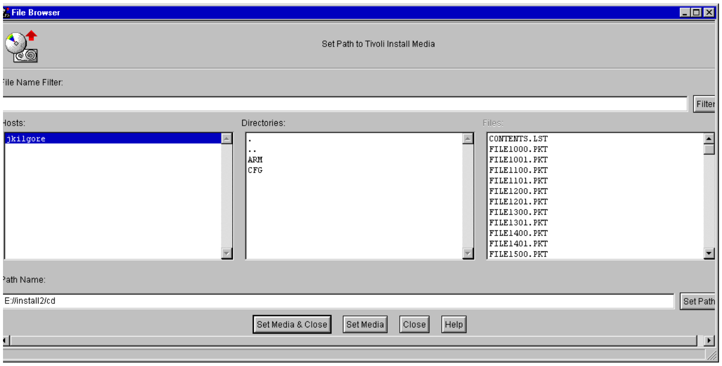


```
TME 10 Distributed Monitoring Universal Monitors
TME 10 Distributed Monitoring SNMP Monitors
TME 10 Distributed Monitoring OS/2 Monitors
TME 10 Distributed Monitoring OS/400 Monitors
```

jkilgore

If the desired monitoring collection is not listed in the **Select Product to Install** scrolling list, proceed to 2. If the monitoring collection that you wish to install is listed in the scrolling list, skip to step 3 on page 1-5.

2. Press the **Select Media...** button to display the **File Browser** dialog.



ile Name Filter:

osts:
jkilgore

Directories:
.
..
ARM
CFG

Files:
CONTENTS.LST
FILE1000.PKT
FILE1001.PKT
FILE1100.PKT
FILE1101.PKT
FILE1200.PKT
FILE1201.PKT
FILE1300.PKT
FILE1301.PKT
FILE1400.PKT
FILE1401.PKT
FILE1500.PKT

ath Name:
E://install2/cd

The **File Browser** dialog enables you to identify and specify the path to the installation media.

If you already know the path to the CD-ROM image:

a. Enter the full path in the **Path Name** field.

b. Press the **Set Path** button to change to the specified directory.

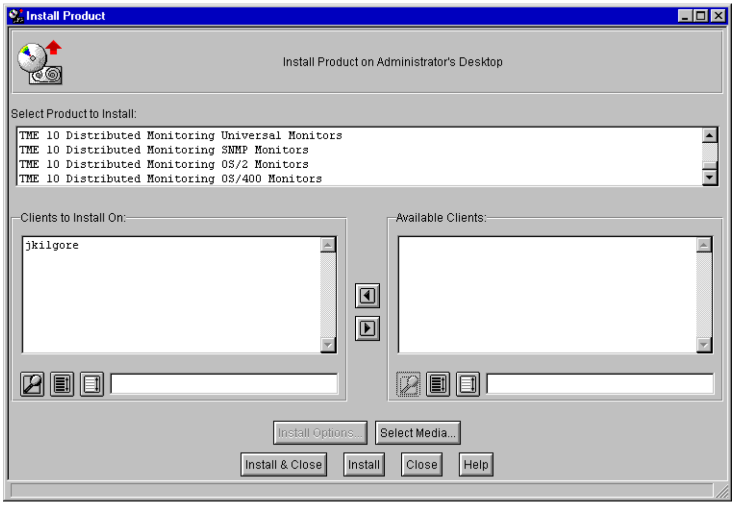
c. Press the **Set Media & Close** button to save the new media path and return to the **Install Product** dialog. The dialog now contains a list of products that are available for installation.

If you do not know the exact path to the CD-ROM image:

d. From the **Host** scrolling list, choose the host on which the install media is mounted.

e. Choose a file from the **Files** scrolling list.

f. Press the **Set Media & Close** button to save the new media path and return to the **Install Product** dialog.



Install Product on Administrator's Desktop

Select Product to Install:

```
TME 10 Distributed Monitoring Universal Monitors
TME 10 Distributed Monitoring SNMP Monitors
TME 10 Distributed Monitoring OS/2 Monitors
TME 10 Distributed Monitoring OS/400 Monitors
```

Clients to Install On:

```
jkilgore
```

Available Clients:

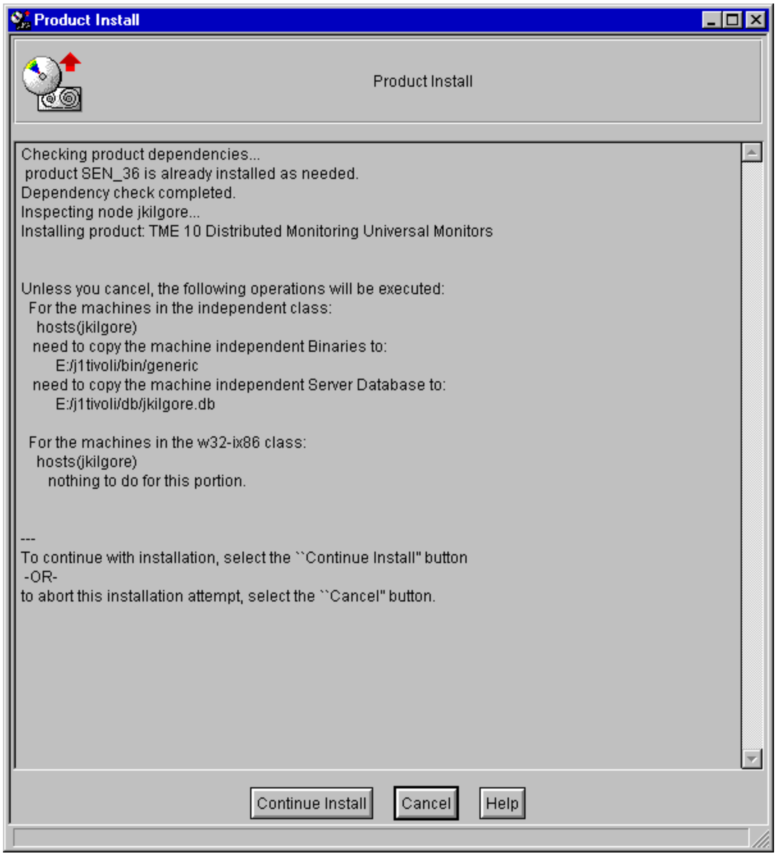
The **Product Install** dialog now contains a list of monitoring collections that are available for installation.

3. Select the desired monitoring collection from the **Select Product to Install** list.

4. Use the arrow buttons to move the clients from one choice list to another. The application will be installed on the clients in the **Clients to Install On** list.

   Note: Monitoring collections are only installed on a TMR. If you specify a managed node in the **Clients to Install On** list, the monitoring collection will not be installed on the managed node.

5. Press the **Install & Close** button to install the monitoring collection and close the **Install Product** dialog. The installation process prompts you with a **Product Install** dialog similar to the following:



```
Checking product dependencies...
 product SEN_36 is already installed as needed.
Dependency check completed.
Inspecting node jkilgore...
Installing product: TME 10 Distributed Monitoring Universal Monitors


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts(jkilgore)
   need to copy the machine independent Binaries to:
        E:/j1tivoli/bin/generic
   need to copy the machine independent Server Database to:
        E:/j1tivoli/db/jkilgore.db

  For the machines in the w32-ix86 class:
    hosts(jkilgore)
       nothing to do for this portion.


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

This dialog provides the list of operations that will take place when installing the software. This dialog also warns you of any problems that you may want to correct before you install the monitoring collection.

6. Press the **Continue Install** button to install the selected monitoring collection. This dialog returns status information as the installation process proceeds. When the installation is the complete, the **Product Install** dialog displays a completion message.
7. Press the **Close** button to close the **Product Install** dialog.

Command Line

The following example command installs the TME 10 Compaq Insight Manager monitoring collection:

