NASTEL™

Driving Business Transaction Performance

M6/OSM 600.002.1

AutoPilot® M6 Plug-in for Operating System Monitors

Installation and User's Guide

Version 6.0

PUBLISHED BY:

RESEARCH & DEVELOPMENT

NASTEL TECHNOLOGIES, INC.

48 SOUTH SERVICE ROAD, SUITE 205

MELVILLE, NY 11747

DOCUMENT TITLE: **AUTOPILOT® M6 PLUG-IN FOR OPERATING SYSTEM MONITORS**

INSTALLATION AND USER'S GUIDE

DOCUMENT RELEASE DATE: **MAY 2022**

NASTEL DOCUMENT NUMBER: **M6/OSM 600.002.1**

ACKNOWLEDGEMENTS:

THE FOLLOWING TERMS ARE TRADEMARKS OF NASTEL TECHNOLOGIES CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: AUTOPILOT/IT, AP-WMQ, AUTOPILOT ENTERPRISE, AUTOPILOT/WEB, M6 FOR WMQ, AUTOPILOT M6, MQCONTROL.

THE FOLLOWING TERMS ARE TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: IBM, MQ, WIN-OS/2, AS/400, OS/2, DB2, AND AIX.

THE FOLLOWING TERMS ARE TRADEMARKS OF HEWLETT-PACKARD IN THE UNITED STATES OR OTHER COUNTRIES OR BOTH: OPENVIEW, HP-UX.

COMPAQ, THE COMPAQ LOGO, ALPHASERVER, COMPAQ INSIGHT MANAGER, CDA, DEC, DECNET, TRUCLUSTER, ULTRIX, AND VAX REGISTERED IN U.S. PATENT AND TRADEMARK OFFICE. ALPHA AND TRU64 ARE TRADEMARKS OF COMPAQ INFORMATION TECHNOLOGIES GROUP, L.P IN THE UNITED STATES AND OTHER COUNTRIES.

SNMPC, SNMPC, WORKGROUP, AND SNMPC ENTERPRISE ARE TRADEMARKS OF CASTLE ROCK COMPUTING IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

JAVA AND THE JAVA LOGOS ARE TRADEMARKS OF SUN MICROSYSTEMS INC. IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

INSTALLANYWHERE IS A REGISTERED TRADEMARK OF ZEROG SOFTWARE IN THE UNITED STATES OR OTHER COUNTRIES, OR BOTH.

THIS PRODUCT INCLUDES SOFTWARE DEVELOPED BY THE APACHE SOFTWARE FOUNDATION (HTTP://WWW.APACHE.ORG/). THE JAKARTA PROJECT" AND "TOMCAT" AND THE ASSOCIATED LOGOS ARE REGISTERED TRADEMARKS OF THE APACHE SOFTWARE FOUNDATION.

INTEL, PENTIUM AND INTEL486 ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION IN THE UNITED STATES, OR OTHER COUNTRIES, OR BOTH.

MICROSOFT, WINDOWS, WINDOWS, WINDOWS XP, THE WINDOWS LOGOS, AND SQL SERVER ARE REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION.

UNIX IS A REGISTERED TRADEMARK IN THE UNITED STATES AND OTHER COUNTRIES LICENSED EXCLUSIVELY THROUGH X/OPEN COMPANY LIMITED.

"LINUX" AND THE LINUX LOGOS ARE REGISTERED TRADEMARKS OF LINUS TORVALDS, THE ORIGINAL AUTHOR OF THE LINUX KERNEL. ALL OTHER TITLES, APPLICATIONS, PRODUCTS, AND SO FORTH ARE COPYRIGHTED AND/OR TRADEMARKED BY THEIR RESPECTIVE AUTHORS.

SCO CUSA, SCO DOCTOR, SCO DOCTOR FOR NETWORKS, SCO DOCTOR LITE, SCO GLOBAL ACCESS, SCO MPX, SCO MULTIVIEW, SCO NIHONGO OPENSERVER, SCO OK, THE SCO OK LOGO, SCO OPENSERVER, SCO OPEN SERVER, SCO PORTFOLIO, SCO POS SYSTEM, SCO TOOLWARE, AND THE WORLD NEVER STOPS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF CALDERA INTERNATIONAL, INC. IN THE U.S.A. AND OTHER COUNTRIES. ALL RIGHTS RESERVED.

OTHER COMPANY, PRODUCT, AND SERVICE NAMES, MAY BE TRADEMARKS OR SERVICE MARKS OF OTHERS.

Table of Contents

Figures

Tables

Chapter 1: Introduction

Welcome to the *Nastel AutoPilot M6 Operating System Monitor Guide*. This guide describes installation, and use of the Windows Performance Monitor and the UNIX Operating System (OS) Monitors. Please review this guide carefully before installing the product.

The *AutoPilot M6 Operating System Monitor* for Windows will hereinafter be referred to as the Performance Monitor and the UNIX OS Monitors will hereinafter be referred to as the OS Monitors.

1.1 How This Guide is Organized

Chapter 1: Identifies the users and history of the document, as well as additional and alternate documents. System requirements are outlined in addition to supplying support and reference information.

Chapter 2: Contains a brief functional description of the Performance Monitor.

Chapter 3: Provides instructions for new installations of the Performance Monitor for Windows.

Chapter 4: Provides instruction for upgrading and new installations of the UNIX OS Monitors.

Chapter 5: Provides post-installation set-up and starting instructions.

Chapter 6: Provides counter, instance and metrics information.

Appendix A: Provides a detailed list of all reference information required for the installation of AutoPilot.

Appendix B: Contains conventions used in this document.

Glossary: Contains a listing of unique and common acronyms and words and their definition.

1.2 History of This Document

Table 1-1. Document History

Release Date:	Document Number	For AutoPilot Version	Summary
May 2008	M6/OSM	M6 600.001	Initial release
August 2010	M6/OSM 600.002	All M6 versions	New option for the Windows Agent
May 2022	M6/OSM 600.002.1	All M6 versions	Added Internet Explorer caveat to System Requirements. Changed title to *AutoPilot M6 Plug-in for Operating System Monitors Installation and User's Guide*

1.2.1 User Feedback

Nastel encourages all Users and Administrators of AutoPilot to submit comments, suggestions, corrections, and recommendations for improvement for all AutoPilot documentation. Please send your comments via Post/Mail, or by *e*-mail. Send messages to: support@nastel.com. You will receive a written response, along with status of any proposed change, update, or correction.

1.3 Related Documents

The complete listing of related and referenced documents is listed in Appendix A of this guide.

1.4 Release Notes

See README.HTM files on installation media or AutoPilot installation directory.

1.5 Intended Audience

The *Nastel AutoPilot M6 Operating System Monitor Guide* is intended for use by installers and administrators of Nastel's AutoPilot and AutoPilot M6 OS Monitors. There are three user groups defined for the purpose of installation and use.

- **Installer:** Should be familiar with Java Run Time Environment 1.5.1 (JRE 1.5.1) or higher (included in AutoPilot M6.) Software installation procedures and set-up on Windows and UNIX. Basic understanding of TCP/IP.
- **Administrator:** Should have a working knowledge of middleware, TCP/IP and systems management. An understanding of Java Runtime Environment (JRE), TCP/IP, and installation procedures for the platform where AutoPilot M6 is installed.
- **User:** Requires only local operating system operations knowledge and basic knowledge of AutoPilot.

1.6 System Requirements

The Performance Monitor for Windows installation requires approximately 1M disk space. The OS Monitors for UNIX installation require approximately 500K disk space.

M6 Web Server has been tested with Internet Explorer 5.5 ServicePack1, Netscape 4.0 and Java browser plug-in 1.5.1 on Windows environments. All references in this document to Internet Explorer are subject to Internet Explorer being available. M6 console works with Internet Explorer if it is available. Customers who plan to move away from this older functionality are encouraged to consider using the equivalent functionality that is available within Nastel XRay. For each component, XRay's sensors can provide metrics about message processing and point to backlogs in the processing pipeline by showing lag times in message streaming and indexing.

1.7 Terms and Abbreviations

A list of Terms and Abbreviations used in this document is located in the Glossary.

1.8 Technical Support

If you need technical support, you can contact Nastel Technologies by telephone or by e-mail. To contact Nastel technical support by telephone, call **(800) 963-9822 ext. 1**, if you are calling from outside the United States dial **001-631-761-9100 (x9190)** or **+44 20 7084 6205**. To contact Nastel technical support by e-mail, send a message to mailto:support@nastel.com. To access the Nastel automated support system (User ID and Password required), go to: http://support.nastel.com/. Contact your local AutoPilot Administrator for further information.

1.9 Conventions

Refer to Appendix B for typographical and naming conventions used in all AutoPilot documentation.

Chapter 2: About M6 OS Monitors

2.1 Performance Monitor for Windows

The M6 Performance Monitor for Windows collects Windows performance counter data and publishes the data to the AutoPilot Operating System (OS) Monitor Expert. Windows performance counters provide data on the behavior of performance objects on a computer. Performance objects (also called categories) include memory, processors, processes, etc. They can also include application-level objects, as defined by specific applications. Each performance object has one or more counters associated with it. Some counters can provide individual data values for specific instances (for example, Process counters).

The Performance Monitor publishes performance counter data at a user-defined interval. For counters that provide values for specific instances, the grouping, or hierarchy, of the performance data can also be configured. The hierarchy that Windows uses for publishing this data is:

Performance Object
- └ Performance Counter
  - └ Counter Instance

This is referred to as "By Counter". This hierarchy is useful for monitoring the values for specific performance counters across all active instances (e.g. monitoring the current value of a specific set of performance counters for all processes). As an example, for the counter *Handle Count* in performance object *Process*, the data has the following hierarchy:

Process
- └ Handle Count
  - └ notepad
  - └ notepad#1
  - └ svchost
  - └ …

where the individual process instances are under Handle Count.