```
winstall -c /cdrom/tivoli -s graceland -i COMPAQI
```

where:

-c /cdrom/tivoli Specifies the path to the CD-ROM image.

graceland Specifies that managed node **graceland** is the installation server.

-i COMPAQI Specifies the index file from which this monitoring collection is installed. To install a monitoring collection other than the TME 10 Compaq Insight Manager monitoring collection, substitute the desired monitoring collection.

Note: The above example installs the TME 10 Compaq Insight Manager monitoring collection from the command line. If you install the TME 10 SNMP monitoring collection from the TME 10 desktop, the TME 10 SNMP monitoring collection, the TME 10 Compaq Insight Manager monitoring collection, and the TME 10 SNMP MIB-II monitoring collection are all installed as part of the TME 10 SNMP monitoring collection installation process.

See the **winstall** command in the *TME 10 Framework Reference Manual* for more information.

2

ARM Monitoring Collection

The ARM monitoring collection enables you to use TME 10 Distributed Monitoring to collect data from the ARM Agents.

ARM Monitoring Collection

The following table lists the monitoring sources for the ARM Agents. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use the **warmcmd getdata** command (see the *TME 10 Distributed Monitoring ARM Agents User's Guide*).

GUI Name	CLI Name
Transaction Count	TxCount
Failed Transactions	TxFailed
Aborted Transactions	TxAbort
Average Transaction Response Time	AvgTxRT
Transaction RT Standard Deviation	StdDevTxRT
Minimum Transaction Response Time	MinTxRT

GUI Name	CLI Name
Maximum Transaction Response Time	MaxTxRT
Bucket Count # 1	TxBucket1
Bucket Count # 2	TxBucket2
Bucket Count # 3	TxBucket3
Bucket Count # 4	TxBucket4
Bucket Count # 5	TxBucket5
Bucket Count # 6	TxBucket6
Bucket Count # 7	TxBucket7
Hung Transaction	Not applicable
Transaction Too Long	Not applicable
Failed Transaction	Not applicable
Aborted Transaction	Not applicable

Transaction Count

This monitor returns the number of times the transaction completed successfully during the last collection interval.

Failed Transactions

This monitor returns the number of times the transaction failed during the last collection interval.

Aborted Transactions

This monitor returns the number of times the transaction aborted during the last collection interval.

Average Transaction Response Time

This monitor returns the average response time (in seconds) of the transaction during the last collection interval.

Transaction RT Standard Deviation

This monitor returns the standard deviation (in seconds) of the transaction response time, over the last collection interval.

Minimum Transaction Response Time

This monitor returns the minimum response time (in seconds) of the transaction during the last collection interval.

Maximum Transaction Response Time

This monitor returns the maximum response time (in seconds) of the transaction during the last collection interval.

Bucket Count #1

This monitor returns the number of times the transaction response time was below bucket boundary 1.

Bucket Count #2

This monitor returns the number of times the transaction response time fell between bucket boundaries 1 and 2.

Bucket Count #3

This monitor returns the number of times the transaction response time fell between bucket boundaries 2 and 3.

Bucket Count #4

This monitor returns the number of times the transaction response time fell between bucket boundaries 3 and 4.

Bucket Count #5

This monitor returns the number of times the transaction response time fell between bucket boundaries 4 and 5.

Bucket Count #6

This monitor returns the number of times the transaction response time fell between bucket boundaries 5 and 6.

Bucket Count #7

This monitor returns the number of times the transaction response time was above bucket boundary 6.

Hung Transaction

This is an asynchronous monitor. It returns information about whether a transaction did not update or stop within the maximum length of time specified in the ARM client's configuration file.

Transaction Too Long

This is an asynchronous monitor. It returns information about whether a transaction did not stop within the amount of time specified in the ARM client's configuration file.

Failed Transaction

This is an asynchronous monitor. It returns information about whether a transaction completed with the failed state.

Aborted Transaction

This is an asynchronous monitor. It returns information about whether a transaction completed with the aborted state.

3

Compaq Insight Manager Monitoring Collection Introduction

The Compaq Insight Manager monitoring collection allows you to monitor most of the MIBs defined within the Compaq Insight Manager. This collection supports all values except those for Disk Array and SCSI Disks.

Note: If you install the TME 10 SNMP monitoring collection from the TME 10 desktop, the TME 10 SNMP monitoring collection, the TME 10 Compaq Insight Manager monitoring collection, and the TME 10 SNMP MIB-II monitoring collection are all installed as part of the TME 10 SNMP monitoring collection installation process. If you install the TME 10 SNMP monitoring collection from the command line, you must install the TME 10 Compaq Insight Manager monitoring collection individually from the command line.

Monitoring Sources

The following tables list the monitoring sources provided with this monitoring collection. The tables include the name of the monitoring source as it is displayed in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog, its CLI equivalent, and the OID that it monitors.

Common MIB Hardware Monitoring Sources

GUI Name	CLI Name	OID
Physical System Memory	HardwareMemory	1.3.6.1.4.1.232.1.2.3.2.0
System Keyboard	HardwareKbd	1.3.6.1.4.1.232.1.2.7.1.0
Video Controller	HardwareVideo	1.3.6.1.4.1.232.1.2.8.1.0
System Model	HardwareProduct	1.3.6.1.4.1.232.2.2.4.2.0
System Serial Number	HardwareSerNo	1.3.6.1.4.1.232.2.2.2.1.0
Processor Number	HardwareCPUIdx	1.3.6.1.4.1.232.1.2.2.1.1.1.0
Processor Name	HardwareCPUName	1.3.6.1.4.1.232.1.2.2.1.1.3.0
Processor Speed	HardwareCPUSpeed	1.3.6.1.4.1.232.1.2.2.1.1.4.0
Floppy Drive Type	HardwareFpyType	1.3.6.1.4.1.232.1.2.11.1.1.2.0

SNMP Monitoring Sources

GUI Name	CLI Name	OID
SNMP Software Type	SoftwareType	1.3.6.1.4.1.232.11.2.7.2.1.3.0
SNMP Software Binary	SoftwareName	1.3.6.1.4.1.232.11.2.7.2.1.4.0
SNMP Software Version	SoftwareVers	1.3.6.1.4.1.232.11.2.7.2.1.8.0
SNMP Software Description	SoftwareDescr	1.3.6.1.4.1.232.11.2.7.2.1.5.0

GUI Name	CLI Name	OID
SNMP Software Location	SoftwareLoc	1.3.6.1.4.1.232.11.2.7.2.1.7.0
SNMP Software Status	SoftwareStatus	1.3.6.1.4.1.232.11.2.7.2.1.2.0

Operating System Monitoring Sources

GUI Name	CLI Name	OID
Operating System Name	OSName	1.3.6.1.4.1.232.11.2.2.1.0
Operating System Version	OSVers	1.3.6.1.4.1.232.11.2.2.2.0
Operating System Description	OSDescr	1.3.6.1.4.1.232.11.2.2.3.0
Filesystem Description	OSFilesysDescr	1.3.6.1.4.1.232.11.2.4.1.1.2.0
Filesystem Size	OSFilesysSpace	1.3.6.1.4.1.232.11.2.4.1.1.3.0
Filesystem File Count	OSFilesysNames	1.3.6.1.4.1.232.11.2.4.1.1.6.0

Host MIB Hardware Monitoring Sources

GUI Name	CLI Name	OID
System ROM Version	HrdwrRomVers	1.3.6.1.4.1.232.1.2.6.1.0
Hardware Cache Level	HrdwrCacheLvl	1.3.6.1.4.1.232.1.2.2.3.1.2.0.2
Hardware Cache Size	HrdwrCacheSiz	1.3.6.1.4.1.232.1.2.2.3.1.3.0.2

GUI Name	CLI Name	OID
Hardware Cache Speed	HrdwrCacheSpeed	1.3.6.1.4.1.232.1.2.2.3.1.4.0.2
Hardware Cache Status	HrdwrCacheStat	1.3.6.1.4.1.232.1.2.2.3.1.5.0.2
Hardware Serial Port Address	HrdwrSerPortAdr	1.3.6.1.4.1.232.1.2.9.1.1.2.0
Hardware Parallel Port Address	HrdwrParPortAdr	1.3.6.1.4.1.232.1.2.10.1.1.2.0

CPU/Disk Monitoring Sources

GUI Name	CLI Name	OID
1 Minute Load Average	ResrcCPU1Min	1.3.6.1.4.1.232.11.2.3.1.1.2.0
5 Minute Load Average	ResrcCPU5Min	1.3.6.1.4.1.232.11.2.3.1.1.3.0
30 Minute Load Average	ResrcCPU30Min	1.3.6.1.4.1.232.11.2.3.1.1.4.0
60 Minute Load Average	ResrcCPU60Min	1.3.6.1.4.1.232.11.2.3.1.1.5.0
Filesystem Used	ResrcFSUsedMB	1.3.6.1.4.1.232.11.2.4.1.1.4.0
File Count	ResrcFSFilesCnt	1.3.6.1.4.1.232.11.2.4.1.1.7.0
Filesystem % Used	ResrcFSUsedPct	1.3.6.1.4.1.232.11.2.4.1.1.5.0
Maximum File Count	ResrcFSFilesMax	1.3.6.1.4.1.232.11.2.4.1.1.6.0

Environment Monitoring Sources

GUI Name	CLI Name	OID
Overall Thermal Condition	EnvTempCond	1.3.6.1.4.1.232.6.2.6.1.0
System Temperature Status	EnvTempStat	1.3.6.1.4.1.232.6.2.6.3.0
System Fan Status	EnvFanStat	1.3.6.1.4.1.232.6.2.6.4.0
CPU Fan Status	EnvCPUFanStat	1.3.6.1.4.1.232.6.2.6.5.0

EISA Bus Monitoring Sources

GUI Name	CLI Name	OID
Server Health Uptime	HutilPowerOn	1.3.6.1.4.1.232.6.2.8.1.0
1 Minute EISA Load	HutilEISA1Min	1.3.6.1.4.1.232.6.2.8.2.0
5 Minute EISA Load	HutilEISA5Min	1.3.6.1.4.1.232.6.2.8.3.0
30 Minute EISA Load	HutilEISA30Min	1.3.6.1.4.1.232.6.2.8.4.0
60 Minute EISA Load	HutilEISA60Min	1.3.6.1.4.1.232.6.2.8.5.0

4

Compaq Insight Manager Monitoring Collection Sources

1 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last minute. The CLI name for this monitor is **HutilEISA1Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.2.0.

1 Minute Load Average

This monitor returns an integer which represents the average load of the system for past minute. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU1Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.2.0.

5 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last five minutes. The CLI name for this monitor is **HutilEISA5Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.3.0.

5 Minute Load Average

This monitor returns an integer which represents the average load of the system for past five minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU5Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.3.0.

30 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last thirty minutes. The CLI name for this monitor is **HutilEISA30Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.4.0.

30 Minute Load Average

This monitor returns an integer which represents the average load of the system for past thirty minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU30Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.4.0.

60 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last sixty minutes. The CLI name for this monitor is **HutilEISA60Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.5.0.

60 Minute Load Average

This monitor returns an integer which represents the average load of the system for past sixty minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU60Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.5.0.

CPU Fan Status

This monitor returns a string which represents the current status of the processor fan(s) in the system. Possible values are **other**, **ok**, **degraded**, and **failed.** A value of **other** indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of **failed** is ever reached. The CLI name for this monitor is **EnvCPUFanStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.5.0.

File Count

This monitor returns an integer which represents the current number of files that exist on the primary filesystem. The CLI name for this monitor is **ResrcFSFilesCnt**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.7.0.

Filesystem % Used

This monitor returns an integer which represents the percentage of disk space used on the primary filesystem. The CLI name for this monitor is **ResrcFSUsedPct.** It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.5.0.

Filesystem Description

This monitor returns a string which contains the type of filesystem being used by the operating system running on the system. The CLI name for this monitor is **OSFilesysDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.2.0.

Filesystem File Count

This monitor returns an integer which indicates the current number of files on the filesystem. The CLI name for this monitor is **OSFilesysNames**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.6.0.

Filesystem Size

This monitor returns an integer which indicates the size (in KB) of the filesystem (on which the OS resides). The CLI name for this monitor is **OSFilesysSpace**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.3.0.

Filesystem Used

This monitor returns an integer which represents the amount of disk space (in MB) used on the primary filesystem. The CLI name for this monitor is **ResrcFSUsedMB**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.4.0.

Floppy Drive Type

This monitor returns a string which indicates which type of floppy drive the system has. The possible choices are **360K**, **720K**, **1.2Mb**, **1.44Mb**, and **other**. The CLI name for this monitor is **HardwareFpyType**. It monitors OID 1.3.6.1.4.1.232.1.2.11.1.1.2.0.

Hardware Cache Level

This monitor returns an integer indicating the level of the cache, where level 1 is on-board the CPU, etc. The CLI name for this monitor is **HrdwrCacheLvl**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.2.0.2.

Hardware Cache Size

This monitor returns an integer which contains the size (in KB) of the hardware cache. The CLI name for this monitor is **HrdwrCacheSize**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.3.0.2.

Hardware Cache Speed

This monitor returns an integer which contains the speed (in nanoseconds) of the hardware cache. A value of 0 means the cache speed is unknown. The CLI name for this monitor is **HrdwrCacheSpeed**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.4.0.2.

Hardware Cache Status

This monitor returns a string which contains the status of the cache. The possible values are **unknown**, **ok**, **degraded**, and **failed**. The CLI name for this monitor is **HrdwrCacheStat**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.5.0.2.

Hardware Parallel Port Address

This monitor returns an integer which represents the hardware address of the system parallel port. The CLI name for this monitor is **HrdwrParPortAdr**. It monitors OID 1.3.6.1.4.1.232.1.2.10.1.1.2.0.

Hardware Serial Port Address

This monitor returns an integer which represents the hardware address of the system serial port. The CLI name for this monitor is **HrdwrSerPortAdr**. It monitors OID 1.3.6.1.4.1.232.1.2.9.1.1.2.0.

Maximum File Count

This monitor returns an integer which represents the maximum number of files that can exist on the primary filesystem. The CLI name for this monitor is **ResrcFSFilesMax.** It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.6.0.

Operating System Description

This monitor returns a string which contains a description of the operating system running on the system. The CLI name for this monitor is **OSDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.2.3.0.

Operating System Name

This monitor returns a string which contains the name of the operating system running on the system. The CLI name for this monitor is **OSName**. It monitors OID 1.3.6.1.4.1.232.11.2.2.1.0.

Operating System Version

This monitor returns a string which contains the version of the operating system running on the system. The CLI name for this monitor is **OSVers**. It monitors OID 1.3.6.1.4.1.232.11.2.2.2.0.

Overall Thermal Condition

This monitor returns a string which represents the current overall temperature condition of the system. Possible values are **other**, **ok**, **degraded**, and **failed.** A value of **other** indicates the system does not monitor this indication. The CLI name for this monitor is **EnvTempCond**. It monitors OID 1.3.6.1.4.1.232.6.2.6.1.0.

Physical System Memory

This monitor returns an integer indicating the amount (in KB) of physical memory in the system. The CLI name for this monitor is **HardwareMemory**. It monitors OID 1.3.6.1.4.1.232.1.2.3.2.0.

Processor Name

This monitor returns a string that indicates which CPU chip is present. The CLI name for this monitor is **HardwareCPUName**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.3.0.

Processor Number

This monitor returns an integer that designates a CPU in a MP system. The CLI name for this monitor is **HardwareCPUIdx**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.1.0.

Processor Speed

This monitor returns an integer which indicates the clock speed (in Mhz) of the CPU. The CLI name for this monitor is **HardwareCPUSpeed**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.4.0.

Server Health Uptime

This monitor returns an integer which represents the current number of files that exist on the primary filesystem. The CLI name for this monitor is **HutilPowerOn**. It monitors OID 1.3.6.1.4.1.232.6.2.8.1.0.

SNMP Software Binary

This monitor returns a string that contains the name of the executable which is providing the SNMP agent functionality. The CLI name for this monitor is **SoftwareName**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.4.0.

SNMP Software Description

This monitor returns a string that describes the SNMP agent software.

The CLI name for this monitor is **SoftwareDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.5.0.

SNMP Software Location

This monitor returns a string which contains the full path to the executable which is providing the SNMP agent functionality. The CLI name for this monitor is **SoftwareLoc**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.7.0.

SNMP Software Status

This monitor returns a string which contains the operational status of the SNMP agent software. Possible values are **loaded**, **notloaded**, and **other**. The CLI name for this monitor is **SoftwareStatus**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.2.0.

SNMP Software Type

This monitor returns a string which indicates what type of software is providing the SNMP agent functionality. Possible values are **driver**, **agent**, **sysutil**, and **other**. The CLI name for this monitor is **SoftwareType**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.3.0.

SNMP Software Version

This monitor returns a string that indicates the version of the SNMP agent software. The CLI name for this monitor is **SoftwareVers**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.8.0.

System Fan Status

This monitor returns a string which represents the current status of the fan(s) in the system. Possible values are **other**, **ok**, **degraded**, and **failed.** A value of other indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of **failed** is ever reached. The CLI name for this monitor is **EnvFanStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.4.0.

System Keyboard

This monitor returns a string that describes the system keyboard. The CLI name for this monitor is **HardwareKbd**. It monitors OID 1.3.6.1.4.1.232.1.2.7.1.0.

System Model

This monitor returns a string that indicates the system model. The CLI name for this monitor is **HardwareProduct**. It monitors OID 1.3.6.1.4.1.232.2.2.4.2.0.

System ROM Version

This monitor returns a string which contains the system ROM identification. The CLI name for this monitor is **HrdwrRomVers**. It monitors OID 1.3.6.1.4.1.232.1.2.6.1.0.

System Serial Number

This monitor returns a string that indicates the system serial number. The CLI name for this monitor is **HardwareSerNo**. It monitors OID 1.3.6.1.4.1.232.2.2.2.1.0.

System Temperature Status

This monitor returns a string which represents the current overall temperature status of the system. Possible values are **other**, **ok**, **degraded**, and **failed.** A value of **other** indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of failed is ever reached. The CLI name for this monitor is **EnvTempStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.3.0.

Video Controller

This monitor returns a string that describes the system video adaptor. The CLI name for this monitor is **HardwareVideo**. It monitors OID 1.3.6.1.4.1.232.1.2.8.1.0.

5

Netware Monitoring Collection

The Netware monitoring collection enables you to use TME 10 Distributed Monitoring to monitor a Netware system.

Netware Monitoring Collection

The following table lists the monitoring sources for NetWare 3.1x and NetWare 4.1x. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

GUI Name	CLI Name
CPU Utilization	CPUUtilization
Critical File Monitor	CriticalFile
Process Count	ProcessCount
Thread Count	ThreadCount
Volume Space Used	VolumeSpaceUsed
Volume Space Remaining	VolumeSpaceRemaining
% Volume Space Used	VolumeUsedSpacePercent

GUI Name	CLI Name
% Volume Free Space	VolumeFreeSpacePercent
Number of NetWare Connected Users	Numberofconnectedusers
Cache Blocks in Use	CacheBlocksinUse
Percentage of Cache Blocks iin Use	PercentofCacheinUse
No ECBs Available	NoECBsAvailableCount
Logins Enabled	HostStatus
Server Up Time	ServerUpTime
Application Monitor	AppStatus

The following table lists the monitoring sources that are only available for NetWare 3.1x/Bindery. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

GUI Name	CLI Name
Print Jobs	JobsInQueue
Print Job Size	SizeOfQueue

The following table lists the monitoring sources that are only available for NetWare 4.1x. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

GUI Name	CLI Name
Error Packets Received	ErrorPacketsReceived
Error Packets Transmitted	ErrorPacketsTransmitted
Error Percent Packets Received	ErrorPercentPacketsReceived
Error Percent Packets Transmitted	ErrorPercentPacketsTransmitted
Total # Cache Buffers	TotalCacheBuffers
Original # Cache Buffers	OriginalCacheBuffers
Dirty Cache Buffers	DirtyCacheBuffers
LRU Time	LRU

CPU Utilization

This monitor returns information on the current CPU utilization. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **CPUUtilization**.

Critical File Monitor

This monitor returns information on a critical file. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **CriticalFile**.

Process Count

This monitor returns the number of processes currently running. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **ProcessCount**.

Thread Count

This monitor returns the number of threads that are currently running. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **ThreadCount**.

Volume Space Used

This monitor returns the amount of space (in MB) used on a volume. You must specify the desired volume name. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The volume name must be followed by a colon (for example; **Disk1:**). The CLI name for this monitor is **VolumeSpaceUsed**.

Volume Space Remaining

This monitor returns the amount of space remaining (in MB) on a volume. You must specify the desired volume name. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The volume name must be followed by a colon (for example; **Disk1:**). The CLI name for this monitor is **VolumeSpaceRemaining**.

% Volume Space Used

This monitor returns the amount of space (as a percentage of total space) used on a volume. You must specify the desired volume name. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The volume name must be followed by a colon (for example; **Disk1:**). The CLI name for this monitor is **VolumeUsedSpacePercent**.

% Volume Space Free

This monitor returns the amount of space (as a percentage of total space) free on a volume. You must specify the desired volume name. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The volume name must be followed by a colon (for example; **Disk1:**). The CLI name for this monitor is **VolumeFreeSpacePercent.**

Number of NetWare Connected Users

This monitor returns the number of NetWare users that are currently connected. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **NumberofConnectedUsers**.

Cache Blocks in Use

This monitor returns the number of cache blocks that are currently in use. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **CacheBlocksinUse**.

Percentage of Cache Blocks in Use

This monitor returns the percentage of cache space that is currently in use. This monitor can have three threshold levels (**warning**, **critical**, and **severe**) set. The CLI name for this monitor is **PercentofCacheinUse**.

No ECBs Available

This monitor returns information to indicate that no ECBs are currently available. The CLI name for this monitor is **NoECBsAvailableCount**.

Logins Enabled

This monitor returns information that indicates if a host is up or down. The CLI name for this monitor is **HostStatus**.

Server Up Time

This monitor returns information on the amount of time that a server has been up. The CLI name for this monitor is **ServerUpTime**.

Application Monitor

This monitor returns information the availability of an application. The CLI name for this monitor is **AppStatus**.

Print Jobs

This monitor returns information on the current number of print jobs. The CLI name for this monitor is **JobsinQueue**.

Print Job Size

This monitor returns information on the size of a print job. The CLI name for this monitor is **SizeofQueue**.

Error Packets Received

This monitor returns information on the number of error packets that have been received. The CLI name for this monitor is **ErrorPacketsReceived**.

Error Packets Transmitted

This monitor returns information on the number of error packets that have been transmitted. The CLI name for this monitor is **ErrorPacketsTransmitted**.

Error Percent Packets Received

This monitor returns information on the number of error packets received (as a percentage of the total number of packets received). The CLI name for this monitor is **PercentErrorPacketsReceived**.

Error Percent Packets Transmitted

This monitor returns information on the number of error packets transmitted (as a percentage of the total number of packets sent). The CLI name for this monitor is **PercentErrorPacketsTransmitted**.

Total # Cache Buffers

This monitor returns information on the total number of cache buffers. The CLI name for this monitor is **TotalCacheBuffers**.

Original Cache Buffers

This monitor returns information on the number of original cache buffers. The CLI name for this monitor is **OriginalCacheBuffers**.

Dirty Cache Buffers

This monitor returns information on the number of dirty cache buffers. The CLI name for this monitor is **DirtyCacheBuffers**.

LRU Time

This monitor returns information on the LRU time. The CLI name for this monitor is **LRU**.

6

NT Monitoring Collections

The set of NT monitoring collections provide the same monitoring capabilities as the NT Performance Monitor application. Each collection contains several monitoring sources which allow you to manage different aspects of your network, such as system memory, disk space, networking protocols, and server status.

By using TME 10 Distributed Monitoring with the NT monitoring collections, you can easily manage any number of NT hosts from a central location. You can distribute a TME 10 Distributed Monitoring profile so that all NT servers monitor the same internal resources and make the same corrective actions if a threshold is reached. Additionally, updating the monitors as your network configuration changes is simpler than updating the alerts in Performance Monitor on multiple NT machines. Just add a new monitor to a TME 10 Distributed Monitoring profile, or edit an existing one, and distribute the profile.