With this hierarchy, however, the metrics for individual instances are spread out. To monitor the behavior of specific instances, the following hierarchy is better:

Performance Object
- └ Counter Instance
  - └ Performance Counter

This is referred to as *By Instance*. Using the Process object mentioned above, the data hierarchy would be:

Process
- └ notepad
  - └ Handle Count
  - └ ID Process
  - └ Thread Count
  - └ …

This hierarchy groups the metrics for a specific process instance together.

The hierarchy that the Performance Monitor uses to publish the data is controlled by specifying the –C or –I arguments for grouping by Counter or grouping by Instance, respectively, at startup (or installation if installing the performance monitor as a service).

The Performance Monitor reads a configuration file to determine what performance counters to collect and publish. This configuration file has the form of an `.ini` file, with a separate group for each performance object. The performance counter entries must adhere to the specific performance counter syntax, which can be complicated. To aid the creating and maintenance of these configuration files, a utility called the AutoPilot M6 Performance Monitor for Windows Configuration Utility (Configuration Utility) is provided. The process monitor can also be configured to monitor all system services and some general system information. Refer to Section 6.1.1 for detailed information about the process monitor metrics.

2.1.1 Configuration Utility

The Configuration Utility (apperfmoncfg.exe) is a graphical tool that can be used to define the specific Windows performance counters that the performance monitor is to collect. The Configuration Utility provides a way of selecting the performance metrics to collect without having to worry about the performance counter syntax. Refer to Section 3.5 for detailed information about the Configuration Utility.

2.2 OS Monitor for UNIX

The OS Monitor for UNIX captures the processor performance, resources, and file system metrics from Unix server or workstations. Using minimal system resources, the software continuously monitors the `syslog` file, process and processor performance. OS Monitor collects the system metrics and publishes them to AutoPilot.

2.2.1 System Configuration & File System

System configuration data that rarely changes, or disk utilization data that changes gradually, may need to be monitored only once a day. Use business views to generate message/alert if disk utilization reach maximum threshold, for example if threshold > 80%, severity=warning.

2.2.2 Processor Performance

Dynamic performance statistics, with characteristics that vary with the time of day, should be monitored many times each day to give an accurate picture.

Some of the metrics described are:

- Total physical memory available
- Amount of memory applications are using
- Amount of memory that is free
- Amount of processes that will be running
- Amount of swap memory available
- Amount of swap memory used
- Amount of users currently logged on the system

2.2.2.1 Syslog (System Logger)

OS Monitor parses the Syslog message into host name, facility (category), severity, message text or time of day, and publishes the message (fact) to AutoPilot console.

2.3 OS Monitor Architecture

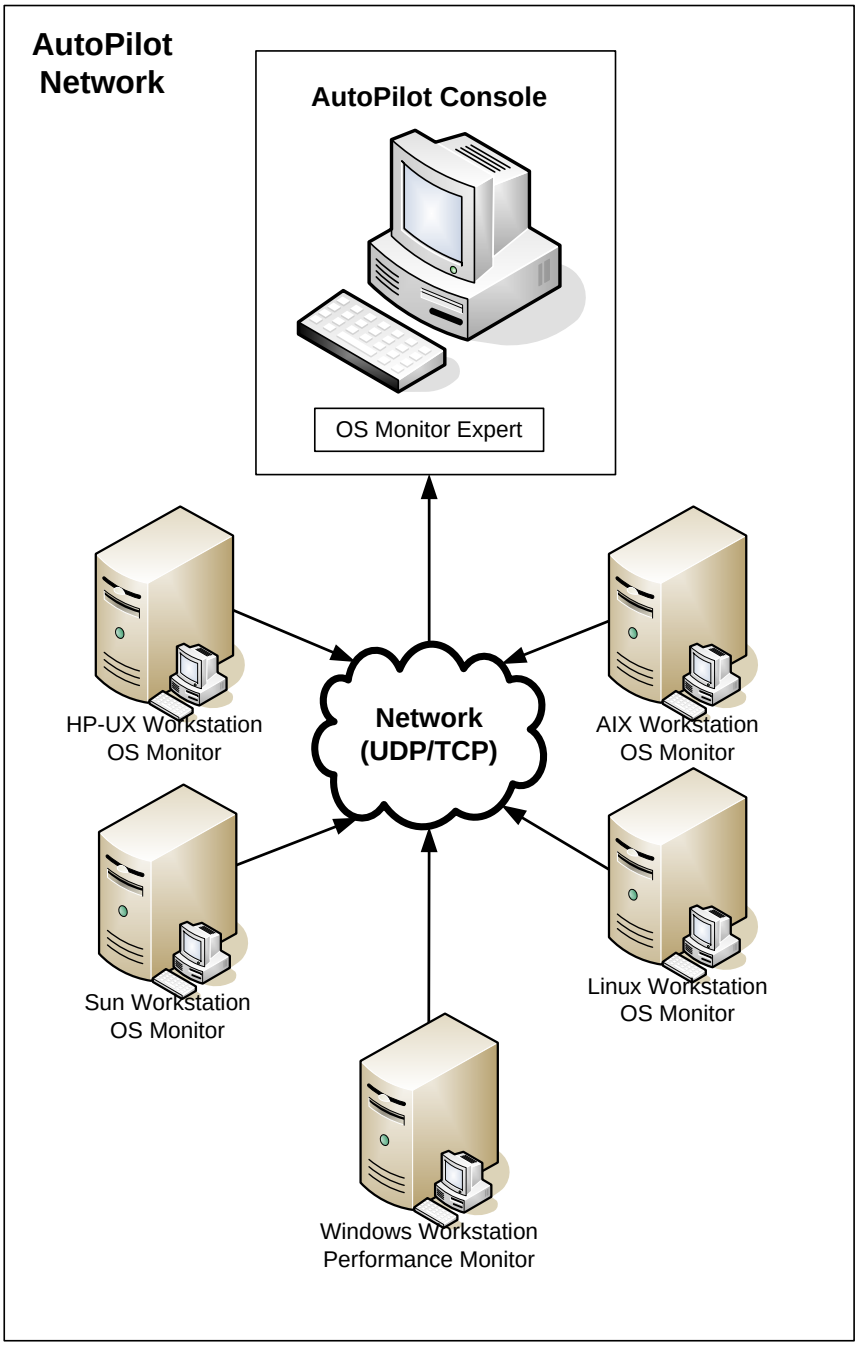

Figure 2-1. OS Monitor Architecture

Chapter 3: Installation and Configuration on Windows

This chapter contains general information related to M6 Performance Agent installation on Windows.

3.1 Installation Materials

The installation media contains all required components for M6 Performance Agent installation.

3.2 System Requirements

- Windows 2000, 2003, XP or Vista
- AutoPilot M6
- Application Instrumentation Module (AIM) 6.0.4 or higher

3.2.1 Installation Requirement

Ensure that an AutoPilot managed node is installed and configured and TCP/IP connectivity is available from the AutoPilot managed node to the server where metrics will be collected by the Performance Agent process wrapper.

3.3 Documentation

Prior to installation, review all text files and installation procedures provided on the media. It is recommended that all installation related materials are printed so that the installer can review them prior to installation and better understand and follow the detailed instructions within. The following files are included in the installation package:

- README.TXT
- Installation and User's Guide: M6/OSM [*VERSION*].[*RELEASE*].PDF. The guide Suffix will vary with software release and document revisions. For example: M6/OSM 600.001, supports version 6, document release 1.

3.4 Installation

This section provides detailed instructions for installing M6 Performance Agent on Windows.

3.4.1 Performance Agent Installation

	**IMPORTANT!**	If a previous version of the Windows OS Monitor is installed, `napsvc` must be uninstalled manually prior to installing the Performance Agent.

1. Insert the AutoPilot M6 Performance Agent installation media.
2. Follow the screens by accepting the license and clicking **Next** to proceed to the next screen.
3. The default Destination folder is: `C:\Program Files\nastel\apperfmon`, but can be changed.
4. After installation is complete, it is recommended to enable *Launch AutoPilot Performance Monitor 6.0 Configuration*. This will automatically display the Nastel AutoPilot Performance Monitor for Windows Configuration Utility.
5. If **Run as Service** was selected on the *Configure Performance Agent* screen, ensure that the service is *Started* by either rebooting or using the *Services* screen. If **Run as Service** was not selected when installing the Performance Agent, it can be accomplished manually. (Refer to Section 3.7.)

3.5 Configuring the Performance Monitor

Before using the Performance Monitor, it must be configured. This consists of defining what performance counters and instances (for those counters that provide data on a per-instance basis) to monitor. This can be done using the GUI-based Configuration Utility provided (recommended) with this application or manually (only if familiar with counter syntax).

3.5.1 GUI-based Configuration

To aid in maintenance configuration, the AutoPilot Performance Monitor for Windows Configuration Utility, hereinafter called Configuration Utility, is available. It is a GUI-based utility for selecting objects, counters, and instances to monitor and build correct counter paths.

The Configuration Utility can be automatically started after installation by clicking *Launch AutoPilot Performance Monitor 6.0 Configuration* on the last installation screen. You can start the Configuration Utility manually by one of two methods:

1. Running `apperfmoncfg.exe` or
2. Navigating to **Start>Programs>Nastel AutoPilot Performance Monitor>Agent Configuration**

When the Configuration Utility starts, it attempts to load the last configuration file that was processed (if there is one), or the default configuration file:

([*InstallDir*]`\apperfmon\config\apperfmon.ini`).

If the file does not exist, you are prompted as to whether you'd like to create the file. Selecting **Yes** sets the configuration file name to this file. Selecting **No** leaves the configuration file blank; which can be set later.