The NT monitoring collections do not run on top of or interact with the Performance Monitor. Nor do they affect the alerts you have already established. Therefore, it is possible to monitor a resource with TME 10 Distributed Monitoring and the Performance Monitor simultaneously.

The set of NT monitoring collections can be divided into three categories:

- Core NT resource monitoring collections, which allow you to monitor NT objects that are installed on all NT servers and clients.
- Network protocol monitoring collections, which allow you to monitor the availability and use of standard network protocols, such as TCP/IP, and those defined by Novell and Microsoft.

- Network resource monitoring collections, which allow you to monitor the NT objects necessary to use network protocols.

Core NT Resource Monitoring Collections

Monitoring Collection	Described on Page
NT Browser	6-4
NT Cache	6-5
NT_EventLog	6-9
NT Logical Disk	6-15
NT Memory	6-17
NT Network Monitor	6-24
NT Objects	6-30
NT Paging File	6-31
NT Physical Disk	6-32
NT Process	6-33
NT Processor	6-35
NT Redirector	6-36
NT Server	6-39
NT Server Work Queues	6-41
NT System	6-42
NT Thread	6-45

Network Protocol Monitoring Collections

Monitoring Collection	Described on Page
NT ICMP	6-11
NT IP	6-13
NT NetBEUI	6-20
NT NWLink IPX	6-25
NT NWLink NetBIOS	6-28
NT TCP	6-44
NT UDP	6-46

Network Resource Monitoring Collections

Monitoring Collection	Described on Page
NT Client Services for NetWare	6-7
NT NBT Connection	6-19
NT NetBEUI Resource	6-22
NT Network Interface	6-23

The tables on the following pages in this chapter list the monitoring sources of each TME 10 NT monitoring collection. The left column lists the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The right column is the internal name of the monitor, which you can use to write scripts to define default values.

NT_Browser Monitoring Collection

A browser allows users to locate resources on a network. Each domain has a master browser and usually has one or more backup browsers. When a server connects to a network, it initially announces itself to the master browser. After the connection is established, the server periodically reannounces its presence. The **NT_Browser** monitoring collection measures browser availability and usage.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept arguments.

GUI Name	CLI Name
Announcements Domain/sec	AnnDomPerSec
Announcements Server/sec	AnnSrvPerSec
Announcements Total/sec	AnnTotPerSec
Duplicate Master Announcements	DupMasterAnn
Election Packets/sec	ElectionPktsPerSec
Enumerations Domain/sec	EnumDomPerSec
Enumerations Other/sec	EnumOtherPerSec
Enumerations Server/sec	EnumSrvPerSec
Enumerations Total/sec	EnumTotPerSec
Illegal Datagrams/sec	IllegalGramsPerSec
Mailslot Allocations Failed	MailslotAllocFailed

GUI Name	CLI Name
Mailslot Opens Failed/sec	MailslotOpensFailedPerSec
Mailslot Receives Failed	MailslotRcvFailed
Mailslot Writes Failed	MailslotWrFailed
Mailslot Writes/sec	MailslotWrPerSec
Missed Mailslot Datagrams	MissedMailslotGrams
Missed Server Announcements	MissedSrvAnn
Missed Server List Requests	MissSrvListReq
Server Announce Allocations Failed/sec	SrvAnnAllocFailPerSec
Server List Requests/sec	SrvListReqPerSec

NT_Cache Monitoring Collection

The **NT_Cache** monitoring collection monitors the activity of cache memory. A cache is a type of memory that is designed to hold recently-accessed data so that this data can be re-accessed quicker. Files on Windows NT are cached in main memory in units of pages. The cache preserves file pages in memory for as long as possible to permit access to the data through the file system without having to access the disk.

When the system reads from or writes to main memory, a copy is also saved in the cache, along with the associated main memory address. The cache monitors addresses of subsequent reads to see if the required data is already in the cache. If the required data is in the cache (a “cache hit”), then the data is returned immediately and the main memory read is aborted (or is not started). If the data is not cached (a “cache miss”), then the data is fetched from main memory and also saved in the cache.

Note: Use the **NT_Memory** monitoring collection to monitor main memory.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Async Copy Reads/sec	AsyncCopyRdPerSec
Async Data Maps/sec	AsyncDataMapsPerSec
Async Fast Reads/sec	AsyncFastRdPerSec
Async MDL Reads/sec	AsyncMDLRdPerSec
Async Pin Reads/sec	AsyncPinRdPerSec
Copy Read Hits Percentage	CopyRdHitsPrc
Copy Reads/sec	CopyRdPerSec
Data Flush Pages/sec	DataFlushPagPerSec
Data Flushes/sec	DataFlushPerSec
Data Map Hits Percentage	DataMapHitsPrc
Data Map Pins/sec	DataMapPinsPerSec
Data Maps/sec	DataMapsPerSec
Fast Read Not Possibles/sec	FastRdNotPossPerSec
Fast Read Resource Misses/sec	FastRdResMissPerSec
Fast Reads/sec	FastRdPerSec

GUI Name	CLI Name
Lazy Write Flushes/sec	LazyWrFlushesPerSec
Lazy Write Pages/sec	LazyWrPagPerSec
MDL Read Hits Percentage	MDLRdPerSec
MDL Reads/sec	MDLRdHitsPrc
Pin Read Hits Percentage	PinRdHitsPrc
Pin Reads/sec	PinRdPerSec
Sync Copy Reads/sec	SyncCopyRdPerSec
Sync Data Maps/sec	SyncDataMapsPerSec
Sync Fast Reads/sec	SyncFastRdPerSec
Sync MDL Reads/sec	SyncMDLRdPerSec
Sync Pin Reads/sec	SyncPinRdPerSec

NT_ClientServiceForNW Monitoring Collection

Client Service for NetWare allows users to access file and print resources on NetWare servers. The **NT_ClientServiceForNW** monitoring collection monitors the activity of this service.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Bytes Total/sec	BytesTotPerSec
File Data Operations/sec	FileDataOperPerSec
Packets/sec	PktsPerSec
Bytes Received/sec	BytesRcvPerSec
Packets Received/sec	PktsRcvPerSec
Bytes Transmitted/sec	BytesTransPerSec
Packets Transmitted/sec	PktsTransPerSec
File Read Operations/sec	FileRdOperPerSec
Read Operations Random/sec	RdOperRandPerSec
Read Packets/sec	RdPktsPerSec
File Write Operations/sec	FileWrOperPerSec
Write Operations Random/sec	WrOperRandPerSec
Write Packets/sec	WrPktsPerSec
Server Sessions	SrvSessions
Server Reconnects	SrvReconnects
Connect NetWare 2.x	ConnectNW2TwoX
Connect NetWare 3.x	ConnectNWThreeX
Connect NetWare 4.x	ConnectNWFourX
Server Disconnects	SrvDisconnects

GUI Name	CLI Name
Packet Burst Read NCP Count/sec	PktBrstRdNCPCntPerSec
Packet Burst Read Timeouts/sec	PktBrstRdTimeoutsPerSec
Packet Burst Write NCP Count/sec	PktBrstWrNCPCntPerSec
Packet Burst Write Timeouts/sec	PktBrstWrTimeoutsPerSec
Packet Burst IO/sec	PktBrstIOPerSec

NT_EventLog Monitoring Collection

The NT event log facility logs important system events. The NT_EventLog monitoring collection provides the capability to monitor events as they are logged by the NT event log facility.

The following table lists the monitoring sources that this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

The **Application Log Event Total**, **System Log Event Total**, and **Security Log Event Total** monitors return the total number of events logged to the event log. You can specify a numeric argument. The monitor is activated when the event total threshold is reached.

All of the other monitors in this collection have a new operator group with only one value, “**Event is logged**.” The monitor is triggered when a new event is logged. The **Application Log Event**, **System Log Event**, and **Security Log Event** monitors are triggered when any application, system, or security event, respectively, is logged. The other log event monitors are triggered when an event of the specified type and source is logged.

Monitoring Collection	CLI Name
Application Log Event	Appevent
Application Log Information Event	Appinfoevent
Application Log Audit_Success Event	Appsuccevent
Application Log Audit_Failure Event	Appfailevent
Application Log Warning Event	Appwarnevent
Application Log Error Event	Apperrevent
Application Log Event Total	Applnumevent
Application Log Source	Applsrcevent
System Log Event	Sysevent
System Log Information Event	Sysinfoevent
System Log Audit_Success Event	Syssuccevent
System Log Audit_Failure Event	Sysfailevent
System Log Warning Event	Syswarnevent
System Log Error Event	Syserrevent
System Log Event Total	Sysnumevent
System Log Source	Syslsrcevent
Security Log Event	Secevent

Monitoring Collection	CLI Name
Security Log Information Event	Secinfoevent
Security Log Audit_Success Event	Secsuccevent
Security Log Audit_Failure Event	Secfailevent
Security Log Warning Event	Secwarnevent
Security Log Error Event	Secerrevent
Security Log Event Total	Secnumevent
Security Log Source	Seclsrcevent

NT_ICMP Monitoring Collection

The Internet Control Message Protocol (ICMP) allows gateways and hosts to send error and control messages to other gateways and hosts. ICMP provides communication between the Internet software between any two machines. This protocol is part of Internet Protocol (IP).

Some of the **NT_ICMP** monitoring sources measure ICMP activity on subscribing hosts. Others monitor the various error counts specific to the ICMP protocol.

Note: Monitors created from this monitoring collection will not be able to monitor for ICMP activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Messages Outbound Errors	MessagesOutErrors
Messages Received Errors	MessagesRcvErrors
Messages Received/sec	MessagesRcvPerSec
Messages Sent/sec	MessagesSentPerSec
Messages/sec	MessagesPerSec
Received Address Mask	RcvAddressMask
Received Address Mask Reply	RcvAddressMaskReply
Received Destination Unreachable	RcvDestUnreachable
Received Echo Reply/sec	RcvEchoReplyPerSec
Received Echo/sec	RcvEchoPerSec
Received Parameter Problem	RcvParamProb
Received Redirect/sec	RcvRedirectPerSec
Received Source Quench	RcvSrcQuench
Received Time Exceeded	RcvTimeExceeded
Received Timestamp Reply/sec	RcvTimestampReplyPerSec
Received Timestamp/sec	RcvTimestampPerSec
Sent Address Mask	SentAddressMask
Sent Address Mask Reply	SentAddressMaskReply

GUI Name	CLI Name
Sent Destination Unreachable	SentDestUnreachable
Sent Echo Reply/sec	SentEchoReplyPerSec
Sent Echo/sec	SentEchoPerSec
Sent Parameter Problem	SentParamProb
Sent Redirect/sec	SentRedirectPerSec
Sent Source Quench	SentSrcQuench
Sent Time Exceeded	SentTimeExceeded
Sent Timestamp/sec	SentTimestampPerSec
Sent Timestamp Reply/sec	SentTimestampReplyPerSec

NT_IP Monitoring Collection

The Internet Protocol (IP) routes small messages from one machine to another based on address information carried in the message. These messages are known as datagrams. All datagrams contain a header segment and a data segment. The header includes the source and destination addresses as well as fragmentation information. This fragmentation information specifies how the system should divide a datagram into two or more pieces.

The **NT_IP** monitoring sources measure IP activity on subscribing hosts. The most common items to monitor include problems in sending and receiving datagrams and fragments of datagrams.

Note: Monitors created from this monitoring collection will not be able to monitor for IP activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 DIstributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to

configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Datagrams/sec	GramsPerSec
Datagrams Forwarded/sec	GramsForwardedPerSec
Datagrams Outbound Discarded	GramsOutDiscarded
Datagrams Outbound No Route	GramsOutNoRoute
Datagrams Received/sec	GramsRcvPerSec
Datagrams Received Address Errors	GramsRcvAddrErrors
Datagrams Received Delivered/sec	GramsRcvDelPerSec
Datagrams Received Discarded	GramsRcvDiscarded
Datagrams Received Header Errors	GramsRcvHeaderErrors
Datagrams Received Unknown Protocol	GramsRcvUnkProtocol
Datagrams Sent/sec	GramsSentPerSec
Fragment Reassembly Failures	FragReassemFailures
Fragmentation Failures	FragFailures
Fragmented Datagrams/sec	FragGramsPerSec
Fragments Created/sec	FragCreatedPerSec
Fragments Reassembled/sec	FragReassemPerSec

GUI Name	CLI Name
Fragments Received/sec	FragRcvPerSec

NT_LogicalDisk Monitoring Collection

The **NT_LogicalDisk** monitoring collection measures activity on a logical disk of a subscribing managed node. A logical disk is a partition on a hard or fixed disk drive and is assigned a drive letter, such as **C**. A physical disk can contain more than one logical disk. Each logical disk stores file, program, and page data. The system reads the disk to retrieve this data, and writes to the disk to record changes in the data.

The following monitoring sources are commonly used to measure the activity of a logical disk:

- Disk Queue Length
- Disk Reads/sec
- Disk Transfers/sec
- Disk Writes/sec
- Free Megabytes
- Percent Disk Read Time
- Percent Disk Time
- Percent Disk Write Time
- Percent Free Space

Note: Use the **NT_PhysicalDisk** monitoring collection to monitor hard and fixed disks.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a logical disk as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Avg Disk Bytes/Read	AvgDskBytesPerRd
Avg Disk Bytes/Transfer	AvgDskBytesPerTran
Avg Disk Bytes/Write	AvgDskBytesPerWr
Avg Disk sec/Read	AvgDskSecPerRd
Avg Disk sec/Transfer	AvgDskSecPerTran
Avg Disk sec/Write	AvgDskSecPerWr
Disk Bytes/sec	DskBytesPerSec
Disk Queue Length	DskQueLen
Disk Read Bytes/sec	DskRdBytesPerSec
Disk Reads/sec	DskRdPerSec
Disk Transfers/sec	DskTranPerSec
Disk Write Bytes/sec	DskWrBytesPerSec
Disk Writes/sec	DskWrPerSec
Free Megabytes	FreeMegabytes
Percent Disk Read Time	PrcDskRdTime
Percent Disk Time	PrcDskTime
Percent Disk Write Time	PrcDskWrTime
Percent Free Space	PrcFreeSpace

NT_Memory Monitoring Collection

The **NT_Memory** monitoring collection monitors the activity of both real and virtual memory on the subscribing managed node. Real memory is allocated in units of pages. Virtual memory may exceed real memory in size, causing page traffic as virtual pages are moved between disk and real memory.

The following monitoring sources are commonly used to measure the activity of a real and virtual memory:

- Page Faults/sec
- Pages Input/sec
- Pages/sec

Note: Use the **NT_Cache** monitoring collection to monitor cache memory.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Available Bytes	AvailBytes
Cache Bytes	CacheBytes
Cache Bytes Peak	CacheBytesPeak
Cache Faults/sec	CacheFltsPerSec
Commit Limit	CommitLimit
Committed Bytes	CommittedBytes
Demand Zero Faults/sec	DemZeroFltsPerSec

GUI Name	CLI Name
Free System Page Table Entries	FreeSysPgTableEntries
Page Faults/sec	PgFltsPerSec
Page Reads/sec	PgRdPerSec
Page Writes/sec	PgWrPerSec
Pages Input/sec	PagesInputPerSec
Pages Output/sec	PagesOutputPerSec
Pages/sec	PagesPerSec
Pool Nonpaged Allocs	PoolNonpagedAllocs
Pool Nonpaged Bytes	PoolNonpagedBytes
Pool Paged Allocs	PoolPagedAllocs
Pool Paged Bytes	PoolPagedBytes
Pool Paged Resident Bytes	PoolPagedResBytes
System Cache Resident Bytes	SysCacheResBytes
System Code Resident Bytes	SysCodeResBytes
System Code Total Bytes	SysCodeTotBytes
System Driver Resident Bytes	SysDriverResBytes
System Driver Total Bytes	SysDriverTotBytes
Transition Faults/sec	TranFltsPerSec
Write Copies/sec	WrCopiesPerSec

NT_NBTConnection Monitoring Collection

The **NT_NBTConnection** (NetBEUI TCP/IP) monitoring collection measures the rates at which the subscribing managed node sends and receives bytes of data from a remote computer over a single NBT connection. Therefore, this monitoring collection is only useful for remote hosts connected through NBT. This monitoring collection is useful on congested servers and can be used to determine which connection is sourcing a load.

You may also want to use the following protocol-related monitoring collections:

- NT_IP
- NT_ICMP
- NT_NetBEUI
- NT_NetBEUIResource
- NT_NetworkInterface
- NT_TCP
- NT_UDP

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a connected host as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Bytes Received/sec	BytesRcvPerSec
Bytes Sent/sec	BytesSentPerSec
Bytes Total/sec	BytesTotPerSec

NT_NetBEUI Monitoring Collection

The NetBIOS End User Interface (NetBEUI) is a protocol that was designed to support networking on small, simple networks. It is the traditional standard protocol on Microsoft networking products. The **NT_NetBEUI** monitoring collection measures the connectivity metrics and problems that can occur with this protocol.

Note: Use the **NT_NetBEUIResource** monitoring collection to monitor NetBEUI buffers.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a device as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Bytes Total/sec	BytesTotPerSec
Connection Session Timeouts	ConnSessTimeouts
Connections Canceled	ConnCanceled
Connections Open	ConnOpen
Connections No Retries	ConnNoRetries
Connections With Retries	ConnWithRetries
Datagram Bytes/sec	GramBytesPerSec
Datagram Bytes Received/sec	GramBytesRcvPerSec
Datagram Bytes Sent/sec	GramBytesSentPerSec
Datagrams Received/sec	GramsRcvPerSec

GUI Name	CLI Name
Datagrams Sent/sec	GramsSentPerSec
Datagrams/sec	GramsPerSec
Disconnects Local	DisconnectsLocal
Disconnects Remote	DisconnectsRemote
Expirations Ack	ExpirationsAck
Expirations Response	ExpirationsResponse
Failures Adapter	FailuresAdapter
Failures Link	FailuresLink
Failures No Listen	FailuresNoListen
Failures Not Found	FailuresNotFound
Failures Resource Local	FailuresResLocal
Failures Resource Remote	FailuresResRemote
Frame Bytes Received/sec	FrmBytesRcvPerSec
Frame Bytes Rejected/sec	FrmBytesRejPerSec
Frame Bytes Resent/sec	FrmBytesReSentPerSec
Frame Bytes Sent/sec	FrmBytesSentPerSec
Frame Bytes/sec	FrmBytesPerSec
Frames Received/sec	FrmRcvPerSec
Frames Rejected/sec	FrmRejPerSec
Frames Re-Sent/sec	FrmReSentPerSec

GUI Name	CLI Name
Frames Sent/sec	FrmSentPerSec
Frames/sec	FrmPerSec
Packets Received/sec	PktsRcvPerSec
Packets Sent/sec	PktsSentPerSec
Packets/sec	PktsPerSec
Piggyback Ack Queued/sec	PiggyAckQuePerSec
Piggyback Ack Timeouts	PiggyAckTimeouts
Window Send Maximum	WindowSendMax
Window Send Average	WindowSendAve

NT_NetBEUIResource Monitoring Collection

The NetBIOS End User Interface (NetBEUI) is a protocol that was designed to support networking on small, simple networks. It is the traditional standard protocol on Microsoft networking products. The **NT_NetBEUIResource** monitoring collection monitors NetBEUI buffers.

Note: Use the **NT_NetBEUI** monitoring collection to monitor the NetBEUI protocol itself.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a device as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Times Exhausted	TimesExhausted
Used Average	UsedAvg
Used Maximum	UsedMax

NT_NetworkInterface Monitoring Collection

The **NT_NetworkInterface** monitoring collection measures how frequently a subscribing managed node receives and transmits bytes and packets over a network TCP/IP connection. TCP/IP, which is an acronym for Transmission Control Protocol/Internet Protocol, is the most widely used family of network protocols. You may want to use this monitoring collection with the TCP and IP monitoring collections to diagnose low-level problems.

Note: Monitors created from this monitoring collection will not be able to monitor for network interface activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of an interface when you add a monitor from this collection.

GUI Name	CLI Name
Bytes Received/sec	BytesRcvPerSec
Bytes Sent/sec	BytesSentPerSec
Bytes Total/sec	BytesTotPerSec

GUI Name	CLI Name
Current Bandwidth	CurrentBandwidth
Output Queue Length	OutputQueLen
Packets Outbound Discarded	PktsOutDiscarded
Packets Outbound Errors	PktsOutErrors
Packets Received/sec	PktsRcvPerSec
Packets Received Discarded	PktsRcvDiscarded
Packets Received Errors	PktsRcvErrors
Packets Received Non-Unicast/sec	PktsRcvNonUcastPerSec
Packets Received Unicast/sec	PktsRcvUcastPerSec
Packets Received Unknown	PktsRcvUnknown
Packets Sent Non-Unicast/sec	PktsSentNonUcastPerSec
Packets Sent Unicast/sec	PktsSentUcastPerSec
Packets Sent/sec	PktsSentPerSec
Packets/sec	PktsPerSec

NT_NetworkMonitor Monitoring Collection

Monitors created from the **NT_NetworkMonitor** monitoring collection will not be able to measure network monitor activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to

configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a device as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Broadcast Frames Received/sec	BcastFrmRcvPerSec
Broadcast Percentage	BcastPct
Multicast Frames Received/sec	McastFrmRcvPerSec
Multicast Percentage	McastPct
Network Utilization	NetUtilization
Total Bytes Received/sec	TotBytesRcvPerSec
Total Frames Received/sec	TotFrmRcvPerSec

NT_NWLinkIPX Monitoring Collection

The NT NetWare Link is a protocol developed by Microsoft that is compatible with Novell’s Internetwork Packet eXchange (IPX) protocol. The NWLink IPX transport handles datagram transmission to and from computers using the IPX protocol. The **NT_NWLinkIPX** monitoring collection allows you to monitor this protocol.

Note: Use the **NT_NWLinkNetBIOS** monitoring collection to monitor the interface that applications use to communicate over the IPX protocol.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Bytes Total/sec	BytesTotPerSec
Connection Session Timeouts	ConnSessTimeouts
Connections Canceled	ConnCanceled
Connections No Retries	ConnNoRetries
Connections Open	ConnOpen
Connections With Retries	ConnWithRetries
Datagram Bytes/sec	GramBytesPerSec
Datagram Bytes Received/sec	GramBytesRcvPerSec
Datagram Bytes Sent/sec	GramBytesSentPerSec
Datagrams Received/sec	GramsRcvPerSec
Datagrams Sent/sec	GramsSentPerSec
Datagrams/sec	GramsPerSec
Disconnects Local	DisconnectsLocal
Disconnects Remote	DisconnectsRemote
Expirations Ack	ExpirationsAck
Expirations Response	ExpirationsResponse
Failures Adapter	FailuresAdapter
Failures Link	FailuresLink
Failures No Listen	FailuresNoListen

GUI Name	CLI Name
Failures Not Found	FailuresNotFound
Failures Resource Local	FailuresResLocal
Failures Resource Remote	FailuresResRemote
Frame Bytes Received/sec	FrmBytesRcvPerSec
Frame Bytes Rejected/sec	FrmBytesRejPerSec
Frame Bytes Resent/sec	FrmBytesReSentPerSec
Frame Bytes Sent/sec	FrmBytesSentPerSec
Frame Bytes/sec	FrmBytesPerSec
Frames Received/sec	FrmRcvPerSec
Frames Re-Sent/sec	FrmReSentPerSec
Frames Rejected/sec	FrmRejPerSec
Frames Sent/sec	FrmSentPerSec
Frames/sec	FrmPerSec
Packets Received/sec	PktsRcvPerSec
Packets Sent/sec	PktsSentPerSec
Packets/sec	PktsPerSec
Piggyback Ack Queued/sec	PiggyAckQuePerSec
Piggyback Ack Timeouts	PiggyAckTimeouts
Window Send Average	WindowSendAve
Window Send Maximum	WindowSendMax

NT_NWLinkNetBIOS Monitoring Collection

The NT NetWare Link is a protocol developed by Microsoft that is compatible with Novell’s Internetwork Packet eXchange (IPX) protocol. The NWLink NetBIOS protocol layer handles the interface to applications communicating over the IPX protocol. The **NT_NWLinkIPX** monitoring collection allows you to monitor this interface.

Note: Use the **NT_NWLinkIPX** monitoring collection to monitor datagram transmission over the IPX protocol.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Bytes Total/sec	BytesTotPerSec
Connection Session Timeouts	ConnSessTimeouts
Connections Canceled	ConnCanceled
Connections No Retries	ConnNoRetries
Connections Open	ConnOpen
Connections With Retries	ConnWithRetries
Datagram Bytes Received/sec	GramBytesRcvPerSec
Datagram Bytes Sent/sec	GramBytesSentPerSec
Datagram Bytes/sec	GramBytesPerSec
Datagrams Received/sec	GramsRcvPerSec

GUI Name	CLI Name
Datagrams Sent/sec	GramsSentPerSec
Datagrams/sec	GramsPerSec
Disconnects Local	DisconnectsLocal
Disconnects Remote	DisconnectsRemote
Expirations Ack	ExpirationsAck
Expirations Response	ExpirationsResponse
Failures Adapter	FailuresAdapter
Failures Link	FailuresLink
Failures No Listen	FailuresNoListen
Failures Not Found	FailuresNotFound
Failures Resource Local	FailuresResLocal
Failures Resource Remote	FailuresResRemote
Frame Bytes Received/sec	FrmBytesRcvPerSec
Frame Bytes Rejected/sec	FrmBytesRejPerSec
Frame Bytes Re-Sent/sec	FrmBytesReSentPerSec
Frame Bytes Sent/sec	FrmBytesSentPerSec
Frame Bytes/sec	FrmBytesPerSec
Frames Received/sec	FrmRcvPerSec
Frames Rejected/sec	FrmRejPerSec
Frames Resent/sec	FrmReSentPerSec

GUI Name	CLI Name
Frames Sent/sec	FrmSentPerSec
Frames/sec	FrmPerSec
Packets Received/sec	PktsRcvPerSec
Packets Sent/sec	PktsSentPerSec
Packets/sec	PktsPerSec
Piggyback Ack Queued/sec	PiggyAckQuePerSec
Piggyback Ack Timeouts	PiggyAckTimeouts
Window Send Average	WindowSendAve
Window Send Maximum	WindowSendMax

NT_Objects Monitoring Collection

An object is any system resource, such as a file, a physical device, or memory, that can be shared by more than one process. Information about an object can be used to detect the unnecessary consumption of computer resources. Each object requires memory to store basic information about the object.

Processes and **Threads** are the most commonly used monitors of the **NT_Objects** collection.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Events	Events
Mutexes	Mutexes
Processes	Processes
Sections	Sections
Semaphores	Semaphores
Threads	Threads

NT_PagingFile Monitoring Collection

The **NT_PagingFile** monitoring collection measures activity on a paging file of a subscribing managed node. A paging file is a system file that contains the contents of the virtual pages of memory that the Virtual Memory Manager has temporarily removed from physical memory. A paging file is also known as a swap file.

This monitoring collection contains two monitoring sources:

- Percent Usage
- Percent Usage Peak

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a paging file as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Percent Usage	PrcUsage

GUI Name	CLI Name
Percent Usage Peak	PrcUsagePeak

NT_PhysicalDisk Monitoring Collection

The **NT_PhysicalDisk** monitoring collection measures activity on a physical disk of a subscribing managed node. A physical disk is a hard or fixed disk drive. It contains one or more logical partitions, which are assigned a drive letter, such as **C**. Physical disks are used to store file, program, and paging data. The system reads the disk to retrieve these items and writes to the disk to record changes to these items.

The following monitoring sources are commonly used to measure the activity of a physical disk:

- Disk Queue Length
- Disk Reads/sec
- Disk Transfers/sec
- Disk Writes/sec
- Percent Disk Read Time
- Percent Disk Time
- Percent Disk Write Time

Note: Use the **NT_LogicalDisk** monitoring collection to monitor logical disks. It contains monitoring sources that measure how much space is available.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a physical device when you add a monitor from this collection.

GUI Name	CLI Name
Avg Disk Bytes/Read	AvgDskBytesPerRd
Avg Disk Bytes/Transfer	AvgDskBytesPerTran
Avg Disk Bytes/Write	AvgDskBytesPerWr
Avg Disk sec/Read	AvgDskSecPerRd
Avg Disk sec/Transfer	AvgDskSecPerTran
Avg Disk sec/Write	AvgDskSecPerWr
Disk Bytes/sec	DskBytesPerSec
Disk Read Bytes/sec	DskRdBytesPerSec
Disk Reads/sec	DskRdPerSec
Disk Queue Length	DskQueLen
Disk Transfers/sec	DskTranPerSec
Disk Write Bytes/sec	DskWrBytesPerSec
Disk Writes/sec	DskWrPerSec
Percent Disk Read Time	PrcDskRdTime
Percent Disk Time	PrcDskTime
Percent Disk Write Time	PrcDskWrTime

NT_Process Monitoring Collection