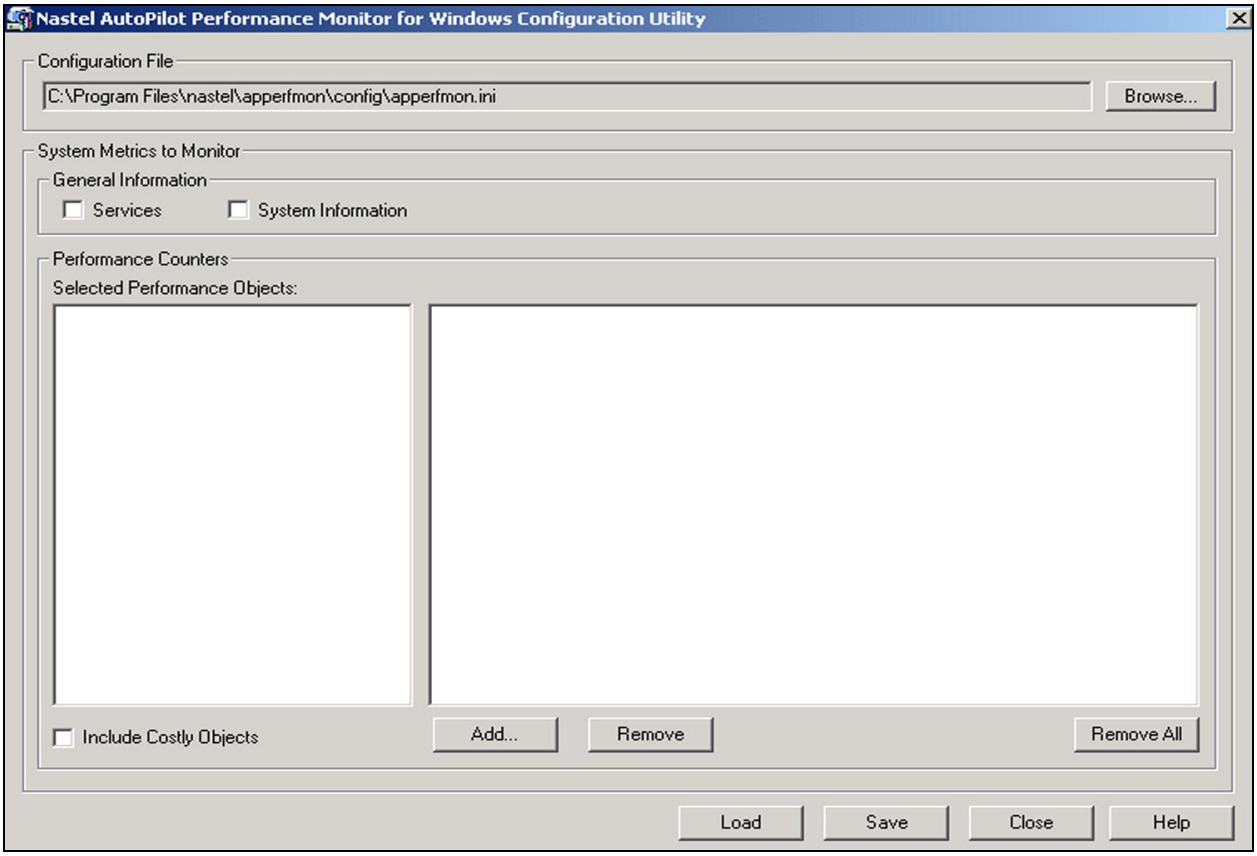

Figure 3-1. Configuration Utility

To use the configuration utility, perform the following steps: (Refer to Figure 3-1 above for steps 1-6.)

	**IMPORTANT!**	The Performance Agent must be restarted for configuration file changes to take effect.

	**NOTE:**	For additional information about the configuration utility, click **Help** on the lower right-hand side of the *Configuration Utility* screen.

1. To select a different configuration file, click **Browse** and locate the file you would like to load.
   a. If the file already exists, you are prompted to load it now. Selecting **Yes** loads the file. Selecting **No** leaves the current data displayed. If the file is going to be loaded and the currently loaded configuration has changed, you are prompted to save the current changes. Click **Load** to load (or reload) the current file.
   b. If the file does not exist, you are prompted to clear the current configuration. Selecting **Yes** clears the currently displayed objects and counters. Selecting **No** leaves the current data displayed.
2. If you want to display all system services information, enable *Services* under *System Metrics to Monitor*. Services information is grouped by service type class (process, driver, adapter). Refer to Table 6-1 for a list of all services published.
3. If you want to display general system information, enable *System Information* under *System Metrics to Monitor*. General system information includes platform, memory, file system and other general information. Refer to Table 6-2 for a list of all system information published.
4. If you do not want to add any counters or instances, skip to step 9.
5. Enable *Include Costly Objects* if you want to include performance counters that require a large amount of processor time and/or memory overhead.

	**NOTE:**	If *Include Costly Objects* is enabled, the display of the *Performance Counter Browser* screen may be delayed.

6. Click **Add** to display the *Performance Counter Browser* screen to select new or additional counters and instances.

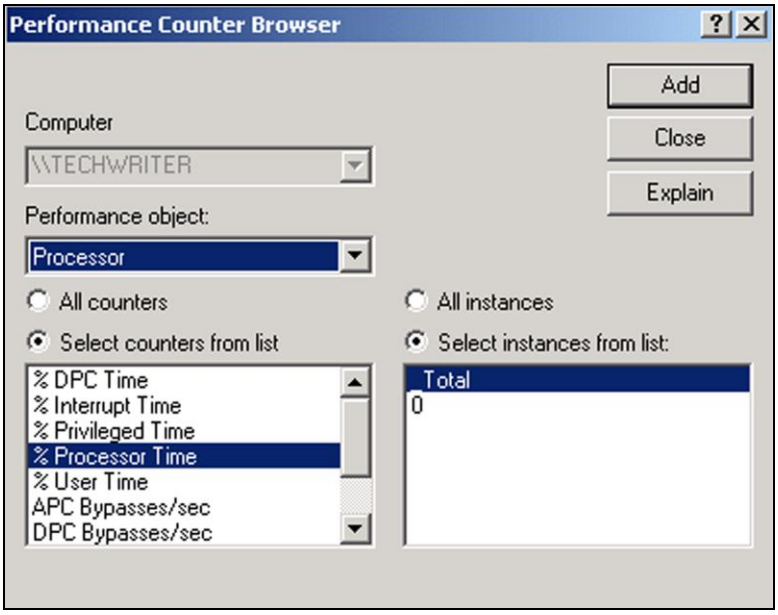

Figure 3-2. Performance Counter Browser

7. If you want to select all counters:
   a. Select **All counters**.
   b. If instances are available, either select **All instances** or **Select instances from list**. If **Select instances from list** was selected, select the desired instance. To select multiple instances, hold down the **Ctrl** key while selecting your desired instances.
   c. Click **Add**. This will add all the counters and selected instances to the list of counters being monitored for the selected object.
   d. Click **Close** to return to the *Configuration Utility* screen.
8. If you want to select individual counters:
   a. Select *Select counters from list* and select the desired counters. To select multiple counters, hold down the **Ctrl** key while selecting your desired instances.
   b. If instances are available, either select **All instances** or **Select instances from list**. If **Select instances from list** was selected, select the desired instance. To select multiple instances, hold down the **Ctrl** key while selecting your desired instances.
   c. Click **Add**. This will add all the selected counters and instances to the list of counters being monitored for the selected object.
   d. Click **Close** to return to the *Configuration Utility* screen.
9. On the *Configuration Utility* screen, if you want to remove any counters, select the counter to be removed and click **Remove**. If you want to remove all counters, click **Remove All**.
10. When you are finished, click **Save** to save your changes to the selected configuration file.
11. Click **Close** to close the *Configuration Utility* screen.
12. Restart the Performance Agent for changes to take effect.

3.5.2 Configuration File Format

	**CAUTION!**	Do not manually configure the Performance Agent unless you are knowledgeable about the counter syntax used in this application.

A configuration file must be created that has the structure of an .ini file. The general format of the file looks like:

[Performance Object 1]

Performance Object 1 Counter1

Performance Object 1 Counter2

…

[Performance Object 2]

Performance Object 2 Counter1

…

There is a separate group for each performance object being monitored and an entry for each counter for that performance object. See below for an example of a configuration file.

[Processor]

\Processor(_Total)\% Processor Time

[Memory]

\Memory\Available Kbytes

[Process]

\Process(*)\Working Set

The syntax of the counters, which have the general format of a file path, must match the specific syntax required by Windows.

3.6 Command-Line Arguments

Table 3-1 lists all command-line arguments applicable to the Performance Monitor for Windows.

Table 3-1. Command-Line Arguments

Argument	Description
`-console`	Start Performance Monitor as console program.
`-install`	Install Performance Monitor service instance.
`-update`	Update Performance Monitor service instance parameters.
`-remove`	Remove Performance Monitor service instance.
`-h<Host>`	Host name or IP address of the node where the AutoPilot Performance OS Monitor Expert is deployed. THIS ARGUMENT IS REQUIRED.
`-s<Port>`	AutoPilot Performance Monitor Expert port number to which metrics are sent. Default: 6000
`-i<Interval>`	Generate Performance metrics every <*Interval*> seconds. Default: 30
`-f<CfgFile>`	Configuration file defining Windows performance counters to collect. Default: `[InstallDir]\config\apperfmon.ini`
`-C|-I`	Determines grouping of fact data for counters supporting instances: -C = By Counter -I = By Instance Default: By Counter (matches Windows grouping)
`-U|-T`	Communication protocol between Performance Monitor and fact collector: -U = UDP -T = TCP Default: UDP
`-l`	Log AutoPilot Performance Monitor Expert messages to `[InstallDir]\log\` (creates separate file per Performance Object category)
`-t`	Enable trace.
`-?`	Display usage.