The **NT_Process** monitoring collection measures process activity on a subscribing managed node. A process is created when a program runs and can be an application, a service, or a subsystem. A process includes a virtual address space, an executable program, one or more threads, a portion of the user’s resource quotas, and the system resources that the operating system has allocated to the threads.

The following monitoring sources are commonly used to measure the activity of a single processor:

- ID Process
- Page Faults/sec
- Percent Processor Time
- Working Set

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a process as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Elapsed Time	ElapsedTime
Handle Count	HandleCnt
ID Process	IDProc
Page Faults/sec	PgFltsPerSec
NT Service Status	NtServices
Page File Bytes	PgFileBytes
Page File Bytes Peak	PgFileBytesPeak
Percent Privileged Time	PrcPrivTime
Percent Processor Time	PrcCpuTime

GUI Name	CLI Name
Percent User Time	PrcUsrTime
Pool Nonpaged Bytes	PoolNonpagedBytes
Pool Paged Bytes	PoolPagedBytes
Priority Base	PriorityBase
Private Bytes	PrivateBytes
Thread Count	ThreadCnt
Virtual Bytes	VirtBytes
Virtual Bytes Peak	VirtBytesPeak
Working Set	WorkingSet
Working Set Peak	WorkingSetPeak

NT_Processor Monitoring Collection

The monitoring sources in the **NT_Processor** monitoring collection measure the activity of individual processors on the subscribing managed node. A processor performs arithmetic and logical computations and starts programs.

The following monitoring sources are commonly used to measure the activity of a single processor:

- Interrupts/sec
- Percent Processor Time

Note: Use the **NT_System** monitoring collection to monitor processes as a whole.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to

configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a CPU as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
APC Bypasses/sec	APCBypassesPerSec
DPC Bypasses/sec	DPCBypassesPerSec
DPC Rate	DPCRate
DPCs Queued/sec	DPCsQuePerSec
Interrupts/sec	IntsPerSec
Percent DPC Time	PrcDPCTime
Percent Interrupt Time	PrcIntTime
Percent Privileged Time	PrcPrivTime
Percent Processor Time	PrcCpuTime
Percent User Time	PrcUsrTime

NT_Redirector Monitoring Collection

The NT_Redirector monitoring collection monitors any activity related to remote drivers that a WIndows NT system can access through the system’s redirector. The redirector accepts input/output requests for remote files, named pipes, and mailslots, then sends these requests to a network service on another host. The **NT_Redirector** monitoring collection monitors the redirector on the subscribing hosts.

The following monitoring sources are commonly used to measure redirector activity:

- Bytes Total/sec
- File Data Operations/sec

- File Read Operations/sec
- File Write Operations/sec
- Network Errors/sec
- Packets/sec
- Read Bytes Network/sec
- Server Sessions
- Write Bytes Network/sec

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Bytes Received/sec	BytesRcvPerSec
Bytes Total/sec	BytesTotPerSec
Bytes Transmitted/sec	BytesTransPerSec
Connects Core	ConnectsCore
Connects Lan Manager 2.0	ConnectsLanManTwoZero
Connects Lan Manager 2.1	ConnectsLanManTwoOne
Connects Windows NT	ConnectsWinNT
Current Commands	CurrentCommands
File Data Operations/sec	FileDataOperPerSec
File Read Operations/sec	FileRdOperPerSec
File Write Operations/sec	FileWrOperPerSec

GUI Name	CLI Name
Network Errors/sec	NetErrorsPerSec
Packets/sec	PktsPerSec
Packets Received/sec	PktsRcvPerSec
Packets Transmitted/sec	PktsTransPerSec
Read Bytes Cache/sec	RdBytesCachePerSec
Read Bytes Network/sec	RdBytesNetPerSec
Read Bytes Non-Paging/sec	RdBytesNonPagPerSec
Read Bytes Paging/sec	RdBytesPagPerSec
Read Operations Random/sec	RdOperRandPerSec
Read Packets/sec	RdPktsPerSec
Read Packets Small/sec	RdPktsSmallPerSec
Reads Denied/sec	RdDeniedPerSec
Reads Large/sec	RdLargePerSec
Server Disconnects	SrvDisconnects
Server Reconnects	SrvReconnects
Server Sessions	SrvSessions
Server Sessions Hung	SrvSessionsHung
Write Bytes Cache/sec	WrBytesCachePerSec
Write Bytes Network/sec	WrBytesNetPerSec
Write Bytes Non-Paging/sec	WrBytesNonPagPerSec

GUI Name	CLI Name
Write Bytes Paging/sec	WrBytesPagPerSec
Write Operations Random/sec	WrOperRandPerSec
Write Packets Small/sec	WrPktsSmallPerSec
Write Packets/sec	WrPktsPerSec
Writes Denied/sec	WrDeniedPerSec
Writes Large/sec	WrLargePerSec

NT_Server Monitoring Collection

The **NT_Server** monitoring collection measures the interactions with an NT server. It does not monitor the machine in which the NT server is installed, nor does the collection monitor the availability of the NT server operating system. The following monitoring sources are commonly used:

- Bytes Total/sec
- Context Block Queue Time
- Context Blocks Queued/sec
- Errors Access Permissions
- Files Open
- Files Opened Total
- Server Sessions

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles. Monitors in this collection do not accept monitor arguments.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Blocking Requests Rejected	BlockingReqRej
Bytes Received/sec	BytesRcvPerSec
Bytes Total/sec	BytesTotPerSec
Bytes Transmitted/sec	BytesTransPerSec
Context Blocks Queued/sec	CntxtBlocksQuePerSec
Errors Access Permissions	ErrorsAccessPerm
Errors Granted Access	ErrorsGrantedAccess
Errors Logon	ErrorsLogon
Errors System	ErrorsSys
File Directory Searches	FileDirSearches
Files Open	FilesOpen
Files Opened Total	FilesOpenedTot
Logon Total	LogonTot
Logon/sec	LogonPerSec
Pool Nonpaged Bytes	PoolNonpagedBytes
Pool Nonpaged Failures	PoolNonpagedFailures
Pool Nonpaged Peak	PoolNonpagedPeak
Pool Paged Bytes	PoolPagedBytes
Pool Paged Failures	PoolPagedFailures

GUI Name	CLI Name
Pool Paged Peak	PoolPagedPeak
Server Sessions	SrvSessions
Sessions Errored Out	SessionsErroredOut
Sessions Forced Off	SessionsForcedOff
Sessions Logged Off	SessionsLoggedOff
Sessions Timed Out	SessionsTimedOut
Work Item Shortages	WorkItemShort

NT_ServerWorkQueues Monitoring Collection

The following table lists the monitoring sources the **NT_ServerWorkQueue** collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a CPU as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Active Threads	ActiveThreads
Available Threads	AvailThreads
Available Work Items	AvailWorkItems
Borrowed Work Items	BorrowedWorkItems
Bytes Received/sec	BytesRcvPerSec

GUI Name	CLI Name
Bytes Sent/sec	BytesSentPerSec
Bytes Transferred/sec	BytesTransPerSec
Context Blocks Queued/sec	CntxtBlocksQuePerSec
Current Clients	CurrentClients
Queue Length	QueLen
Read Bytes/sec	RdBytesPerSec
Read Operations/sec	RdOperPerSec
Total Bytes/sec	TotBytesPerSec
Total Operations/sec	TotOperPerSec
Work Item Shortages	WorkItemShort
Write Bytes/sec	WrBytesPerSec
Write Operations/sec	WrOperPerSec

NT_System Monitoring Collection

The monitoring sources in the **NT_System** monitoring collection measure the activity of all the processors on the subscribing managed node. A processor performs arithmetic and logical computations and starts programs.

The following monitoring sources are commonly used to measure total processor activity:

- File Data Operations/sec
- File Read Operations/sec
- File Write Operations/sec
- Percent Total Processor Time

- System Up Time

Note: Use the **NT_Processor** monitoring collection to monitor individual processes.

The following table lists the monitoring sources this monitoring collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

With the exception of the **Registry Number** and the **Registry String** monitors, monitors in this collection do not accept monitor arguments.

The **Registry Number** and **Registry String** monitors accept two arguments. The first argument, **RootKey**, is a fixed-choice list of four options; you can view these options by running **regedt32** on the NT system. The second argument, **Path**, is a path to the variable.

The **Registry Number** monitor monitors numeric values within the registry. The **Registry String** monitor monitors string values within the registry.

Note: The **Registry String** monitor root keys **HKEY_LOCAL_MACHINE** and **HKEY_USERS** can be used with TME 10 Distributed Monitoring's proxy feature.

GUI Name	CLI Name
Alignment Fixups/sec	AlignFixupsPerSec
Context Switches/sec	CntxtSwtchPerSec
Exception Dispatches/sec	ExceptDispPerSec
File Control Bytes/sec	FileCtrlBytesPerSec
File Control Operations/sec	FileCtrlOperPerSec
File Data Operations/sec	FileDataOperPerSec
File Read Bytes/sec	FileRdBytesPerSec

GUI Name	CLI Name
File Read Operations/sec	FileRdOperPerSec
File Write Bytes/sec	FileWrBytesPerSec
File Write Operations/sec	FileWrOperPerSec
Floating Emulations/sec	FloatEmulPerSec
Percent Total DPC Time	PrcTotDPCTime
Percent Total Interrupt Time	PrcTotIntTime
Percent Total Privileged Time	PrcTotPrivTime
Percent Total Processor Time	PrcTotCpuTime
Percent Total User Time	PrcTotUsrTime
Processor Queue Length	CpuQueLen
Registry Number	RegistryNumber
Registry String	RegistryString
System Calls/sec	SysCallsPerSec
System Up Time	SysUpTime
Total Interrupts/sec	TotIntsPerSec

NT_TCP Monitoring Collection

The Transmission Control Protocol (TCP) is built on top of the Internet Protocol (IP). TCP adds reliable communication, flow-control, multiplexing and connection-oriented communication. It provides full-duplex, process-to-process connections.

Some of the **NT_TCP** monitoring sources measure TCP activity on the subscribing hosts. Others monitor the number of TCP connections that are in each of the possible TCP connection states.

Note: Monitors created from this monitoring collection will not be able to monitor for ICMP activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Connections Active	ConnActive
Connections Established	ConnEstablished
Connection Failures	ConnFailures
Connections Passive	ConnPassive
Connections Reset	ConnReset
Segments Received/sec	SegRcvPerSec
Segments Retransmitted/sec	SegRetranPerSec
Segments Sent/sec	SegSentPerSec
Segments/sec	SegPerSec

NT_Thread Monitoring Collection

A thread is an executable entity that belongs to a process. Every running process has at least one thread. All threads in a process have equal access to the process’s address space, object handles, and other resources. The **NT_Thread** monitoring collection measures thread activity on a subscribing host.

The following monitoring source is commonly used:

- Percent Processor Time

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

You must specify the name of a thread as a monitor argument when you add a monitor from this collection.

GUI Name	CLI Name
Context Switches/sec	CntxtSwtchPerSec
Elapsed Time	ElapsedTime
ID Process	IDProc
ID Thread	IDThread
Percent Privileged Time	PrcPrivTime
Percent Processor Time	PrcCpuTime
Percent User Time	PrcUsrTime
Priority Base	PriorityBase
Priority Current	PriorityCurrent
Start Address	StartAddress
Thread State	ThreadState
Thread Wait Reason	ThreadWaitReason

NT_UDP Monitoring Collection

The User Datagram Protocol (UDP) is part of the Transmission Control

Protocol/Internet Protocol (TCP/IP) suite. The UDP allows a host that transmits a message to distinguish between two or more recipients that reside on a single machine. Each UDP message contains a destination and a source port number so that the message can be delivered correctly, and the recipient can send a reply.

Some of the **NT_UDP** monitoring sources measure UDP activity on subscribing hosts. Others monitor the various error counts specific to the UDP protocol.

Note: Monitors created from this monitoring collection will not be able to monitor for UDP activity unless SNMP is installed on the subscribing hosts.

The following table lists the monitoring sources this collection provides. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitors in this collection do not accept monitor arguments.

GUI Name	CLI Name
Datagrams No Port/sec	GramsNoPortPerSec
Datagrams Received Errors	GramsRcvErrors
Datagrams Received/sec	GramsRcvPerSec
Datagrams Sent/sec	GramsSentPerSec
Datagrams/sec	GramsPerSec

7

OS/2 Monitoring Collection

The OS/2 monitoring collection allows you to monitor system resources on OS/2 systems.

OS/2 Monitors

The following table lists the monitoring sources that are included in the OS/2 monitoring collection:

Monitor Source	OS/2 .exe file	Script Name
# Disk Space Free	os2fsinf.exe	diskavail-os2
# DIsk Space Used	os2fsinf.exe	diskused-os2
# Disk SPace %	os2fsinf.exe	diskusedpct-os2
# Application Instances	pstat (OS/2 command)	appinstances-os2
# Application Status	pstat (OS/2 command)	appstatus-os2
# File Checksum	sumos2.exe	filechk-os2
# File Compare	diffos2.exe	filedirr-os2
# File pattern matches	grep (OS/2 tool)	countsrt-os2

Monitor Source	OS/2 .exe file	Script Name
# File Permissions	ls (OS/2 tool), wc (OS/2 tool)	filesize-os2
# Host Availability	ping.exe (OS/2 TCP/IP)	host-os2
# Load Average	pstat (OS/2 command), grep (OS/2 tool)	loadavg-os2
# Memory Avail		memavail-os2
# Swap space Available	swapos2.exe	swapavail-os2
# Remote oserv Status	wping (TME 10 Framework for OS/2)	oserv-os2
Uptime		uptime-os2
# Asynchronous Numeric		
# Asynchronous String		
# Numeric Script		
# String Script		
# File System Used	os2fsinf.exe	fsused-os2
# File System Free	os2fsinf.exe	fsfree-os2
# File System % Free	os2fsinf.exe	fspctf-os2
# File System Total Free	os2fsinf.exe	fstotf-os2
# File System Total Used	os2fsinf.exe	fsto-os2
# Number of Logons	monlogons.exe	
# Number of Sessions	sessions.exe	

Monitor Source	OS/2 .exe file	Script Name
# Number of Connections	connsopens.exe	
# Number of Files Open by Clients	connsopens.exe	
# Number of Shares	shares.exe	
# Number of Bytes out to Clients	bytessent.exe	
# Number of Bytes in	bytesrcvd.exe	
# Average Response Time	avgresp.exe	
# Request Buffer Shortage	regbuf.exe	
# Big Buffer Shortage	bigbuf.exe	
# Number of Items in Print Queue	monprintq.exe	
# CPU usage (% used)	cpu_use.exe	
# Process Count		pstat and awk
# Thread Count		pstat and awk
# Available Swap Space	swapmon.exe	
# Used Swap Space	swapmon.exe	

8

SNMP Monitoring Collection Introduction

The Simple Network Management Protocol (SNMP) is a protocol designed to give a system administrator the capability to remotely manage a computer network by polling and setting terminal values and monitoring network events. SNMP contains three basic elements:

management information base (MIB) Indicates the type of information SNMP can obtain from a network. The MIB is structured like a tree. At the top of the tree is the most general information available about a network. Each branch of the tree then gets more detailed into a specific network area.

SNMP agents Collects network and terminal information as specified in the MIB.

SNMP manager Polls the agents for requested information.

Using TME 10 Distributed Monitoring as an SNMP Manager

Because TME 10 Distributed Monitoring monitors system resources, it can be used as the SNMP manager. The TME 10 Distributed Monitoring SNMP monitoring tools contain three monitoring collections that enable you to monitor the following types of MIBs:

- Compaq Insight Manager
- MIB-II (defined by RFC 1213)
- User-specified

The **wsnmpmon** command is also provided. You can use this command to test a monitor derived from the generic SNMP monitoring collection before you add it to a TME 10 Distributed Monitoring profile and distribute the profile.

9

SNMP Compaq Insight Manager Monitoring Sources

This monitoring collection allows you to monitor most of the MIBs defined within the Compaq Insight Manager. This collection supports all values except those for Disk Array and SCSI Disks.

You must specify the following argument to configure a Compaq Insight Manager monitoring source:

SystemName Specifies the name of the host to run the monitor on.

The CLI name for this collection is **Compaq**.

1 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last minute. The CLI name for this monitor is **HutilEISA1Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.2.0.

1 Minute Load Average

This monitor returns an integer which represents the average load of the system for past minute. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU1Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.2.0.

5 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last five minutes. The CLI name for this monitor is **HutilEISA5Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.3.0.

5 Minute Load Average

This monitor returns an integer which represents the average load of the system for past five minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU5Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.3.0.

30 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last thirty minutes. The CLI name for this monitor is **HutilEISA30Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.4.0.

30 Minute Load Average

This monitor returns an integer which represents the average load of the system for past thirty minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is **ResrcCPU30Min**. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.4.0.

60 Minute EISA Load

This monitor returns an integer which represents the EISA bus utilization (as a percentage of the theoretical maximum) for the last sixty minutes. The CLI name for this monitor is **HutilEISA60Min**. It monitors OID 1.3.6.1.4.1.232.6.2.8.5.0.

60 Minute Load Average

This monitor returns an integer which represents the average load of the system for past sixty minutes. Average load is defined as the number of jobs in the run queue. The CLI name for this monitor is

ResrcCPU60Min. It monitors OID 1.3.6.1.4.1.232.11.2.3.1.1.5.0.

CPU Fan Status

This monitor returns an integer which represents the current status of the processor fan(s) in the system. Possible values are **ok**, **degraded**, **failed**, and **other**. A value of other indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of failed is ever reached. The CLI name for this monitor is **EnvCPUFanStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.5.0.

File Count

This monitor returns an integer which represents the current number of files that exist on the primary filesystem. The CLI name for this monitor is **ResrcFSFilesCnt**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.7.0.

Filesystem % Used

This monitor returns an integer which represents the percentage of disk space used on the primary filesystem. The CLI name for this monitor is **ResrcFSUsedPct.** It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.5.0.

Filesystem Description

This monitor returns a string which contains the type of filesystem being used by the operating system running on the system. The CLI name for this monitor is **OSFilesysDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.2.0.

Filesystem File Count

This monitor returns an integer which indicates the current number of files on the filesystem. The CLI name for this monitor is **OSFilesysNames**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.6.0.

Filesystem Size

This monitor returns an integer which indicates the size (in KB) of the filesystem (on which the OS resides). The CLI name for this monitor is **OSFilesysSpace**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.3.0.

Filesystem Used

This monitor returns an integer which represents the amount of disk space (in MB) used on the primary filesystem. The CLI name for this monitor is **ResrcFSUsedMB**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.4.0.

Floppy Drive Type

This monitor returns an integer which indicates which type of floppy drive the system has. The possible choices are **360K**, **720K**, **1.2Mb**, **1.44Mb**, and **other**. The CLI name for this monitor is **HardwareFpyType**. It monitors OID 1.3.6.1.4.1.232.1.2.11.1.1.2.0.

Hardware Cache Level

This monitor returns an integer indicating the level of the cache, where level 1 is on-board the CPU, etc. The CLI name for this monitor is **HrdwrCacheLvl**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.2.0.2.

Hardware Cache Size

This monitor returns an integer which contains the size (in KB) of the hardware cache. The CLI name for this monitor is **HrdwrCacheSize**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.3.0.2.

Hardware Cache Speed

This monitor returns an integer which contains the speed (in nanoseconds) of the hardware cache. A value of 0 means the cache speed is unknown. The CLI name for this monitor is **HrdwrCacheSpeed**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.4.0.2.

Hardware Cache Status

This monitor returns an integer which contains the status of the cache. The possible values are **unknown**, **ok**, **degraded**, and **failed**. The CLI name for this monitor is **HrdwrCacheStat**. It monitors OID 1.3.6.1.4.1.232.1.2.2.3.1.5.0.2.

Hardware Parallel Port Address

This monitor returns an integer which represents the hardware address of the system Parallel Port. The CLI name for this monitor is **HrdwrParPortAdr**. It monitors OID 1.3.6.1.4.1.232.1.2.10.1.1.2.0.

Hardware Serial Port Address

This monitor returns an integer which represents the hardware address of the system Serial Port. The CLI name for this monitor is **HrdwrSerPortAdr**. It monitors OID 1.3.6.1.4.1.232.1.2.9.1.1.2.0.

Maximum File Count

This monitor returns an integer which represents the maximum number of files that can exist on the primary filesystem. The CLI name for this monitor is **ResrcFSFilesMax**. It monitors OID 1.3.6.1.4.1.232.11.2.4.1.1.6.0.

Operating System Description

This monitor returns a string which contains a description of the operating system running on the system. The CLI name for this monitor is **OSDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.2.3.0.

Operating System Name

This monitor returns a string which contains the name of the operating system running on the system. The CLI name for this monitor is **OSName**. It monitors OID 1.3.6.1.4.1.232.11.2.2.1.0.

Operating System Version

This monitor returns a string which contains the version of the operating system running on the system. The CLI name for this monitor is **OSVers**. It monitors OID 1.3.6.1.4.1.232.11.2.2.2.0.

Overall Thermal Condition

This monitor returns an integer which represents the current overall temperature condition of the system. Possible values are **ok**, **degraded**, **failed**, and **other**. A value of **other** indicates the system does not monitor this indication. The CLI name for this monitor is **EnvTempCond**. It monitors OID 1.3.6.1.4.1.232.6.2.6.1.0.

Physical System Memory

This monitor returns an integer indicating the amount (in KB) of physical memory in the system. The CLI name for this monitor is **HardwareMemory**. It monitors OID 1.3.6.1.4.1.232.1.2.3.2.0.

Processor Name

This monitor returns a string that indicates which CPU chip is present. The CLI name for this monitor is **HardwareCPUName**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.3.0.

Processor Number

This monitor returns an integer that designates a CPU in a MP system. The CLI name for this monitor is **HardwareCPUIdx**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.1.0.

Processor Speed

This monitor returns an integer which indicates the clock speed (in Mhz) of the CPU. The CLI name for this monitor is **HardwareCPUSpeed**. It monitors OID 1.3.6.1.4.1.232.1.2.2.1.1.4.0.

Server Health Uptime

This monitor returns an integer which represents the current number of files that exist on the primary filesystem. The CLI name for this monitor is **HutilPowerOn**. It monitors OID 1.3.6.1.4.1.232.6.2.8.1.0.

SNMP Software Binary

This monitor returns a string that contains the name of the executable which is providing the SNMP agent functionality. The CLI name for this monitor is **SoftwareName**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.4.0.

SNMP Software Description

This monitor returns a string that describes the SNMP agent software. The CLI name for this monitor is **SoftwareDescr**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.5.0.

SNMP Software Location

This monitor returns a string which contains the full path to the executable which is providing the SNMP agent functionality. The CLI name for this monitor is **SoftwareLoc**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.7.0.

SNMP Software Status

This monitor returns an integer which contains the operational status of the SNMP agent software. Possible values are **loaded**, **notloaded**, and **other**. The CLI name for this monitor is **SoftwareStatus**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.2.0.

SNMP Software Type

This monitor returns an integer which indicates what type of software is providing the SNMP agent functionality. Possible values are **driver**, **agent**, **sysutil**, and **other**. The CLI name for this monitor is **SoftwareType**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.3.0.

SNMP Software Version

This monitor returns a string that indicates the version of the SNMP agent software. The CLI name for this monitor is **SoftwareVers**. It monitors OID 1.3.6.1.4.1.232.11.2.7.2.1.8.0.

System Fan Status

This monitor returns an integer which represents the current status of the fan(s) in the system. Possible values are **ok**, **degraded**, **failed**, and **other**. A value of **other** indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of **failed** is ever reached. The CLI name for this monitor is **EnvFanStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.4.0.

System Keyboard

This monitor returns a string that describes the system keyboard. The CLI name for this monitor is **HardwareKbd**. It monitors OID 1.3.6.1.4.1.232.1.2.7.1.0.

System Model

This monitor returns a string that indicates the system model. The CLI name for this monitor is **HardwareProduct**. It monitors OID 1.3.6.1.4.1.232.2.2.4.2.0.

System ROM Version

This monitor returns a string which contains the system ROM identification. The CLI name for this monitor is **HrdwrRomVers**. It monitors OID 1.3.6.1.4.1.232.1.2.6.1.0.

System Serial Number

This monitor returns a string that indicates the system serial number. The CLI name for this monitor is **HardwareSerNo**. It monitors OID 1.3.6.1.4.1.232.2.2.2.1.0.

System Temperature Status

This monitor returns an integer which represents the current overall temperature status of the system. Possible values are **ok**, **degraded**, **failed**, and **other**. A value of **other** indicates the system does not monitor this indication. The system will take action on the status value automatically by shutting down if a value of **failed** is ever reached. The

CLI name for this monitor is **EnvTempStat**. It monitors OID 1.3.6.1.4.1.232.6.2.6.3.0.

Video Controller

This monitor returns a string that describes the system video adaptor. The CLI name for this monitor is **HardwareVideo**. It monitors OID 1.3.6.1.4.1.232.1.2.8.1.0.

10

SNMP MIB-II Monitoring Sources

You can use the SNMP MIB II monitoring collection to monitor SNMP MIB II values from a network interface. SNMP MIB II defines the second version of the Management Information Base (MIB-II) for use with network management protocols in TCP/IP-based internets. This monitoring collection contains a subset of the IETF RFC 1213MIB-II entries. This collection runs on UNIX, Windows NT, Windows 95, and Novell systems that are running SNMP agents.

Note: If you install the TME 10 SNMP monitoring collection from the TME 10 desktop, the TME 10 SNMP monitoring collection, the TME 10 Compaq Insight Manager monitoring collection, and the TME 10 SNMP MIB-II monitoring collection are all installed as part of the TME 10 SNMP monitoring collection installation process. If you install the TME 10 SNMP monitoring collection from the command line, you must install the TME 10 SNMP MIB-II monitoring collection individually from the command line.

This collection does not support the following MIB-II groups:

- Address Translation
- ICMP
- EGP
- Transmission
- SNMP

Agents from these groups can be monitored with the generic SNMP monitoring collection.

Monitoring Source Tables

The following tables list the monitoring sources provided with this monitoring collection. The tables include the name of the monitoring source as displayed in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog, its CLI equivalent, and the OID that it monitors.

System Monitoring Sources

GUI Name	CLI Name	OID
Host Description	HostDescr	1.3.6.1.2.1.1.1.0
Host Up Time	HostUpTime	1.3.6.1.2.1.1.3.0
Host Contact	HostContact	1.3.6.1.2.1.1.4.0
Host Name	HostName	1.3.6.1.2.1.1.5.0
Host Location	HostLocation	1.3.6.1.2.1.1.6.0
Host Services	HostServices	1.3.6.1.2.1.1.7.0

Interface Monitoring Sources

Note: Most of the OID values listed in the following table contain the **$NET** environment variable. By default, this value is **2**. However, certain operating systems and machine types require different values for this environment variable. On HP 9000 Series 700 machines using HP-UX 9.x this value is **4**. On Series 800 machines using HP-UX 9.x, this value is **1**. On all HP machines using HP-UX 10.0, the value is **4**. Finally, on OSF/1 machines, this value is **1**.

GUI Name	CLI Name	OID
Network Interface Count	NetIntfCnt	1.3.6.1.2.1.2.1.0
Network Interface Description	NetIntfDescr	1.3.6.1.2.1.2.2.1.2.$NET
Network Interface Type	NetIntfType	1.3.6.1.2.1.2.2.1.3.$NET
Network Interface MTU	NetIntfMTU	1.3.6.1.2.1.2.2.1.4.$NET
Network Interface Speed	NetIntfSpeed	1.3.6.1.2.1.2.2.1.5.$NET
Network Interface Admin Status	NetIntfAdmnStat	1.3.6.1.2.1.2.2.1.7.$NET
Network Interface Operational Status	NetIntfOperStat	1.3.6.1.2.1.2.2.1.8.$NET
Bytes Received	NetBytesRcvd	1.3.6.1.2.1.2.2.1.10.$NET
Bytes Transmitted	NetBytesXmtd	1.3.6.1.2.1.2.2.1.16.$NET
Broadcast Bytes Received	NetBcstBytesRcvd	1.3.6.1.2.1.2.2.1.12.$NET
Broadcast Bytes Transmitted	NetBcstBytesXmtd	1.3.6.1.2.1.2.2.1.18.$NET
Receive Errors	NetBytesRcvdErr	1.3.6.1.2.1.2.2.1.14.$NET
Transmit Errors	NetBytesXmtdErr	1.3.6.1.2.1.2.2.1.20.$NET
Receives Discarded	NetBytesRcvdDisc	1.3.6.1.2.1.2.2.1.13.$NET
Transmits Discarded	NetBytesXmtdDisc	1.3.6.1.2.1.2.2.1.19.$NET
Transmit Queue Length	NetBytesXmtdQlen	1.3.6.1.2.1.2.2.1.21.$NET

Internet Protocol Monitoring Sources

GUI Name	CLI Name	OID
IP Forwarding Status	IPForwarding	1.3.6.1.2.1.4.1.0
Default IP TTL	IPDefaultTTL	1.3.6.1.2.1.4.2.0
IP Received Packets	IPInRecieves	1.3.6.1.2.1.4.3.0
IP Rcvd Discarded Packets	IPInDiscards	1.3.6.1.2.1.4.8.0
IP Transmit Packets	IPOutRequests	1.3.6.1.2.1.4.10.0
IP Xmit Discarded Packets	IPOutDiscards	1.3.6.1.2.1.4.11.0
IP Transmit No-Route Errors	IPOutNoRoutes	1.3.6.1.2.1.4.12.0

Transmission Control Protocol Monitoring Sources

GUI Name	CLI Name	OID
Maximum TCP Connection	TCPMaxConn	1.3.6.1.2.1.6.4.0
TCP Connection Resets	TCPEstabReset	1.3.6.1.2.1.6.8.0
TCP Current Connections	TCPCurrEstab	1.3.6.1.2.1.6.9.0

User Datagram Protocol Monitoring Sources

GUI Name	CLI Name	OID
UDP No-Port Errors	UDPNoPorts	1.3.6.1.2.1.7.2.0

GUI Name	CLI Name	OID
UDP Receive Errors	UDPInErrors	1.3.6.1.2.1.7.3.0

Some of the OID values listed in this chapter contain the **$NET** environment variable. This variable refers to the index of the primary network interface for a specific SNMP agent. By default, this value is **2**. However, certain operating systems and machine types require different values for this environment variable. On HP 9000 Series 700 machines using HP-UX 9.x, this value is **4**. On Series 800 machines using HP-UX 9.x, this value is **1**. On all HP machines using HP-UX 10.0, the value is **4**. Finally, on OSF/1 machines, this value is **1**.

Broadcast Bytes Received

This monitor returns the total number of broadcasted bytes received by the primary network interface. The CLI name for this monitor is **NetBcstBytesRcvd**. Its OID is 1.3.6.1.2.1.2.2.1.12.$NET.

Broadcast Bytes Transmitted

This monitor returns the total number of bytes broadcasted by the primary network interface. The CLI name for this monitor is **NetBcstBytesXmtd**. Its OID is 1.3.6.1.2.1.2.2.1.18.$NET.

Bytes Received

This monitor returns the total number of bytes received by the primary network interface.The CLI name for this monitor is **NetBytesRcvd**. Its OID is 1.3.6.1.2.1.2.2.1.10.$NET.

Bytes Transmitted

This monitor returns the total number of bytes transmitted by the primary network interface. The CLI name for this monitor is **NetBytesXmtd**. Its OID is 1.3.6.1.2.1.2.2.1.16.$NET.

Default IP TTL

This monitor returns the default TTL (time to live) value for IP datagrams that originate on this host. The CLI name for this monitor is **IPDefaultTTL**. Its OID is 1.3.6.1.2.1.4.2.0.

Host Contact

This monitor returns the person who should be contacted regarding administration of the system. The CLI name for this monitor is **HostContact**. Its OID is 1.3.6.1.2.1.1.4.0.

Host Description

This monitor returns a string that describes the system’s hardware and operating system. The CLI name for this monitor is **HostDescr**. Its OID is 1.3.6.1.2.1.1.1.0.

Host Location

This monitor returns a string representing the physical location of the host. The CLI name for this monitor is **HostLocation**. Its OID is 1.3.6.1.2.1.1.6.0.

Host Name

This monitor returns the name of the system, which may or may not be a fully qualified name. The CLI name for this monitor is **HostName**. Its OID is 1.3.6.1.2.1.1.5.0.

Host Services

This monitor returns an integer which represents the sum of all network services provided by the system. The value is the sum of the following:

Physical	1
Datalink	2
Internet	4
End-to-End	8
Applications	64

The CLI name for this monitor is **HostServices**. Its OID is 1.3.6.1.2.1.1.7.0.

Host Up Time

This monitor returns the number of seconds the SNMP subsystem has been active, If the system is rebooted, this counter is reset to **0**. The CLI name for this monitor is **HostUpTime**. Its OID is 1.3.6.1.2.1.1.3.0.

IP Forwarding Status

This monitor returns an integer indicating that a system is (**1**) or is not (**2**) forwarding IP packets. The CLI name for this monitor is **IPForwarding**. Its OID is 1.3.6.1.2.1.4.1.0.

IP ReceivedDiscarded Packets

This monitor returns the total number of all IP datagrams received and discarded due to lack of input buffer space. The CLI name for this monitor is **IPInDiscards**. Its OID is 1.3.6.1.2.1.4.8.0.

IP Received Packets

This monitor returns the total number of all IP datagrams received (including errors). The CLI name for this monitor is **IPInReceives**. Its OID is 1.3.6.1.2.1.4.3.0.

IP Transmit No-Route Errors

This monitor returns the total number of all output IP datagrams that were discarded because no route could be found to the destination. The CLI name for this monitor is **IPOutNoRoutes**. Its OID is 1.3.6.1.2.1.4.12.0.

IP Transmit Packets

This monitor returns the total number of IP datagrams that originated on this system. The CLI name for this monitor is **IPOutRequests**. Its OID is 1.3.6.1.2.1.4.10.0.

IP Transmit Discarded Packets

This monitor returns the total number of all output IP datagrams discarded due to lack of output buffer space. The CLI name for this monitor is **IPOutDiscards**. Its OID is 1.3.6.1.2.1.4.11.0.

Network Interface Admin Status

This monitor returns the desired administrative status of the primary network interface. One of the following values will be returned: **up**, **down**, or **testing**. The CLI name for this monitor is **NetIntfAdmnStat**. Its OID is 1.3.6.1.2.1.2.2.1.7.$NET.

Network Interface Count

This monitor returns an integer which represents the number of network interfaces a system has. The CLI name for this monitor is **NetIntfCnt**. Its OID is 1.3.6.1.2.1.2.1.0.

Network Interface Description

This monitor returns a description of the primary network interface on a host. The SNMP MIB will return the description of any of the network interfaces, and the default selection of the primary interface can be overridden by modification of the **rfc1213.OID** file or by creating a user-defined SNMP monitor. The CLI name for this monitor is **NetIntfDescr** Its OID is 1.3.6.1.2.1.2.2.1.2.$NET.

Network Interface MTU

This monitor returns the MTU (size of largest possible transmitted packet) for the primary network interface. The CLI name for this monitor is **NetIntfMTU**. Its OID is 1.3.6.1.2.1.2.2.1.4.$NET.

Network Interface Operational Status

This monitor returns the operational status of the network interface. A value of **up**, **down**, or **testing** will be returned. The CLI name for this monitor is **NetIntfOperStat**. Its OID is 1.3.6.1.2.1.2.2.1.8.$NET.

Network Interface Speed

This monitor returns the maximum transmission rate for the primary network interface on the system. The CLI name for this monitor is **NetIntfSpeed**. Its OID is 1.3.6.1.2.1.2.2.1.5.$NET.

Network Interface Type

This monitor returns an integer which represents the type of the primary network interface for a system. The following table lists the possible return values:

Type	Value
other	1
regular	2
ddn-x25	4
rfc877-x25	5
ethernet	6
802.3	7
802.4	8
802.5	9
802.6	10
starLan	11
fddi	15
lapb	16
sdlc	17
ds1	18

Type	Value
e1	19
basic ISDN	20
primISDN	21
Prop PPP	22
ppp	23
loopback	24

The CLI name for this monitor is **NetIntfType**. Its OID is 1.3.6.1.2.1.2.2.1.3.$NET.

Receives Discarded

This monitor returns the total number of packets which were received and discarded due to a problem on the system (most likely lack of network buffer space). The CLI name for this monitor is **NetBytesRcvdDisc**. Its OID is 1.3.6.1.2.1.2.2.1.13.$NET.

Receive Errors

This monitor returns the total number of packets which were received and discarded because the packet contained an error. The CLI name for this monitor is **NetBytesRcvdErr**. Its OID is 1.3.6.1.2.1.2.2.1.14.$NET.

Transmits Discarded

This monitor returns the total number of output packets which were not sent due to a problem on the system (most likely a lack of network buffer space). The CLI name for this monitor is **NetBytesXmtdDisc**. Its OID is 1.3.6.1.2.1.2.2.1.19.$NET.

Transmit Errors

This monitor returns the total number of output packets which were not sent because the packet contained an error. The CLI name for this monitor is **NetBytesXmtdErr**. Its OID is 1.3.6.1.2.1.2.2.1.20.$NET.

Transmit Queue Length

This monitor returns the number of output packets waiting to be transmitted. The CLI name for this monitor is **NetBytesXmtdQlen**. Its OID is 1.3.6.1.2.1.2.2.1.21.$NET.

Maximum TCP Connection

This monitor returns the maximum number of simultaneous TCP connections for the system. The CLI name for this monitor is **TCPMaxConn**. Its OID is 1.3.6.1.2.1.6.4.0.

TCP Connection Resets

This monitor returns the number of TCP connections that have been reset. The CLI name for this monitor is **TCPEstabReset**. Its OID is 1.3.6.1.2.1.6.8.0.

TCP Current Connections

This monitor returns the number of current TCP connections. The CLI name for this monitor is **TCPCurrEstab**. Its OID is 1.3.6.1.2.1.6.9.0.

UDP No-Port Errors

This monitor returns the total number of UDP input packets for which there was no listening process on the system. The CLI name for this monitor is **UDPNoPorts**. Its OID is 1.3.6.1.2.1.7.2.0.

UDP Receive Errors

This monitor returns the total number of UDP input packets that were not delivered even though there was a listening process on the system. The CLI name for this monitor is **UDPInErrors**. Its OID is

1.3.6.1.2.1.7.3.0.

11

User-Defined SNMP Monitoring Sources

The generic SNMP monitoring collection comes with the **User Defined Numeric** and **User Defined String** monitoring sources. These monitoring sources allow you to monitor the information returned from any SNMP agent request.

You must specify four arguments to configure a user-defined monitoring source:

system Specifies the name of the host to run the monitor on. The default value from the GUI is **localhost**. You must specify a value if you are using the monitor from the CLI.

seconds Specifies the number of seconds to wait for results. The default value from the GUI is **5**. You must specify a value if you are using the monitor from the CLI.

name Specifies the SNMP community name. The default value from the GUI is **public**. You must specify a value if you are using the monitor from the CLI.

MIB Specifies the MIB number.

The CLI name for this collection is **UserSNMP**.

User Defined Numeric

The **User Defined Numeric** monitoring source executes a user-supplied script or program and interprets its output numerically. The program should print its result as a numeric value to standard output. Numeric means that the output of the script is interpreted as a number (possibly a real number, such as **12.5** or **0.731**). All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

The CLI name for this monitor is **UserSNMPNumeric**.

User Defined String

The **User Defined String** monitoring source executes a user-supplied script or program and interprets its output as a string. The program should print its result data to standard output. All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

The CLI name for this monitor is **UserSNMPString**.

12

TME 10 Monitoring Sources

This monitoring collection allows you to use TME 10 Distributed Monitoring to monitor your TME 10 installation.

To reduce system overhead, the daemon status monitor maintains a snapshot of all currently running processes, rather than continually checking for running processes.

TME 10 Monitoring Collection

The following table lists the monitoring sources that are provided by this collection. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Monitoring Collection	CLI Name
Number of Oserv Restarts	OservRestarts
Number of Oserv Errors	OservErrors
Tivoli DB Directory Free Space	DBdiskusagepctMB
Object Calls Made	ObjCallsmade
Object Calls Failed	ObjCallsFailed

Monitoring Collection	CLI Name
Local Methods Invoked	LocalMethodscalled
Remote Methods Invoked	RemoteMethodscalled
Data blocks in	OservBlocksIn
Data blocks Out	OservBlocksOut
Data Blocks In/Out	OservBlocksInOut
Size of OservLog	OservLogSize

Number of Oserv Restarts

This monitor returns the number of times that the oserv daemon has been restarted. The CLI name for this monitor is **OservRestarts**.

Number of Oserv Errors

This monitor returns the number of oserv errors. The CLI name for this monitor is **OservErrors**.

Tivoli DB Directory Free Space

This monitor returns the amount of free space remaining (in MB) in the TME 10 database directory. The CLI name for this monitor is **DBdiskusagepctMB**.

Object Calls Made

This monitor returns the number of object calls made. **ObjCallsmade**.

Object Calls Failed

This monitor returns the number of object call failures. The CLI name for this monitor is **ObjCallsFailed**.

Local Methods Invoked

This monitor returns the number of local methods that have been invoked. The CLI name for this monitor is **LocalMethodscalled**.

Remote Methods Invoked

This monitor returns the number of remote methods that have been invoked. The CLI name for this monitor is **RemoteMethodscalled**.

Data blocks in

This method returns the number of data blocks that have been received by the oserv daemon. The CLI name for this monitor is **OservBlocksIn**.

Data blocks Out

This monitor returns the number of data blocks that the oserv daemon has sent. The CLI name for this monitor is **OservBlocksOut**.

Data Blocks In/Out

This monitor returns the total number of data blocks that have been sent and received by the oserv daemon. The CLI name for this monitor is **OservBlocksInOut**.

Size of OservLog

This monitor returns the current size (in MB) of the oserv log. The CLI name for this monitor is **OservLogSize**.

13

TME 10 Enterprise Console Monitoring Sources

This monitoring collection allows you to use TME 10 Distributed Monitoring to monitor the TME 10 Enterprise Console.

TME 10 Enterprise Console Monitoring Collection

The following table lists the monitoring sources provided by this collection. The first column lists the names of the sources as they appear in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. The second column provides a name that enables you to use a script to configure your TME 10 Distributed Monitoring profiles.

Note: With the exception of the **TEC Server Status** and **TEC Database Status** monitors, the monitors in this collection can only be run on TME 10 Enterprise Console 2.6.

Monitoring Collection	CLI Name
TEC Server Status	ServerStatus_26
TEC DataBase Status	DataBaseStatus_26
TEC DB Space Free	DBspaceFree_26
TEC DB Space Used	DBspaceUsed_26

Monitoring Collection	CLI Name
TEC DB Space Percentage Used	DBspaceUsedpct_26
TEC DB Space Percentage Free	DBspaceFreepct_26
TEC Log Space Free	LogspaceFree_26
TEC Log Space Used	LogspaceUsed_26
TEC Log Space Percentage Used	LogspaceUsedpct_26
TEC Log Space Percentage Free	LogspaceFreepct_26
TEC Event Space Free	EventspaceFree_26
TEC Event Space Used	EventspaceUsed_26
TEC Event Space Percentage Used	EventspaceUsedpct_26
TEC Event Space Percentage Free	EventspaceFreepct_26

You must specify a size for the TME 10 Enterprise Console database and the TME 10 Enterprise Console event space when the TME 10 Enterprise Console is installed, but this only specifies an initial size, not an absolute limit. If there is space available and the TME 10 Enterprise Console needs additional space for its database or event space, the TME 10 Enterprise Console allocates additional space. Therefore, the TME 10 Enterprise Console monitoring collection may report the percentage of space used as greater than 100 percent.

For example, if the TME 10 Enterprise Console database is set to an initial size of 30 megabytes, but the TME 10 Enterprise Console then expands the database to 45 megabytes, the **TEC DB Space Percentage Used** monitor will report the database space used as 150 percent.

TEC Server Status

This monitor returns the status of the TME 10 Enterprise Console server. The CLI name for this monitor is **ServerStatus_26**.

TEC Database Status

This monitor returns the status of the TME 10 Enterprise Console database. The CLI name for this monitor is **DataBaseStatus_26**.

TEC DB Space Free

This monitor returns the amount of free space (in MB) available in the TME 10 Enterprise Console database. The CLI name for this monitor is **DBspaceFree_26**.

TEC DB Space Used

This monitor returns the amount of space used (in MB) by the TME 10 Enterprise Console database. The CLI name for this monitor is **DBspaceUsed_26**.

TEC DB Space Percentage Used

This monitor returns the amount of space used, as a percentage of the total space allocated, by the TME 10 Enterprise Console database. The CLI name for this monitor is **DBSpaceUsedpct_26**.

TEC DB Space Percentage Free

This monitor returns the amount of free space, as a percentage of the total space allocated, that is available in the TME 10 Enterprise Console database. The CLI name for this monitor is **DBSpaceFreepct_26**.

TEC Log Space Free

This monitor returns the amount of free space (in MB) that is available in the TME 10 Enterprise Console log. The CLI name for this monitor is **LogspaceFree_26**.

TEC Log Space Used

This monitor returns the amount of space used (in MB) by the TME 10 Enterprise Console log. The CLI name for this monitor is **LogspaceUsed_26**.

TEC Log Space Percentage Used

This monitor returns the amount of space used, as a percentage of the total space allocated, by the TME 10 Enterprise Console log.The CLI name for this monitor is **LogspaceUsedpct_26**.

TEC Log Space Percentage Free

This monitor returns the amount of free space, as a percentage of the total space allocated, that is available in the TME 10 Enterprise Console log. The CLI name for this monitor is **LogsapceFreepct_26**.

TEC Event Space Free

This monitor returns the amount of free space (in MB) that is available for TME 10 Enterprise Console events. The CLI name for this monitor is **EventspaceFree_26**.

TEC Event Space Used

This monitor returns the amount of space used (in MB) by TME 10 Enterprise Console events. The CLI name for this monitor is **EventspaceUsed_26**.

TEC Event Space Percentage Used

This monitor returns the amount of space used, as a percentage of the total space allocated, by TME 10 Enterprise Console events. The CLI name for this monitor is **EventspaceUsedpct_26**.

TEC Event Space Percentage Free

This monitor returns the amount of free space, as a percentage of the total space allocated, that is available for TME 10 Enterprise Console events. The CLI name for this monitor is **EventspaceFreepct_26**.

14

UNIX Monitoring Collection

The UNIX monitoring collection provides monitoring sources that enable you to use TME 10 Distributed Monitoring to manage distributed computing resources effectively. Each monitoring source enables you to manage a different aspect of the operating system, such as daemons or other processes, disk space, and network collisions. You can also use the asynchronous and user-defined monitoring sources to create your own powerful monitoring sources.

The following tables list the monitoring sources that are provided with this monitoring collection. Some monitoring sources are listed in multiple tables.

Disk Resource Monitoring Sources

GUI Name	CLI Name	Described on Page
Inodes free	inodes	14-24
Inodes used	inodesused	14-26
Percent inodes used	inodesusedpct	14-55
Percent space used	diskusedpct	14-57

GUI Name	CLI Name	Described on Page
Space free	diskavail	14-65
Space used	diskused	14-67
Tivoli DB free space	tivdbspace	14-72

Security Monitoring Sources

GUI Name	CLI Name	Described on Page
Check file permissions	fileperm	14-10
Compare files	filediff	14-14
Daemon status	daemon	14-17
File checksum	filechk	14-19
File size	filesize	14-20
Occurrences in file	countstr	14-47
Process instances	daemonct	14-59
User logins by user	ulogins	14-76
Users logged in	ulogintot	14-78

Network Monitoring Sources

GUI Name	CLI Name	Described on Page
Client RPC timeouts	rpctmout	14-12
Host status	host	14-22
Network collisions	netcoll	14-40
Network collisions/packet	netcollpct	14-42
NFS bad calls	badnfs	14-44
Input packet errors	netinerr	14-28
Input packets	netin	14-30
Output packet errors	netouterr	14-49
Output packets	netout	14-51
Remote oserv status	oserv	14-61
RPC bad calls	badrpc	14-63

System Resources Monitoring Sources

GUI Name	CLI Name	Described on Page
Available swap space	swapavail	14-8
Host status	host	14-22
Lingering terminated processes	zombies	14-34

GUI Name	CLI Name	Described on Page
Load average	loadavg	14-36
Mail queue length	mailqlen	14-38
Page-outs	pageouts	14-53

Printer Monitoring Sources

GUI Name	CLI Name	Described on Page
Daemon status	daemon	14-17
Jobs in print queue	printjobs	14-32
Status of print queue	printstat	14-69
Total size queued	printjobsize	14-74

User-Defined Monitoring Sources

GUI Name	CLI Name	Described on Page
Asynchronous numeric	nasync	14-5
Asynchronous string	sasync	14-7
Numeric script	ncustom	14-46
String script	scustom	14-71

NAME

Asynchronous numeric

Purpose

Responds to events sent to the TME 10 Distributed Monitoring engine tagged with the given channel name.

Partial CLI Synopsis

nasync -a *channel_name*

Description

The **Asynchronous numeric** monitoring source indicates a channel to monitor for messages.

Channel names are arbitrary and are completely under user control. The relationship between a criterion monitoring a channel of a given name and specific events generated for that channel is only the name. Names only need to be registered with the **waddchan** command if it is desired that they appear in the **Choices** pop-up dialog.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

channel_name Specifies the name of the channel to monitor.

Examples

A job has been created with the Task Library to search certain software build directories for object files. The task can be enhanced so that once the object file count (or a total of the file sizes) is computed, the numeric result is forwarded via a **wasync** command to the local TME 10 Distributed Monitoring engine on each managed node subscribed to the job.

NAME

Asynchronous string

Purpose

Responds to events sent to the TME 10 Distributed Monitoring engine tagged with the given channel name.

Partial CLI Synopsis

sasync -a *channel_name*

Description

The **Asynchronous string** monitoring source indicates a channel to monitor for events.

Channel names are arbitrary and are completely under user control. The relationship between a criterion monitoring a channel of a given name and specific events generated for that channel is only the name. Names only need to be “registered” with the **waddchan** command if you want them to appear in the **Choices** pop-up dialog.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use string operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

channel_name Specifies the name of the channel to monitor.

NAME

Available swap space

Purpose

Monitors the amount of available swap space.

Partial CLI Synopsis

swapavail

Description

The **Available swap space** monitoring source monitors the amount of swap space, in megabytes, each subscriber is using. Swap space is usually a disk partition on which page-outs are written.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Less than 10 MB	Send TME 10 notice. Popup alarm. Change icon.
severe	Less than 15 MB	Send TME 10 notice. Change icon.
warning	Less than 20 MB	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

It is generally recommend that you should have at least twice as much swap space as memory. If an available swap space monitor frequently triggers, you may want adjust your swap space accordingly.

NAME

Check file permissions

Purpose

Checks the permissions of a file.

Partial CLI Synopsis

fileperm -a *filename*

Description

The **Check file permissions** monitoring source returns the permissions string from a file and allows string comparisons to be made on the value. You must specify a search pattern that is capable of matching, either exactly or by using regular expressions, the permissions displayed when you execute the **ls -l** command. For example, the permissions for the **/home/report.doc** file might be the following:

```
-rw-r--r--
```

Therefore, to determine whether someone has changed the permissions so that other members of user-group has write access, you could enter one of the following strings (among others):

```
-rw-rw-r--
```

—OR—

```
-rw-rw*
```

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	none
severe	never	none
warning	never	none
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use string operators to evaluate data. You can use the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Note: If you want to use wildcard characters in your file permission string, use the **Matches** operator.

Argument

filename Specifies the name of the file to monitor.

Examples

To determine when the write permissions of the **/etc/passwd** file have changed from **-rw-r--r--**, set up a monitor so that the search string is **-rw-*w***.

NAME

Client RPC timeouts

Purpose

Monitors the number of client RPC requests that have timed out

Partial CLI Synopsis

rpctmout

Description

The **Client RPC timeouts** monitoring source monitors the number of client RPC requests that have timed out since the last time the statistics were reset. If this monitor triggers frequently, you may have network configuration problems.

Note: You must use the **nfsstat** command to reset statistics.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

To help get a snapshot of your network when a client RPC timeout monitor triggers, you can create a script that uses the **nfsstat** command to return information about the server and clients on your network. You can use this information to help debug your system.

NAME

Compare files

Purpose

Compares two files.

Partial CLI Synopsis

filediff -a *filename* **-a** *filename* **-a** *options*

Description

The **Compare files** monitoring source compares two text or binary files. An action can be triggered when files are the same, when different, become the same, or when identical files become different.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use file operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

filename Specifies the name of a file to monitor.

options Specifies one of the following options:

Plain difference
Indicates the comparison will include checks for differences in whitespace and capitalization. This is the default value. Counts the number of bytes in a file. Use **“”** to specify this option from the command line.

Ignore whitespace
Indicates the comparison will ignore differences in spacing. Use **“-bw”** to specify this option from the command line.

Ignore alphabetic case
Indicates the search will not distinguish between uppercase and lowercase characters. Use **“-i”** to specify this option from the command line.

Ignore case & whitespace
Indicates the search will be case-insensitive and ignore spacing differences. Use **“-ibw”** to specify this option from the command line.

Binary Indicates the monitor will compare two binary files. Use **“--bin--”** to specify this option from the command line.

Examples

The **/etc/alias** file defines which mail aliases a user belongs to. To help prevent an ordinary user from receiving mail meant for managers only, you could keep a copy of the **/etc/aliases** file in another directory and use it as a basis for comparison. If unauthorized changes were made to the **/etc/aliases** file, the copy can be used to restore the file to its original condition.

NAME

Daemon status

Purpose

Monitors the availability of a daemon or process.

Partial CLI Synopsis

daemon -a *daemon*

Description

The **Daemon status** monitoring source monitors the availability of a specified daemon or process. For instance, you could monitor for the absence of the **inetd** daemon, and use the **Run program** option to call a script that restarts the daemon.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Becomes unavailable	Send TME 10 notice. Popup alarm. Change icon.
severe	never	none
warning	never	none
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

daemon

Specifies the name of the daemon or process to monitor. You can chose the following daemons from a scrolling list, or enter a a different daemon or process.

- amd (automount)
- biod
- cron
- inetd
- lockd
- lpd
- mountd
- nfsd
- portmap
- snmpd
- statd

Note: Do not specify the absolute path for any of these daemons. Other daemons or processes may require the full path name.

Examples

The TME 10 Enterprise Console uses a a relational database management system (RDBMS) to keep track of system events. If the daemon that controls the RDBMS becomes unavailable, incoming events cannot be written to the database. To help minimize the effects of this problem, you could create a monitor that restarts the RDBMS daemon whenever it goes down.

NAME

File checksum

Purpose

Computes the checksum of a specified file.

Partial CLI Synopsis

filechk -a *filename*

Description

The **File checksum** monitoring source computes the 16-bit checksum of a specified file. A checksum is typically used to determine whether a file is an exact duplicate of another file or to watch for a change in a file.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename	Specifies the name of the file to monitor.

Examples

Suppose the **/etc/passwd** file on your system has a checksum of **26289**. With that knowledge, you could set a critical response to trigger when the checksum of the **/etc/passwd** file changes. If an authorized change has been changed, use the **sum** command to obtain a new checksum.

NAME

File size

Purpose

Computes the size of a specified file.

Partial CLI Synopsis

filesize -a *filename* **-a** *size_units*

Description

The **File size** monitoring source computes the size of a specified file. The size can be determined in bytes, words, or lines. This source is especially helpful for tracking the size of log files or other files that are appended frequently.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

filename Specifies the name of the file to monitor.

size_units Specifies one of the following options:

Bytes Counts the number of bytes in a file. Use **“-c”** to specify this option from the command line.

Words Counts the number of words in a file. Use **“-w”** to specify this option from the command line.

Lines Counts the number of lines in a file. Use **“-l”** to specify this option from the command line.

Examples

On the SunOS platform, the syslogd daemon writes information to /var/adm/messages. To use the file size monitoring source effectively, you could write a script that compresses and archives the messages file, then deletes the original file. Once this script is written, you could set a monitor to trigger a critical response when the size of the messages file exceeds 20,000 bytes. The critical responses could be to send a TME 10 notice, display a pop-up alarm, change icon, and run your script.

NAME

Host status

Purpose

Monitors the status of a host.

Partial CLI Synopsis

host -a *hostname*

Description

The **Host Status** monitoring source checks the status of any host or other entity on the network that can respond to a **ping** request.

Note: Depending on network topology, monitoring hosts may be inherently problematic. For example, if a host that is subscribed to a host status monitor is separated from the monitored host by a router, the failure of the router is interpreted as the failure of the host. Use this monitor with restraint. It is generally a good idea to create profiles with several host status entries and then subscribe only a few hosts to these profiles. Also, you should consider whether a host really needs to be monitored. For example, you may want to monitor only those hosts that act as servers.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Becomes unavailable	Send TME 10 notice. Popup alarm. Change icon.
severe	never	none
warning	never	none
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Argument

hostname Specifies the name of the host to monitor.

Examples

If a key file server becomes unavailable, many employees may not be able to perform their duties. Therefore, you may want to be very certain that the proper administrator is notified immediately by sending mail, displaying a pop-up window, and running a script that pages the administrator.

NAME

Inodes free

Purpose

Checks the number of free inodes on a filesystem.

Partial CLI Synopsis

inodes -a *filename*

Description

The **Inodes free** monitoring source monitors the number of inodes free on a specified filesystem. Each file requires an inode; therefore, the number of inodes available defines how many more files and directories can be created on the filesystem. You can monitor any available filesystem except NFS-mounted filesystems. Most platforms do not report available inodes for NFS-mounted filesystems.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Less than 100 inodes	Send TME 10 notice. Popup alarm. Change icon.
severe	Less than 150 inodes	Send TME 10 notice. Change icon.
warning	Less than 200 inodes	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

It is possible for an application to write many small, temporary files to the **/tmp** directory, but not delete them. If the maximum number of inodes is reached within the **/tmp** filesystem, no more temporary files can be written, even if there is plenty of remaining disk space. You could therefore create a monitor that deletes **/tmp** files older than one day whenever the **critical** severity is reached.

NAME

Inodes used

Purpose

Monitors a filesystem’s inode use.

Partial CLI Synopsis

inodesused -a *filename*

Description

The **Inodes used** monitoring source monitors the number of inodes present on a specified filesystem. Each file requires an inode; therefore, the number of inodes available defines how many more files and directories have been created on the filesystem. You can monitor any available filesystem except NFS-mounted filesystems. Most platforms do not report available inodes for NFS-mounted filesystems.

Because a filesystem can have any maximum size, you must explicitly specify the sizes at which a monitor will trigger.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	Send TME 10 notice. Popup alarm. Change icon.
severe	never	Send TME 10 notice. Change icon.
warning	never	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

It is possible for an application to write many small, temporary files to the **/tmp** directory but not delete them. If the maximum number of inodes is reached within the **/tmp** filesystem, no more temporary files can be written, even if there is plenty of remaining disk space. You could therefore create a monitor that deletes **/tmp** files older than one week whenever the **critical** severity is reached.

NAME

Input packet errors

Purpose

Monitors the number of input packet errors.

Partial CLI Synopsis

netinerr

Description

The **Input packet errors** monitoring source monitors the number of input errors on each TCP/IP interface of a host since it was last booted.

The number of error packets should be less than 2.5 percent of the total number of incoming packets. A high number of packet errors can indicate a hardware problem, such as a bad cable or transceiver.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If you monitor input packet errors, you may also want to monitor output packet errors and network collisions. If these monitors frequently trigger even though your ethernet connections are configured properly, you may need to replace a transceiver or a similar piece of equipment.

NAME

Input packets

Purpose

Monitors the total number of packets received.

Partial CLI Synopsis

netin

Description

The **Input packets** monitoring source monitors the number of packets that have arrived at each TCP/IP interface of a host since it was last booted.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If you monitor input packets, you may also want to monitor output packets and network collisions. If these monitors frequently trigger even though your ethernet connections are configured properly, you may need to replace a transceiver or a similar piece of equipment.

NAME

Jobs in print queue

Purpose

Monitors the number of jobs in a selected printer queue.

Partial CLI Synopsis

printjobs -a *print_queue*

Description

The **Jobs in print queue** monitoring source monitors the number of jobs in a printer queue. For example, you can use this activity to determine whether you are using your printer resources effectively.

The exact upper threshold depends on the speed of the printer and the size of the jobs being printed. It is recommended that you set the upper threshold to four or five for a slow printer with medium-size jobs. You will need to experiment to find the correct threshold for your system.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	Send TME 10 notice. Popup alarm. Change icon.
severe	never	Send TME 10 notice. Change icon.
warning	never	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

print_queue Specifies the name of the printer queue to monitor.

Examples

To help prevent an over-burdened printer from receiving even more print jobs, set up a monitor so that it broadcasts a message stating that the printer is busy. You can create a shell script for each printer so that the message is customized.

NAME

Lingering terminated processes

Purpose

Counts zombie processes.

Partial CLI Synopsis

zombies

Description

The **Lingering terminated processes** monitoring source counts the number of processes which have been terminated but have not been cleaned up by their parent process. (Processes in this condition are frequently called ***zombie*** processes.) A large number of such processes can indicate an errant network daemon (such as **rsh**) or application.

The names of the zombie processes are listed in pop-up alarms and TME 10 notices. Use the **ps** command to learn more about these processes.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	greater than 30	Send TME 10 notice. Popup alarm. Change icon.
severe	greater than 20	Send TME 10 notice. Change icon.
warning	greater than 10	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

To help free system resources, you could write a script that uses the **ps** command to determine the owner of a zombie process. If the owner is not logged in or if there is an equivalent active process, the script could **kill** the zombie process. This script could be run when a **critical** event is triggered.

NAME

Load average

Purpose

Monitors the average number of active processes.

Partial CLI Synopsis

loadavg

Description

The **Load average** monitoring source monitors the average number of processes that were active for the last five minutes.

The definition of a high average load varies depending upon the system. A single-user workstation should have a low average load (1 or 2), while an average of 10 is not uncommon for multi-user hosts. Results of this monitor are platform-dependent, particularly when used on multi-user hosts.

NFS input/output activities waiting for completion are included in the load average. If your TME installation includes an NFS file server, the average load may be abnormally high, and could cause TME 10 Distributed Monitoring to generate an alarm.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Greater than 3 ready jobs	Send TME 10 notice. Popup alarm. Change icon.
severe	Greater than 2 ready jobs	Send TME 10 notice. Change icon.
warning	Greater than 1 ready job	Change icon.

Severity	Trigger When	Default Actions
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If the load average of a server reaches the **critical** state, you could send a mail message to all requesting that users use other system resources or cancel non-essential processes.

NAME

Mail queue length

Purpose

Monitors the number of entries in the outgoing mail queue.

Partial CLI Synopsis

mailqlen

Description

The **Mail queue length** monioring source monitors the number of entries in the outgoing mail queue. If this monitor triggers, your mail system may be overloaded. As a result, you should not use mail to perform a triggered response.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Greater than 50 entries	Send TME 10 notice. Popup alarm. Change icon.
severe	Greater than 40 entries	Send TME 10 notice. Change icon.
warning	Greater than 20 entries	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

An unusually large mail queue could indicate a problem with the mail gateway. Therefore, you may need to inspect the gateway for hardware problems. You could run a script that pages a system administrator if this monitor triggers.

NAME

Network collisions

Purpose

Monitors the number of detected network collisions.

Partial CLI Synopsis

netcoll

Description

The **Network collisions** monitoring source monitors the number of network collisions that have been detected at each TCP/IP interface on a subscriber since the host was booted. A collision occurs when two hosts on a network attempt to transmit packets simultaneously. The number of collisions is an indicator of network traffic.

If the number of collisions is consistently high, your network is probably overloaded. To reduce the number of collisions, it is recommended that you consider reorganizing your network.

Note: This data is not available on AT&T platforms.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	25% increase in collisions	Send TME 10 notice. Popup alarm. Change icon.

Severity	Trigger When	Default Actions
severe	10% increase in collisions	Send TME 10 notice. Change icon.
warning	5% increase in collisions	Change icon.
normal	N/A	none
always	N/A	none

Examples

If a new workstation is added to a network, or if the network subnet arrangement is changed, then network collisions should be monitored to verify that the new configuration is not causing problems.

NAME

Network collisions/packet

Purpose

Monitors the ratio of network collisions detected per output packet.

Partial CLI Synopsis

netcollpct

Description

The **Network collisions/packet** monitoring source computes the ratio of network collisions detected per output packet at each TCP/IP interface on a subscriber since the host was booted. A collision occurs when two hosts on a network attempt to transmit packets simultaneously. The number of collisions is an indicator of network traffic. The result is expressed as a percentage value between 0 and 100.

If the number of collisions per packet is consistently high, your network is probably overloaded. It is recommended that you consider reorganizing your network to reduce the number of collisions.

Note: This data is not available on AT&T platforms.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Greater than 25%	Send TME 10 notice. Popup alarm. Change icon.
severe	Greater than 10%	Send TME 10 notice. Change icon.
warning	Greater than 5%	Change icon.
normal	N/A	none

Severity	Trigger When	Default Actions
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If a new workstation is added to a network, or if the network subnet arrangement is changed, then network collisions should be monitored to verify that the new configuration is not causing problems.

NAME

NFS bad calls

Purpose

Monitors the number of rejected NFS requests.

Partial CLI Synopsis

badnfs

Description

The **NFS bad calls** monitoring source monitors the number of NFS bad calls received since the last time the statistics were reset. The number of NFS bad calls is the sum of the number of calls that were smaller than the minimum size established for a RPC request and the number of calls that could not be decoded by XDR.

Numerous NFS bad calls is an indication that UDP packets are getting corrupted in the path between the server and client machines. This can be caused by hardware problems with the network, such as bad repeaters or cabling, an ailing bridge or router corrupting packets, high error rates on leased circuits (such as X.25 or T1), or network software problems on the server or client machines.

Note: You must use the **nfsstat** command to reset statistics.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.

Severity	Trigger When	Default Actions
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

To help get a snapshot of your network when a NFS bad call monitor triggers, you can create a script that uses the **nfsstat** command to return information about the server and clients on your network and write this information to a file. You can also include the **showmount** command in the script to determine which clients are mounted to a server. You can use this information to help debug your system.

NAME

Numeric script

Purpose

Runs a user-defined script that returns a numeric value and is compatible with TME 10 Distributed Monitoring.

Partial CLI Synopsis

ncustom -a *program*

Description

The **Numeric script** monitoring source executes a user-supplied script or program and interprets its output numerically. The program should print its result as a numeric value to standard output. Numeric means that the output of the script is interpreted as a number (possibly a real number, such as **12.5** or **0.731**).

All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename	Specifies the name of the script to run.

Examples

To help prevent files from accumulating in the **/tmp** directory, you could write a script that executes the **ls** command and counts the entries. If there are 50 or more files in the directory, the script could delete these files.

NAME

Occurrences in file

Purpose

Counts occurrences of a pattern in a file.

Partial CLI Synopsis

countstr -a *pattern* **-a** *filename* [**-a** *search_option*]

Description

The **Occurrences in file** monitoring source searches for a pattern in a file and reports the number of times the pattern was found. The arguments to this monitoring source consists of the pattern, the name of the file to be searched, and search options.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Arguments

pattern Specifies a string or regular expression to search for.

filename Specifies the name of the file to monitor.

search_option Specifies one of the following options:

Normal search Indicates that no special search conditions are required. This is the default value. Use **""** to specify this option from the command line.

Ignore alphabetic case
Indicates the search will not distinguish between uppercase and lowercase characters. Use “**-i**” to specify this option from the command line.

Inverse search Indicates that the search results will contain only those lines that do not contain the target pattern. Use “**-v**” to specify this option from the command line.

Match whole words
Indicates the pattern matching will be done against whole words from the lines of the file Use “**-w**” to specify this option from the command line. This option has no effect on some platforms due to the lack of direct support from the **grep** utility.

Examples

User **jsmith** has been denied **root** user privileges because he has misused these privileges in the past. To help minimize the damages if he somehow learns his root password, you could search for the pattern **‘su root’ succeeded** in the **/var/adm/messages** file on **jsmith**’s host.

NAME

Output packet errors

Purpose

Monitors the number of output packet errors.

Partial CLI Synopsis

netouterr

Description

The **Output packet errors** monitoring source monitors the number of output errors on each TCP/IP interface of a host since it was booted.

It is recommended that this number be less that 2.5 percent of the total number of outgoing packets. A high number of outgoing packets errors can be caused by a faulty controller or by a problem with the network cable, such as a bad cable or transceiver.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If you monitor output packet errors, you may also want to monitor input packet errors and network collisions. If these monitors frequently trigger even though your ethernet connections are configured properly, you may need to replace a transceiver or a similar piece of equipment.

NAME

Output packets

Purpose

Monitors the number of transmitted packets.

Partial CLI Synopsis

netout

Description

The **Output packets** monitoring source monitors the amount of outgoing traffic on each TCP/IP interface of a host since it was booted.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

If you monitor output packets, you may also want to monitor input packets and network collisions. If these monitors frequently trigger even though your ethernet connections are configured properly, you may need to replace a transceiver or a similar piece of equipment.

NAME

Page-outs

Purpose

Monitors pages written out by the virtual memory manager.

Partial CLI Synopsis

pageouts

Description

The **Page-outs** monitors the average number of pages that the virtual memory manager pages out per second on each subscriber. Paging is the method of writing portions of a process’s memory to disk to free up memory.

If the number of page-outs is significantly greater than zero for several monitoring intervals, the host could have a memory problem. You will need to experiment with different thresholds to determine the right values for your system.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	100% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	80% increase	Send TME 10 notice. Change icon.
warning	50% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

You can run this monitor in conjunction with the **Load Average** monitoring source to determine whether an application server is overloaded. If the server is overloaded, you may want to reallocate system resources.

NAME

Percent inodes used

Purpose

Monitors the percentage of available inodes in use.

Partial CLI Synopsis

inodesusedpct -a *filename*

Description

The **Percent inodes used** monitoring source monitors a specified filesystem’s inode allocation as a percentage of the filesystem’s total number of inodes. You can monitor any available filesystem except NFS-mounted filesystems. Most platforms do not report available inodes for NFS-mounted filesystems.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Greater than 98%	Send TME 10 notice. Popup alarm. Change icon.
severe	Greater than 95%	Send TME 10 notice. Change icon.