Usage: `apperfmon.exe [-console | -install | -update | -remove] -h<Host> [-s<Port>] [-i<Interval>] [-f<CfgFile>] [-C | -I] [-U | -T] [-l] [-t] [-?]`

Figure 3-3 is an example of using the `-console` command-line argument for displaying counters.

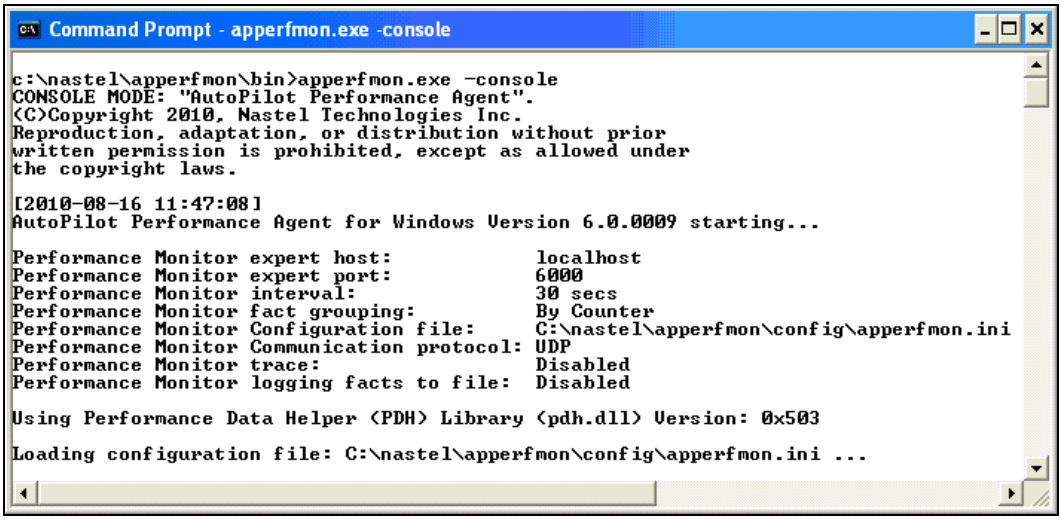

Figure 3-3. Example of Using –console Argument

3.7 Defining Windows Services Instances

This section describes how to install a service instance, if not already done during installation, and how to install, modify and remove additional windows monitors as services.

3.7.1 Install Service Instance

The Performance Monitor can be installed as a Windows service. Several such instances can be defined, where each service instance is defined for a different AutoPilot Performance OS Monitor Expert host. The basic syntax to install a service instance is:

```
apperfmon.exe –install –hMyHost
```

This will create a windows service called “AutoPilot Performance Agent (MyHost)”. This instance will publish the metrics to *MyHost*, using the defaults for the other startup parameters. To install an instance to use startup parameters other than the defaults, include the appropriate arguments on the command line when installing the service instance. For example, to install a service instance to use a configuration file other than the default, enter:

```
apperfmon.exe –install –hMyHost "-fC:\Some Dir\apperfmon2.ini"
```

3.7.2 Modify Service Instance

To modify the startup parameters for a service instance that is installed, use the –update option. The generic syntax for this is:

```
apperfmon.exe –update –hMyHost …
```

where the –h argument is followed by the corresponding arguments for the options to change. (Refer to Table 3-1 above.)

For options that are common across all service instances, the environment variable “APPERFMON_OPTIONS” can be defined containing the startup options that are in common for all instances. If there is a conflict between the variable and the service instance settings defined using the above mechanism, the instance settings override those in the environment variable.

3.7.3 Remove Service Instance

To remove a specific service instance, use the following:

```
apperfmon.exe –remove –hMyHost
```

Removing service instances may require a system reboot if the system cannot immediately remove the service. If the service cannot be immediately removed, Windows sets its startup method to "Disabled" and marks the service for removal, which is done at the next reboot.

This Page Intentionally Left Blank

Chapter 4: Installation and Configuration on UNIX

This section provides detailed instructions for installing AutoPilot/OS Monitor on the compatible UNIX platforms. The AutoPilot/OS Monitor can be installed in the following UNIX environment:

- AIX
- Solaris
- HP-UX
- Linux

4.1 Installation Materials

The installation media contains all required components for AutoPilot/OS Monitor installation.

The UNIX-OS Monitor installation package (file `NSQOSMON.tar.Z`) contains the following files:

- `nsqosmon`
- NSQOSMON_*OS_PackageVersion*.txt (ReadMe)
- NSQOSMON_*OS_PackageVersion*_SCREENSHOTS.doc
- NASTEL APIs
  - libnsqapi.sl (HP-UX), libnsqapi.so (Solaris and Linux), libnsqapi.a (AIX)
  - libnsqsign.sl (HP-UX), libnsqsign.so (Solaris and Linux), libnsqsign.a (AIX)
  - libnsqclnx.sl(HP-UX), libnsqclnx.so (Solaris and Linux), libnsqclnx.a (AIX)

The package file name is `NSQOSMON_<os>_<packageVersion>.tar.Z` (or `.zip`)
(`.tar.Z` for HP-UX, AIX, Solaris, or `.zip` for Linux)

4.2 Installation Requirements

Ensure that an AutoPilot managed node software component is installed and configured, and TCP/IP connectivity is available on the AutoPilot server where OS metrics will be collected by the OS Monitor process wrapper.