warning	Greater than 90%	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

When an NFS server unlinks a file that is opened by a client, the client creates a file with a name of the form **.nfs***xxx* (where *xxx* is a number). When the open file is closed, this file is removed. If the client crashes before the file can be closed, the file remains. These files can accumulate and take up disk space and inode resources. You could therefore create a monitor that deletes all **.nfs***xxx* files whenever the percentage of inodes used reaches 50 percent.

NAME

Percent space used

Purpose

Checks the percentage of disk space used in a filesystem.

Partial CLI Synopsis

diskusedpct -a *filename*

Description

The **Space free** monitoring source monitors the percentage of space used on a specified filesystem. You can monitor any filesystem that is available to the subscriber, including NFS-mounted filesystems.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Greater than 95%	Send TME 10 notice. Popup alarm. Change icon.
severe	Greater than 90%	Send TME 10 notice. Change icon.
warning	Greater than 85%	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your **/home** filesystem can hold 100 MB of data. You can set up a monitor to run a user-created script that archives and compresses your **mail** subdirectory whenever the filesystem is 90% full.

NAME

Process instances

Purpose

Counts copies of a daemon or process.

Partial CLI Synopsis

daemonct -a *daemon*

Description

The **Process instances** monitoring source counts the number of currently-running copies of the specified daemon or process. This is useful for daemons like **nfsd** and **biod**, which typically run several instances simultaneously. It could also be used to track the number of license-restricted applications currently running.

Default actions are not available for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

daemon Specifies the name of the daemon or process to monitor. You can chose the following daemons from a scrolling list, or enter a a different daemon or process.

- amd (automount)
- biod
- inetd
- lockd
- lpd
- mountd

- nfsd
- portmap
- snmpd
- statd

Note: Do not specify the absolute path for any of these daemons. Other daemons or processes may require the full path name.

Examples

Suppose you have an application for which you have 10 licenses. You could use the **Daemon instances** monitoring source to monitor the number of this application running at any time. When 10 copies of the application are running, you could set a critical action of sending e-mail to **all**, requesting that anyone not currently using the application exit the application.

NAME

Remote oserv status

Purpose

Monitors the status of the oserv daemon.

Partial CLI Synopsis

oserv -a *hostname*

Description

The **Remote oserv status** monitoring source checks the status of the **oserv** daemon on any remote TME host. However, you should not monitor the local **oserv** daemon, as the monitors will not function if the daemon is not available. You can monitor an **oserv** daemon in a connected TMR if the connection allows activity in the appropriate direction.

Note: This monitor should not be distributed to a large number of hosts. It would be wasteful to have many hosts checking the status of a a single **oserv** daemon. It is a good idea to group **oserv** monitors in a TME 10 Distributed Monitoring profile with other **oserv** and **host** monitors. Such a profile should be distributed to a small set of nodes.

Profiles that contain this monitor must be set up with a user ID that maps to a defined TME 10 administrator on all subscribing managed nodes. Additionally, the administrator must have authorization to access the target managed node.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

hostname Specifies the name of the TME host to monitor.

Examples

If the **oserv** daemon associated with a given TME host goes down, administrators on that host will not be able to use TME. Therefore, you may want to be very certain that the proper administrator is notified immediately by sending mail, displaying a pop-up window, and running a script that pages the administrator. You may also want to attempt to restart all client instances of the **oserv** daemon by running the **odadmin start clients** command.

NAME

RPC bad calls

Purpose

Monitors the number of rejected RPC requests.

Partial CLI Synopsis

badrpc

Description

The **RPC bad calls** monitoring source monitors the number of RPC bad calls received since the last time the statistics were reset. The number of RPC bad calls is the sum of the number of calls that were smaller than the minimum size established for a RPC request and the number of calls that could not be decoded by XDR.

Numerous NFS bad calls is an indication that UDP packets are getting corrupted in the path between the server and client machines. This can be caused by hardware problems with the network, such as bad repeaters or cabling, an ailing bridge or router corrupting packets, high error rates on leased circuits (such as X.25 or T1), or network software problems on the server or client machines.

Note: You must use the **nfsstat** command to reset statistics.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	50% increase	Send TME 10 notice. Popup alarm. Change icon.
severe	25% increase	Send TME 10 notice. Change icon.

Severity	Trigger When	Default Actions
warning	10% increase	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

To help get a snapshot of your network when a RPC bad call monitor triggers, you can create a script that uses the **nfsstat** command to return information about the server and clients on your network. You can use this information to help debug your system.

NAME

Space free

Purpose

Checks the amount of free disk space in a filesystem.

Partial CLI Synopsis

diskavail -a *filename*

Description

The **Space free** monitoring source monitors the amount of free space on a specified filesystem. You can monitor any filesystem that is available to the subscriber, including NFS-mounted filesystems. This value is reported in megabytes.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Less than 5.0 MB	Send TME 10 notice. Popup alarm. Change icon.
severe	Less than 10.0 MB	Send TME 10 notice. Change icon.
warning	Less than 15.0 MB	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your **/usr** filesystem can hold 200 MB of data. You can set up a monitor to run a user-created script that deletes **.backup** files created by your word processing application that are more than one week old whenever the filesystem has 10 MB of free space.

NAME

Space used

Purpose

Checks the amount of occupied disk space in a filesystem.

Partial CLI Synopsis

diskused -a *filename*

Description

The **Space used** monitoring source monitors the amount of space used on a specified filesystem. You can monitor any filesystem that is available to the subscriber, including NFS-mounted filesystems. The value is reported in megabytes. Because a filesystem can have any maximum size, you must explicitly state the sizes at which a monitor will trigger.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	Send TME 10 notice. Popup alarm. Change icon.
severe	never	Send TME 10 notice. Change icon.
warning	never	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename Specifies the name of a file or directory that resides on the filesystem to be monitored. This value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your **/usr/local** filesystem can hold 150 MB of data. You can set up a monitor to run a user-created script that deletes **.backup** files created by your word processing application that are more than one day old whenever the filesystem contains 135 MB of data.

NAME

Status of print queue

Purpose

Monitors the availability of a printer queue.

Partial CLI Synopsis

printstat -a *print_queue*

Description

The **Status of print queue** monitoring source monitors availability of a printer queue.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Becomes unavailable	Send TME 10 notice. Popup alarm. Change icon.
severe	never	none
warning	never	none
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

print_queue Specifies the name of the printer queue to monitor.

Examples

You can add the command name that restarts the print queue (such as **lpc restart** in SunOS) in the **Run program** text box to help avoid printer downtime.

NAME

String script

Purpose

Runs a user-defined script that returns a string value and is compatible with TME 10 Distributed Monitoring.

Partial CLI Synopsis

scustom -a *program*

Description

The **String script** monitoring source executes a user-supplied script or program and interprets its output as a string. The program should print its result data on standard output.

All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use string operators to evaluate data. You can use the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

filename	Specifies the name of the filesystem to monitor.

Examples

You can use this monitoring source to create a monitor that detects **core** files. To do this, you could write a script that executes the **ls | grep core** command in each directory of a filesystem. If a **core** file is detected, you could trigger a **warning** alarm that indicates the location of the file.

NAME

Tivoli DB free space

Purpose

Monitors the amount of free disk space on the filesystem containing the TME 10 object database.

Partial CLI Synopsis

tivdbspace

Description

The **Tivoli DB free space** monitoring source monitors the amount of free disk space on the filesystem containing the TME 10 object database. This database stores all information about your TME 10 environment and TME 10 applications.

If the allotted space for your TME 10 database is nearly full, you may want to perform the following activities:

- Allocate more disk space on the filesystem.
- Delete any ***.restore** files located in your **$DBDIR**/*host***.db** directory
- Compress and archive the TME 10 notification database and logfile. These files are located in your **$DBDIR**/*host***.db** directory. Once you have archived the notification database, use the **wexpnotif** command to expire all notices in the notification database.
- Delete, archive, or compress the contents of your **$DBDIR**/*host***.db/file_versions** directory.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	Less than 1 MB	Send TME 10 notice. Popup alarm. Change icon.
severe	Less than 5 MB	Send TME 10 notice. Change icon.
warning	Less than 10 MB	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

Adding clients and populating profiles causes the TME 10 object database to expand. You can use the **Tivoli DB free space** monitoring source to create a monitor that copies the TME 10 notification database to an another filesystem and delete the original notices.

NAME

Total size queued

Purpose

Monitors the total size of all jobs queued on a printer queue.

Partial CLI Synopsis

printjobsize -a *print_queue*

Description

The **Total size queued** monitoring source monitors the size of the entries in a specified printer queue. This activity is useful for flagging large print jobs, so you can know when your printer queue is stacking up.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	Send TME 10 notice. Popup alarm. Change icon.
severe	never	Send TME 10 notice. Change icon.
warning	never	Change icon.
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

print_queue Specifies the name of the print queue to monitor.

Examples

To help prevent an overburdened printer from receiving even more print jobs, set up a monitor so that it broadcasts a message stating the printer is busy. You can create a shell script for each printer so that the message is customized.

NAME

User logins by user

Purpose

Monitors the number of a user’s concurrent login sessions.

Partial CLI Synopsis

ulogins -a *user_name*

Description

The **User logins by user** monitoring source monitors the number of concurrent login sessions owned by the specified user. Note that this may not be the same as the total number of processes owned by a particular user.

The following table lists the default actions for this monitoring source:

Severity	Trigger When	Default Actions
critical	never	none
severe	never	none
warning	never	none
normal	N/A	none
always	N/A	none

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Argument

user_name Specifies the login name to monitor.

Examples

If you do not want user **jsmith** to login to host **falstaff**, enter **jsmith** as the value for the *user_name* argument. You can then set the critical response level to trigger when the number of logins is equal to **1**.

NAME

Users logged in

Purpose

Monitors the number of users currently logged in.

Partial CLI Synopsis

ulogintot

Description

The **Users logged in** monitoring source monitors the total number of users currently logged in.

There are no default actions for this monitoring source.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

Your main server has a large load average. In order to minimize its load, you do not want too many users to be logged in at any given time. To help remind users of this, you could send e-mail to those logged in when the number of logins is greater than **3**.

15

Universal Monitoring Collection

The Universal monitoring collection provides monitoring sources that enable you to use TME 10 Distributed Monitoring to manage distributed computing resources effectively. Each monitoring source enables you to manage a different aspect of the operating system, such as daemons or other processes, disk space, and network collisions. You can also use the asynchronous and user-defined monitoring sources to create your own powerful monitoring sources.

To reduce system overhead, the daemon status monitor maintains a snapshot of all currently running processes, rather than continually checking for running processes.

The following tables list the monitoring sources provided with this monitoring collection.

Disk Resource Monitoring Sources

GUI Name	CLI Name	Described on Page
Disk Space Free	diskavail	15-6
Disk Space Used	diskused	15-8
Disk Space Percentage Used	diskusedpct	15-10

GUI Name	CLI Name	Described on Page
Inodes in Use	inodesused	15-21
Inodes Available	inodesfree	15-22
Megabytes in Use	filesystemused	15-8
Megabytes Available	filesystemfree	15-6
Percent Space in Use	filesystempctf	15-10

Security Monitoring Sources

GUI Name	CLI Name	Described on Page
Application Instances	appInstances	15-4
Application Status	appStatus	15-5
File Checksum	filechk	15-12
File Compare	filediff	15-13
File Pattern Matches	countstr	15-15
File Permissions	fileperm	15-17
File Size	filesize	15-19

System Resources Monitoring Sources

GUI Name	CLI Name	Described on Page
Logins Enabled	LoginsEnabled	15-23
Load Average (Five Minute)	loadavg	15-25
Page-outs	pageouts	15-26
Ping All TMR Hosts	PingAlHosts	15-27
Ping Some TMR Hosts	PingSomeHosts	15-27
Swap Space Available	swapavail	15-38
Remote Oserv Status	oserv	15-36

User-Defined Monitoring Sources

GUI Name	CLI Name	Described on Page
Asynchronous Numeric	nasync	15-39
Asynchronous String	sasync	15-40
Numeric Script	ncustom	15-41
String Script	scustom	15-42

NAME

Application instances

Purpose

Counts copies of a application or daemon.

Partial CLI Synopsis

appInstances -a *application*

Description

The **Application instances** monitoring source counts the number of currently-running copies of the specified application or daemon. This monitoring source is useful for tracking the number of license-restricted applications that are currently running.It can also monitor daemons such as **nfsd** and **biod**, which typically run several instances simultaneously.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

application Specifies the name of the application or daemon to monitor. Do not specify the absolute path for a daemon or process.

Examples

Suppose you have an application for which you have 10 licenses. You could use the **Application instances** monitoring source to monitor the number of instances of this application that are running at any given time. When 10 copies of the application are running, you could set a critical action of sending e-mail to the **writers** mail alias, requesting that anyone who is not currently using the application exit the application.

NAME

Application status

Purpose

Monitors the availability of an application or daemon.

Partial CLI Synopsis

appStatus -a *application*

Description

The **Application status** monitoring source monitors the availability of a specified application or daemon. For instance, you could monitor for the absence of the **inetd** daemon, and use the **Run program** option to call a script that restarts the daemon.

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

application	Specifies the name of the daemon or process to monitor. Do not specify the absolute path of the daemon or process.

Examples

The TME 10 Enterprise Console uses a a relational database management system (RDBMS) to keep track of system events. If the daemon that controls the RDBMS becomes unavailable, incoming events cannot be written to the database. To help minimize the effects of this problem, you can create a monitor that restarts the RDBMS daemon whenever it goes down.

NAME

Disk space free

Purpose

Checks the amount of space available on a drive or directory.

Partial CLI Synopsis

diskavail -a *resource*

Description

The **Disk space free** monitoring source monitors the amount of unused space on a specified logical drive or directory. You can monitor any logical drive or directory that is available to the subscriber, including NFS-mounted filesystems. The value is reported in megabytes.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

For NFS-mounted filesystems, the specified directory should be a hard mount point. If the specified directory is an automounted point, the mount point may not exist and invalid results may be returned.

Arguments

resource Specifies the directory to be monitored. On NT machines, you must include the logical drive letter to be monitored. On UNIX machines, this value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your **K:** drive can hold 100 MB of data. You can set up a monitor to run a user-created script that deletes **.backup** files created by your word processing application that are more than one week old whenever the drive has 10 MB or less free space.

NAME

Disk space used

Purpose

Checks the amount of space used on a drive or directory.

Partial CLI Synopsis

diskused -a *resource*

Description

The **Disk space used** monitoring source monitors the amount of space used on a specified logical drive or directory. You can monitor any logical drive or directory that is available to the subscriber, including NFS-mounted filesystems. The value is reported in megabytes. Because a logical drive or filesystem can have any maximum size, you must explicitly state the sizes at which a monitor will trigger.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

For NFS-mounted filesystems, the specified directory should be a hard mount point. If the specified directory is an automounted point, the mount point may not exist and invalid results may be returned.

Arguments

resource Specifies the directory to be monitored. On NT machines, you must include the logical drive letter to be monitored. On UNIX machines, this value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your **/usr/local** filesystem can hold 150 MB of data. You can set up a monitor to run a user-created script that deletes **.backup** files created by your word processing application that are more than one day old whenever the filesystem contains 135 MB of data.

NAME

Disk space used (%)

Purpose

Checks the percentage of disk space used on a drive or directory.

Partial CLI Synopsis

diskusedpct -a *resource*

Description

The **Disk space used (%)** monitoring source monitors the percentage of space used on a specified logical drive or directory. You can monitor any logical drive or directory that is available to the subscriber, including NFS-mounted filesystems. The value is reported in megabytes. Because a logical drive or directory can have any maximum size, you must explicitly state the sizes at which a monitor will trigger.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

For NFS-mounted filesystems, the specified directory should be a hard mount point. If the specified directory is an automounted point, the mount point may not exist and invalid results may be returned.

Arguments

resource Specifies the directory to be monitored. On NT machines, you must include the logical drive letter to be monitored. On UNIX machines, this value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Your **K:** drive can hold 100 MB of data. You can set up a monitor to run a user-created script that deletes **.backup** files created by your word processing application that are more than one week old whenever the drive has 10 MB or less free space.

NAME

File checksum

Purpose

Computes the checksum of a specified file.

Partial CLI Synopsis

filechk -a *filename*

Description

The **File checksum** monitoring source computes the 16-bit checksum of a specified file. A checksum is typically used to determine whether a file is an exact duplicate of another file or to watch for a change in a file.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Note: There is no **sum** command on NT systems, so a perl script to calculate a checksum is installed on i386 systems during installation of the UNIX-NT monitoring collection. Some NT editors also may insert control characters (such as Ctrl-m) in a file. Because of these factors, the same file may return different checksum values on UNIX systems and NT systems.

Arguments

filename	Specifies the name of the file to monitor.

Examples

Suppose the **/etc/passwd** file on your system has a checksum of **26289**. With that knowledge, you could set a critical response to trigger when the checksum of the **/etc/passwd** file changes. If an authorized change has been changed, use the **sum** command to obtain a new checksum.

NAME

File compare

Purpose

Compares two files.

Partial CLI Synopsis

filediff -a *filename* **-a** *filename* **-a** *options*

Description

The **File compare** monitoring source compares two text or binary files. An action can be triggered when files are the same, are different, become the same, or become different.

Monitors that are defined with this monitoring source use file operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

filename Specifies the name of a file to monitor.

options Specifies one of the following options:

Plain difference

Indicates the comparison will include checks for differences in whitespace and capitalization. This is the default value. Counts the number of bytes in a file. Use **“”** to specify this option from the command line.

Ignore whitespace

Indicates the comparison will ignore differences in spacing. Use “**-bw**” to specify this option from the command line.

Ignore alphabetic case
Indicates the search will not distinguish between uppercase and lowercase characters. Use “**-i**” to specify this option from the command line.

Ignore case & whitespace
Indicates the search will be case-insensitive and ignore spacing differences. Use “**-ibw**” to specify this option from the command line.

Binary Indicates the monitor will compare two binary files. Use “**--bin--**” to specify this option from the command line.

Examples

On an NT machine, the **CONFIG.SYS** file contains settings that configure hardware components such as memory, keyboard and mouse. Because some applications edit this file upon installation, you can monitor this file to check for unauthorized programs. To set up this monitor, keep a copy of the **CONFIG.SYS** file in another directory and use it as a basis for comparison. If changes were made to this file, you can check to see whether the changes are acceptable.

NAME

File pattern matches

Purpose

Counts occurrences of a pattern in a file.

Partial CLI Synopsis

countstr **-a** *pattern* **-a** *filename* [**-a** *search_option*]

Description

The **File pattern matches** monitoring source searches for a pattern in a file and reports the number of times the pattern was found. The arguments to this monitoring source consists of the pattern, the name of the file to be searched, and search options.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Arguments

pattern Specifies a string or regular expression to search for.

filename Specifies the name of the file to monitor.

search_option Specifies one of the following options:

Normal search Indicates that no special search conditions are required. This is the default value. Use **""** to specify this option from the command line.

Ignore alphabetic case
Indicates the search will not distinguish between uppercase and lowercase characters. Use "**-i**" to specify this option from the command line.

Inverse search Indicates the search results will contain only those lines that do not contain the target pattern. Use “**-v**” to specify this option from the command line.

Match whole words Indicates the pattern matching will be done against whole words from the lines of the file Use “**-w**” to specify this option from the command line. This option has no effect on some platforms due to lack of direct support from the **grep** utility.

Examples

User **jsmith** has been denied access privileges to the **DIAMOND** NT server because he has misused these privileges in the past. To help minimize the damages if he somehow regains access, you could search for the pattern **jsmith** in the security log that resides on **DIAMOND**.

NAME

File permissions

Purpose

Checks the permissions of a file.

Partial CLI Synopsis

fileperm -a *filename*

Description

The **File permissions** monitoring source returns the permissions string from a file and allows string comparisons to be made on the value. You must specify a search pattern that is capable of matching, either exactly or by using regular expressions, the permissions displayed when you execute the **ls -l** command. For example, the permissions for the **/home/report.doc** file might be the following:

```
-rw-r--r--
```

Therefore, to determine whether the someone has changed the permissions so that other members of user-group have write access, you could enter one of the following strings (among others):

```
-rw-rw-r--
```

—OR—

```
-rw-rw*
```

Monitors that are defined with this monitoring source use string operators to evaluate data. You can use the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Note: If you want to use wildcard characters in your file permission string, use the **Matches** operator.

Arguments

filename Specifies the name of the file to monitor.

Examples

To determine when the write permissions of the **/etc/passwd** file have changed from **-rw-r--r--**, set up a monitor so that the search string is **-rw-*w***.

NAME

File size

Purpose

Computes the size of a specified file.

Partial CLI Synopsis

filesize -a *filename* **-a** *size_units*

Description

The **File size** monitoring source computes the size of a specified file. The size can be determined in bytes, words, or lines. This source is especially helpful for tracking the size of log files or other files that are appended frequently.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Arguments

filename Specifies the name of the file to monitor.

size_units Specifies one of the following options:

Bytes Counts the number of bytes in a file. Use "**-c**" to specify this option from the command line.

Words Counts the number of words in a file. Use "**-w**" to specify this option from the command line.

Lines Counts the number of lines in a file. Use "**-l**" to specify this option from the command line.

Examples

On an NT machine, the **CONFIG.SYS** file contains settings that configure hardware components such as memory, keyboard and mouse. Because some applications edit this file upon installation, you can monitor this file to check for unauthorized programs. To set up this monitor, determine the size of the **CONFIG.SYS** file and use this value as a basis for comparison. If changes were made to this file, you can check to see whether the changes are acceptable.

NAME

Inodes in Use

Purpose

Determines the number of inodes currently in use.

Partial CLI Synopsis

inodesused -a *resource*

Description

The **Inodes in Use** monitoring source determines the number of indoes that are currently in use on a specified filesystem. You can monitor any logical drive or directory that is available to the subscriber, including NFS-mounted filesystems.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

For NFS-mounted filesystems, the specified directory should be a hard mount point. If the specified directory is an automounted point, the mount point may not exist and invalid results may be returned.

Arguments

resource Specifies the directory to be monitored. On NT machines, you must include the logical drive letter to be monitored. On UNIX machines, this value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your K:/ drive has 16384 inodes. You can set up a monitor to warn you when 14000 or more inodes are currently in use.

NAME

Inodes Available

Purpose

Determines the number of inodes currently available.

Partial CLI Synopsis

inodesfree -a *resource*

Description

The **Inodes Available** monitoring source determines the number of indoes that are currently available on a specified filesystem. You can monitor any logical drive or directory that is available to the subscriber, including NFS-mounted filesystems.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

or NFS-mounted filesystems, the specified directory should be a hard mount point. If the specified directory is an automounted point, the mount point may not exist and invalid results may be returned.

Arguments

resource Specifies the directory to be monitored. On NT machines, you must include the logical drive letter to be monitored. On UNIX machines, this value should indicate the mount point or a file or directory located beneath the mount point. Do not specify a device name (for example, **/dev/hd0a**) for this argument.

Examples

Your K:/ drive has 16384 inodes. You can set up a monitor to warn you when fewer than 2000 inodes are available.

NAME

Logins Enabled

Purpose

Monitors the status of a host.

Partial CLI Synopsis

LoginsEnabled -a *hostname*

Description

The **Logins Enabled** monitoring source checks the status of any host or other entity on the network that can respond to a **ping** request.

Note: Depending on network topology, monitoring hosts may be inherently problematic. For example, if a host subscribed to a host status monitor is separated by a router from the monitored host, the failure of the router is interpreted as the failure of the host.

Use this monitor with restraint. It is generally a good idea to create profiles with several host status entries and then subscribe only a few hosts to these profiles. Also, you should consider whether a host really needs to be monitored. For example, you may want to monitor only those hosts that act as servers.

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

hostname	Specifies the name of the host to monitor.

Examples

If a key file server becomes unavailable, many employees may not be able to perform their duties. Therefore, you may want to be very certain that the proper administrator is notified immediately by sending mail, displaying a pop-up window, and running a script that pages the administrator.

NAME

Load average

Purpose

Monitors the average number of active processes.

Partial CLI Synopsis

loadavg

Description

The **Load average** monitoring source monitors the average number of processes that were active for the last five minutes.

The definition of a high average load varies depending upon the system. A single-user workstation should have a low average load (1 or 2), while an average load of 10 is not uncommon for multi-user hosts. Results of this monitor are platform-dependent, particularly when used on multi-user hosts.

NFS input/output activities waiting for completion are included in the load average. If your TME installation includes an NFS file server, the average load may be abnormally high, and could cause TME 10 Distributed Monitoring to generate an alarm.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Examples

If the load average of a server reaches the critical state, you could send a message to the **all** mail alias requesting that users use other system resources or cancel non-essential processes.

NAME

Page-outs

Purpose

Monitors pages written out by the virtual memory manager.

Partial CLI Synopsis

pageouts

Description

The **Page-outs** monitors the average number of pages that the virtual memory manager pages out per second on each subscriber. Paging is the method of writing portions of a process's memory to disk to free up memory.

Note: You must install the **bos.acct** package as part of the AIX operating system in order to use the **Page Outs** monitor in the UNIX-NT monitoring collection. The **vmstat** command, which is used by the **Page Outs** monitor, is part of the **bos.acct** package.

If the number of page-outs is significantly greater than zero for several monitoring intervals, the host could have a memory problem. You will need to experiment with different thresholds to determine the right values for your system.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Examples

You can run this monitor in conjunction with the **Load Average** monitoring source to determine whether an application server is overloaded. If the server is overloaded, you may want to reallocate system resources.

NAME

PingAllHosts, **PingSomeHosts**

Purpose

The PingTMR monitors (**PingAllHosts**, **PingSomeHosts**) have been added to the universal monitoring collection to provide enhanced monitoring capabilities for collecting and reporting machine uptime and downtime across the nodes within a TMR. These monitors use a customized monitoring approach where data is collected across the TMR in parallel. This data is then evaluated in separate passes. This allows for optimized collection using TME 10 internal methods and significantly lowers the overhead that is associated with reporting status on large numbers of endpoints.

Description

To achieve this level of optimization, these monitors employ a different usage model than other monitors employ. Most monitors run within a reporting cycle and then report their results at the scheduled response time, providing a value so that selected responses can be dispatched. These monitors run for an unpredictable amount of time; this time increases when more nodes in a TMR are down. This is based on the amount of time required for network **ping** commands to time out. Also, as the results of each host's **ping** is received, the monitor can issue a TME 10 Enterprise Console event directly to the server that is configured in the monitor arguments. This behavior is independent of the normal value/response action sequence of most monitors.

Because of these execution characteristics, there are several high-level usage issues that the user must understand in order to correctly configure the **PingAllHosts** and **PingSomeHosts** monitors:

- The only information that is reported in the value field of the monitor (and therefore the only information on which responses can be based) is the total number of nodes that are down in a TMR
- Most users will want to use these monitors for TME 10 Enterprise Console reporting of the specific host status. This capability is not available from the normal response cycle.

- The individual host status is not available through the usual execution/response sequence of monitor execution
- Due to the time that is required to collect the host status across a large number of TMRs, you should carefully consider the appropriate frequency of execution for this monitor. Guidelines are provided in this section.
- Events are generated to the TME 10 Enterprise Console server whenever the status of an endpoint changes (a host goes down, or a host that was down goes up)
- Due to the need to relate the host status between two cycles and the fact that the evaluation process runs on the collection results from the previous execution cycle, TME 10 Enterprise Console events are generated two cycles after a host has changed status.

The collection of TMR data is started in the background, since data collection can take longer than 60 seconds. After the TMR data is collected, the data is immediately analyzed. The PingTMR monitor analysis process takes into account the bootstrapping that is inherent in the TMR data collection process and issues a **Sentry-Status** notice when the monitor is first run. This notice indicates the initial lack of TMR data for analysis.

The PingTMR monitors must be run as root on a node where root is a valid TME 10 administrator. If ICMP is not needed, the monitor does not have to be run as root, but there must be a TME 10 administrator login on the node.

Arguments

The arguments for the **PingAllHosts** monitor are:

SendTECEvent Specifies how TEC events are to be sent. Possible values are **TRUE**, **FALSE**, **SECURE**, and **TEST**:

TRUE	Send events with **postemg**.
FALSE	Do not send events.
SECURE	Send events with **wpostemsg**.
TEST	Log state changes to the **/tmp/Event.log** file

EventServerName

Specifies the TEC server. If this value is specified, it is used as the **-S** argument to the **postemsg** or **wpostemsg** command. The value of this argument depends on the type of event (**postemsg** or **wpostemsg**) that is sent.

The arguments for the **PingSomeHosts** monitor are:

PortNumber Specifies the TCP port number to test. To test the **oserv** port on each node, specify **-1**. You can specify other ports if desired. For example, port 80 is the well-known port for an HTTP server, port 25 is the **sendmail** port, and port 515 is the remote printer port.

StartingIndex Specifies the **odlist** index at which to start testing. The monitor can check a subset of all nodes in the TMR. The default value is **1**, which will check all nodes in the TMR. If you specify a value greater than **1**, the monitor will start checking at the specified index in the **odlist**.

BlockSize Specifies the number of sequential nodes in **odlist** to be tested. For example, a `BlockSize` of 25 will ping the next 25 entries in the **odlist**, starting at the node specified in `StartingIndex`. To test all nodes, specify **-1**.

SendTECEvent Specifies how TEC events are to be sent. Possible values are **TRUE**, **FALSE**, **SECURE**, and **TEST**:

TRUE	Send events with **postemsg**.
FALSE	Do not send events.
SECURE	Send events with **wpostemsg**.

TEST	Log state changes to the **/tmp/Event.log** file.

EventServerName

Specifies the TEC server. If this value is specified, it is used as the **-S** argument to the **postemsg** or **wpostemsg** command. The value of this argument depends on the type of event (**postemsg** or **wpostemsg**) that is sent.

Parameters

If the ICMP timeout indicates that some nodes are down, you can change the ICMP timeout value. The ICMP delay time is specified in the **wfping.sh** script. If four seconds is not adequate for your network topology (for example, in a WAN), you can edit this script.

Note: If you install any future product patches or upgrades for the universal monitoring collection, your edited ICMP timeout value will be overwritten and you will need to edit the **wfping.sh** script again.

Version

There is no single recommendation for how often the **PingAllHosts** and **PingSomeHosts** monitors should be run. If you are trying to uphold a Service Level Agreement (SLA) with a fixed response time, you should run the monitor at least as often as specified by the SLA.

You can also use a formula to determine your best option for monitor frequency for the **PingAllHosts** or **PingSomeHosts** monitors. The following formula provides an estimate of the time (in seconds) that is required to collect all of the data:

CollectionTime = **(***DOWN_CNT* *** 4) + ((***O_DOWN_CNT* *** 60)/4) + ((***NODE_CNT* **-** *DOWN_CNT* **-** *O_DOWN_CNT***) * 0.1)**

DOWN_CNT	The average number of nodes in the TMR that will fail the ICMP echo test
O_DOWN_CNT	The average number of nodes in the TMR that will pass the ICMP echo test while the oserv is down
NODE_CNT	The total number of nodes in the TMR

For example, if there are 150 nodes in a TMR, of which on average 12 are down and 8 have an **oserv** that is down:

(12 * 4) + ((8 * 60)/4) + (130 * 0.1) = 181 seconds

Note: This is a pessimistic estimate. The actual time required to gather all of the data should never be this long.

If your collection time estimate is greater than half of your SLA time, you should break your collection into smaller chunks and then run these on different nodes.

System Environment

Under some conditions, it is possible that this monitor will indicate that a node or oserv is down when the node or oserv is actually up. These conditions are:

- Heavy network load The ICMP ping timeout is set to 4 seconds. If a node does not respond to a ICMP ECHO request within 4 seconds, the node is marked DOWN.
- It is possible that some nodes will pass the ICMP echo test, but will still take the long timeout path.

The technique used to detect an up/down condition will not cause **inetd** to start a process.

Since the monitor does NOT attempt to start the oserv if it is down, the monitor may report that the oserv is down, but the oserv may have been started by the time the data analyzer runs (one cycle after the collector). This is the intended design.

Restrictions

This section discusses the architecture of the PingTMR monitoring collection. This information is helpful in understanding how the PingTMR monitoring collection works, but it is not necessary in order to use the monitoring collection.

Data Collection

A machine or process can have three states:

- **Host Down -** The host is not running a TCP/IP stack.
- **Host UP, Process Down -** The host (and TCP/IP stack) is running, but the specified process (usually the **oserv** daemon) is not running.
- **Host Up, Process Up -** The host and the process are both running.

The data collection program efficiently detects these three states while generating a minimum amount of ORB traffic. The following steps are used to collect data:

1. Fetch the dispatcher list from the TMR server.
2. From the dispatcher list, fetch the IP address and port number of each oserv.
3. Traverse the list, attempting to connect to each IP address/port pair.
4. For each failed connection, mark the host as **DOWN**.

Some overrides are possible. You can choose an alternate port to connect to for all hosts. You can also specify a subset of the entire host list to test, by specifying an offset/block size or by specifying a list of host labels.

Mechanically, the data collection program attempts to perform the following actions:

1. Ping the remote system with two ICMP echo packets.
2. Create a local socket and connect to the remote hosts’ IP address/port number (by default the oserv port).

If step one fails, the host is marked down. If step one is successful, then step two is attempted. Step two can result in a fast timeout (if there is no process listening on the remote host) or a slow timeout (in some cases, or if the ICMP echo test is skipped (for NT) and the host is down). For cases where all hosts are up, the **UP**/**DOWN** state of the process can be determined at a rate exceeding 25 hosts per second. For cases where the remote host is down, a slow timeout occurs between 30 seconds and 2 minutes, depending on the TCP/IP implementation of the host running the data collection program.

If the ICMP echo fails after four seconds, the node is marked down. NT does not have the ICMP text, so this monitor is more effective when run on a non-NT node.

By skipping the ICMP check and allowing TCP to timeout on the slowest implementations the data collection program will take *N* x 120 + 5 seconds to collect state information for all nodes in a TMR (where *N* is the number of hosts which are completely down). So in a 250-node installation with 50 hosts down, it will take approximately 100 minutes to complete the state test if the test is run on one host. It is possible to split the test so that 5 hosts test 50 nodes each, reducing the total latency to 20 minutes. Each run of the data collection program causes 3 objcalls, so further reduction into smaller test groups provides a diminishing improvement. Using the ICMP check reduces the same test to less than four minutes on one host.

Clearly, this collection period exceeds TME 10 Distributed Monitoring’s 60-second time window. To compensate for this, the data collector is run in the background (**fping**), logging its data to a file in *$DBDIR*/**pingTMR**. The background subshell is a compound statement that logs to a temp file, then shuffles the logs at the conclusion of the run. The collection monitor should be scheduled to run at an interval which matches the desired latency for down-host detection and the expected number of down hosts. If the interval is too short, the data collection code will detect that another collection is running and will send a notice to the **Sentry-Status** notice group.

The collection monitor can be run on any node that has a root Administrator login, so choosing an INTERP other than NT has obvious advantages.

The total network cost of a TMR ping is:

- 3 x TMR objcalls [per TMR]
- 2 x UDP ICMP echo send/receive [per node]
- 1 x TCP connect to address/port (with immediate disconnect) [per node]

INTERP	TCP/IP connect timeout period (seconds)
HP-UX/9	60
HP-UX/10	30
AIX3R2	60
AIX4R1	60
SunOS4	60
Solaris2	120
W32-IX86 (NT 4.0)	30

Data Analysis Details

There are two interesting state changes in the analysis, a **DOWN** to **UP** transition and an **UP** to **DOWN** transition. When a measurement reveals that a host has not changed state, there is no event of interest. This reduces the total number of events to just those that indicate the first detection of a state change. By providing events for both transitions, TEC rules can be used to filter out the ‘short-down-time’ events when an host/process was down for a short period of time.

The data analysis monitor uses one or two log results from the data collection monitor. If no logs are present, a **Sentry-Status** notice is generated which notes the lack of logfiles and asks if the collection monitors are running.

The analysis monitor first checks the newer lo to determine which nodes are in the newer log but not in the older log. If no older log exists then all nodes in the new log are reported as errors (the monitor returns a numeric value indicating the number of nodes that are down, with the node names in the error text). If both logs exist, the older log is then checked to see which nodes were down but now are up. These nodes generate an **UP** event (to the TEC) if the analysis monitor is configured to do so. Each down node also generates a **DOWN** event when configured. Therefore, a single run of the analysis monitor will result in a numeric return value that is equal to the number of nodes that are currently down, plus a TEC event for each down node and a TEC event for each node that was down but is now up.

There is a locking protocol between the data collection monitor and the analysis monitor such that the analysis will hold off (up to 15 seconds) if the collector is updating the files. The collector logs to a temp file and then renames the files at the end of its run, so the update period is small (less than 1 second).

The analysis monitor copies the logfiles (if they are not locked) and works on the copies so it will not be affected by a collection monitor that runs after the analysis monitor starts.

Logging Details

The data collection and analysis code uses a set of logging files which will reside in *$DBDIR*/**pingTMR**.

- **PingLog.old -** The previous (complete) log from data collection
- **PingLog.now -** The most current (complete) log from data collection
- **PingLog.tmp -** The logfile used by the data collector
- **PingLog.lck -** Indicates a collection is swapping new and old logfiles
- **PingLog.prb -** Indicates a data collection run is occurring.
- **xPingLog.old -** The previous (complete) log from data analyzer
- **xPingLog.now -** The most current (complete) log from data analyzer
- **/tmp/Event**.log **-** If the monitor is run in test mode, this shows what TEC events will be sent.

NAME

Remote oserv status

Purpose

Monitors the status of the oserv daemon.

Partial CLI Synopsis

oserv -a *hostname*

Description

The **Remote oserv status** monitoring source checks the status of the **oserv** daemon on any remote TME host. However, you should not monitor the local **oserv** daemon, as the monitors will not function if the daemon is not available. You can monitor an **oserv** daemon in a connected TMR if the connection allows activity in the appropriate direction.

Note: This monitor should not be distributed to a large number of hosts. It would be wasteful to have many hosts checking the status of a a single **oserv** daemon. It is a good idea to group **oserv** monitors in a TME 10 Distributed Monitoring profile with other **oserv** and **host** monitors. Such a profile should be distributed to a small set of nodes.

Profiles that contain this monitor must be set up with a user ID that maps to a defined TME 10 administrator on all subscribing managed nodes. Additionally, the administrator must have authorization to access the target managed node.

Monitors that are defined with this monitoring source use status operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

hostname Specifies the name of the TME host to monitor.

Examples

If the **oserv** daemon associated with a given TME host goes down, administrators on that host will not be able to use the TME. Therefore, you may want to be very certain that the proper administrator is notified immediately by sending mail, displaying a pop-up window, and running a script that pages the administrator. You may also want to attempt to restart all client instances of the **oserv** daemon by running the **odadmin start clients** command.

NAME

Swap space available

Purpose

Monitors the amount of available swap space.

Partial CLI Synopsis

swapavail

Description

The **Swap space available** monitoring source monitors the amount of swap space (in MB) that each subscriber is using. Swap space is usually a disk partition on which page-outs are written.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Examples

It is generally recommended that you should have at least twice as much swap space as memory. If an available swap space monitor frequently triggers, you may want adjust your swap space accordingly.

NAME

User Monitor, Asynchronous numeric

Purpose

Responds to events sent to the TME 10 Distributed Monitoring engine tagged with the given channel name.

Partial CLI Synopsis

nasync -a *channel_name*

Description

The **User Monitor, Asynchronous numeric** monitoring source indicates a channel to monitor for messages.

Channel names are arbitrary and are completely under user control. The relationship between a criterion monitoring a channel of a given name and specific events generated for that channel is only the name. Names only need to be registered with the **waddchan** command if it is desired that they appear in the **Choices** pop-up dialog.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

channel_name Specifies the name of the channel to monitor.

Examples

A job has been created with the Task Library to search certain software build directories for object files. The task can be enhanced so that once the object file count (or a total of the file sizes) is computed, the numeric result is forwarded via a **wasync** command to the local TME 10 Distributed Monitoring engine on each managed node subscribed to the job.

NAME

User Monitor, Asynchronous string

Purpose

Responds to events sent to the TME 10 Distributed Monitoring engine tagged with the given channel name.

Partial CLI Synopsis

sasync -a *channel_name*

Description

The **User Monitor, Asynchronous string** monitoring source indicates a channel to monitor for events.

Channel names are arbitrary and are completely under user control. The relationship between a criterion monitoring a channel of a given name and specific events generated for that channel is only the name. Names only need to be "registered" with the **waddchan** command if you want them to appear in the **Choices** pop-up dialog.

Monitors that are defined with this monitoring source use string operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, "TME 10 Distributed Monitoring Operator Groups."

Arguments

channel_name Specifies the name of the channel to monitor.

NAME

User Monitor, Numeric script

Purpose

Runs a user-defined script that returns a numeric value.

Partial CLI Synopsis

ncustom -a *program*

Description

The **User Monitor, Numeric script** monitoring source executes a user-supplied script or program and interprets its output numerically. The program should print its result as a numeric value to standard output. Numeric means that the output of the script is interpreted as a number (possibly a real number, such as **12.5** or **0.731**).

All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

Monitors that are defined with this monitoring source use numeric operators to evaluate data. You can select the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

filename	Specifies the name of the script to run.

Examples

To help prevent files from accumulating in the **/tmp** directory, you could write a script that executes the **ls** command and counts the entries. If there are 50 or more files in the directory, the script could delete these files.

NAME

User Monitor, String script

Purpose

Runs a user-defined script that returns a string value.

Partial CLI Synopsis

scustom -a *program*

Description

The **User Monitor, String script** monitoring source executes a user-supplied script or program and interprets its output as a string. The program should print its result data on standard output.

All lines of output that begin with a pound sign (#) will be passed along to all defined response mechanisms. The maximum size of an output line that TME 10 Distributed Monitoring can handle is 4096 bytes.

Monitors that are defined with this monitoring source use string operators to evaluate data. You can use the **trigger when** pop-up menu of the **Edit Monitor** dialog to display a list of the available operators. For information about these operators, see Appendix A, “TME 10 Distributed Monitoring Operator Groups.”

Arguments

filename	Specifies the name of the filesystem to monitor.

Examples

You can use this monitoring source to create a monitor that detects **core** files. To do this, you could write a script that executes the **ls | grep core** command in each directory of a filesystem. If a **core** file is detected, you could trigger a **warning** alarm that indicates the location of the file.

A

TME 10 Distributed Monitoring Operator Groups

The **Edit Monitor** dialog contains the **trigger when** pop-up menu, which allows you to define the conditions that must occur for a monitor to send an alert. The values displayed on this menu vary, depending on whether the monitor uses a numeric value, a string value, or a status message. These options are defined in ***operator groups***. Each option of an operator group serves as a mathematical or logical evaluator.

Each operator description includes the value for the **waddmon -R** argument. The quotation marks for each operator are required. If you want to specify the **(never)** operator from the command line, omit the **-R** argument.

Numeric Operator Group

To define the threshold value, select one of the following options and enter a number in the argument field.

(never)
Indicates the response level does not trigger a response. This value only affects the specific response level, not the entire monitor.

Greater than
Records the value of each sample, compares it to the specified value, and triggers if the current value is greater than the specified value. Because it

checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering “>”.

Less than

Records the value of each sample, compares it to the specified value, and triggers if the current value is less than the specified value. Because it checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering “<”.

Equal to

Records the value of each sample, compares it to the specified value, and triggers if the current value is equal to the specified value. Because it checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering “==”.

Not equal to

Records the value of each sample, compares it to the specified value, and triggers if the current value is not equal to the specified value. Because it checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering “!=”.

Increases beyond

Compares the current value to the trigger value and the previous value and triggers the first time that the current value is greater than the threshold that is specified in the argument field and is also greater than the previous value. This operator can be specified at the command line by entering “->>”.

Decreases below
Compares the current value against the previous value and triggers if the current value is less than the threshold specified in the argument field and is also less than the previous value. This operator can be specified at the command line by entering “-<<”.

Increase of
Checks the current value against the previous value and triggers if the difference is greater than or equal to the threshold value. This operator can be specified at the command line by entering “- >=”.

% increase of
Checks the current value as a percentage of the previous value and triggers if current value is greater than or equal to the specified value. This operator can be specified at the command line by entering “**%** >=”.

Changes by
Measures an absolute difference between the previous and current value. This operator can be specified at the command line by entering “+ >=”.

Outside range
Compares each sample against the defined range limits, and triggers if the current value falls outside the specified range. Because the range check is made against only the current sample, this type of monitor can trigger after any sample. To define the acceptable range, enter a lower and upper limit, separated by a dash, in the argument field. This operator can be specified at the command line by entering “<>”.

String Operator Group

To define the threshold value, select one of the following options and enter a string (which can include wildcard characters) in the argument field.

(never) Indicates the response level does not trigger a response. This value only affects the specific response level, not the entire monitor.

Equal to Records the value of each sample, compares it to the specified value, and triggers if the current value is equal to the specified value. Because it checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering "==".

Not equal to Records the value of each sample, compares it to the specified value, and triggers if the current value is not equal to the specified value. Because it checks only against the current sample, this type of threshold can trigger after any sample. This operator can be specified at the command line by entering "**!=**".

Matches Checks the value returned by the query script value against the threshold value provided for user-defined and asynchronous string monitors. A response is triggered if the response matches the value provided. The query program is named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. This operator can be specified at the command line by entering "**=~**".

Mismatches Checks the value returned by the query script value against the threshold value provided for user-defined and asynchronous string monitors. A response is triggered if the response does not match the value provided. The query program is named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. This operator can be specified at the command line by entering "**!~**".

Changes to Compares the current value (returned by the query script value and compared to the threshold value provided for user-defined and asynchronous string monitors) against the previous value. A response is triggered if the current value matches the value provided and if the previous value did not match the value provided. The query program is named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. This operator can be specified at the command line by entering “->”.

Changes from Compares the current value (returned by the query script value and compared to the threshold value provided for user-defined and asynchronous string monitors) against the previous value. A response is triggered if the current value does not match the value provided and if the previous value did match the value provided. The query program is named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog. This operator can be specified at the command line by entering “-<”.

Status Operator Group

To define the status value, select one of the following options. The resource to be monitored is named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog.

(never) Indicates the response level does not trigger a response. This value only affects the specific response level, not the entire monitor.

Is up/available Checks the status of system resources, such as daemons, hosts, and print queues. A response is triggered if the resource is available. This operator can be specified at the command line by entering “==”.

Is down/unavailable Checks the status of system resources, such as daemons, hosts, and print queues. A response is triggered if the resource is not available. This operator can be specified at the command line by entering “==”.

Becomes available

Checks the status of system resources, such as daemons, hosts, and print queues. A response is triggered if the resource was not available in the previous sample and is available in the current sample. This operator can be specified at the command line by entering “>”.

Becomes unavailable

Checks the status of system resources, such as daemons, hosts, and print queues. A response is triggered if the resource was available in the previous sample and is not available in the current sample. This operator can be specified at the command line by entering “->”.

Same as

Checks the return string. A response is triggered if the return string matches the string that is provided. To define the threshold value, enter a string in the argument field. This operator can be specified at the command line by entering “>”.

Different from

Checks the return string. A response is triggered if the return string is not the same as the one provided. This operator can be specified at the command line by entering “>”.

File Operator Group

To define the status value, select one of the following options. The files to be monitored are named in the **Add Monitor to TME 10 Distributed Monitoring Profile** dialog.

(never)

Indicates the response level does not

trigger a response. This value only affects the specific response level, not the entire monitor.

Files are identical

Compares two files. This option has a default value of **same**, and if the files are identical, TME 10 Distributed Monitoring returns a value of **same**. If the returned value and default value are identical, TME 10 Distributed Monitoring triggers a response. This operator can be specified at the command line by entering “==”.

Files are different

Compares two files. This option has a default value of **diff**, and if the files are identical, TME 10 Distributed Monitoring returns a value of **diff**. If the returned value and default value are identical, TME 10 Distributed Monitoring triggers a response. This operator can be specified at the command line by entering “==”.

Files become identical

Compares two files. This option has a default value of **same**, and if one of the files becomes identical to the other, TME 10 Distributed Monitoring returns a value of **same**. If the returned value and default value are identical, TME 10 Distributed Monitoring triggers a response. This operator can be specified at the command line by entering “->”.

Files become different

Compares two files. This option has a default value of **diff**, and if one of the files becomes different than the other, TME 10 Distributed Monitoring returns a value of **diff.** If the returned value and default value are identical, TME 10 Distributed Monitoring triggers a response. This operator can be specified at the command line by entering “->”.

Index