4.3 Documentation

Prior to installation, review all text files and installation procedures provided on the media. It is recommended that all installation-related materials are printed so that the installer can review them prior to installation, and better understand and follow the instructions. The following files are included in the installation package:

- NSQOSMON_*OS_PackageVersion*.txt (ReadMe)
- Installation and User's Guide: M6/OSM *version.release*.pdf. The guide version and release numbers will vary with software version and document revisions. For example: M6/OSM 600.001, supports version 6, document version 1.

4.4 Installation on HP-UX, AIX, Solaris, and Linux

All references to <*version*> should be substituted with the current AutoPilot/OS Monitor version number (Example: The version 1.2.2 LINUX file would be `NSQOSMON_LINUX_1.2.2.zip`).

	**NOTE:**	The following procedure applies to all UNIX platforms.

1. Logon as root.
2. Upload the file `NSQOSMON_<os>_<version>.tar.Z` (for Linux `.zip`) in binary to the `/tmp` directory on your UNIX system.
3. Extract the files to the *`install_dir`* directory. The OS Monitor is usually installed at `/opt`.
   Example: If *`install_dir`* is `/opt`:

   ```
   cd /opt
   ```

   Linux: `unzip /tmp/OSMON_LINUX_<ver>.zip`

   HP-UX, AIX, Solaris:

   ```
   uncompress /tmp/NSQOSMON_<os>_<ver>.tar.Z
   tar_xvf /tmp/NSQOSMON_<os>.tar
   ```

   This will put the executable and library files into directories `/opt/nastel/os-unix/bin` and `/lib`, respectively, which may contain previously installed AutoPilot Plug-ins.
4. Remove the tar or zip file:
   `rm /tmp/OSMON_<os>_<ver>.tar` (or `.zip`)
5. Change owner if required:
   `chown -R root:sys /opt/nastel/os-unix` (or use `owner:group` = `mqm:mqm`).
6. Set up the shared library search path in a startup script and/or the profile of the user ID that will be used to run the OS monitor (typically root). The AutoPilot/OS Monitor shared libraries (libnsq*.xx) are located in <*`osmon_lib`*>, where <*`osmon_lib`*> = `/opt/nastel/os-unix/lib`

   Solaris: LD_LIBRARY_PATH=$LD_LIBRARY_PATH <*`osmon_lib`*>
   export LD_LIBRARY_PATH
   `.xx = .so` (shared object)

   Linux: same as Solaris

   AIX: LIBPATH=$LIBPATH:<*`osmon_lib`*>
   export LIBPATH
   `.xx = .a` (archive)

   HP-UX: SHLIB_PATH=$SHLIB_PATH:<*`osmon_lib`*>
   export SHLIB_PATH
   `.xx = .sl` (shared library)
7. Configure the AutoPilot UNIX OS Monitor expert as a process wrapper in the AutoPilot console, specifying the UDP and/or TCP ports. See the *AutoPilot User's Guide* for additional details.

8. **For Linux only:**

   Determine special monitor startup options:

   a. Determine if the **-M** multi-line option should be used:

      Execute the sar command:

      ```
      sar -cuwW 1 1
      ```

      If the column headings are spread over multiple lines like this:

      ```
      12:43:37 PM     cswch/s
      12:43:38 PM     11.00

      12:43:37 PM     CPU     %user     %nice   %system   %idle
      12:43:38 PM     all      1.00      0.00      1.00    98.00
      ```

      then you must use the **-M** multi-line option when you start the monitor.

      If the column headings are on one line like this:

      ```
      12:43:37 PM     cswch/s     CPU     %user     %nice   %system     %idle
      12:43:38 PM     11.00       all      1.00      0.00      1.00     98.00
      ```

      then do not use the **-M** option.

   b. Determine the sar disk statistics **-R** option value:

      Execute the sar command or check "man sar" to find the option that gives the disk statistics.

      Example:

      ```
      sar -b 1 1
      12:53:52 PM       tps      rtps      wtps   bread/s   bwrtn/s
      12:53:53 PM      0.00      0.00      0.00      0.00      0.00
      Average:         0.00      0.00      0.00      0.00      0.00
      ```

      In this case, use -Rb when you start the monitor with -a or -c.

      The default value is -Rr.

9. Start UNIX OS monitor process `nsqosmon` as user root or other ID from the command line or from a script. See the options in the next section.
   The root or other ID used to run `nsqosmon` must have:
   - **read** permission to file `/var/log/messages` (or other syslog) and directory `/proc` (AIX, Linux, Solaris)
   - **read** permission to pseudo-file `/dev/Kmem` (AIX and HP-UX)
   - **execute** permission to `/usr/bin/sar`

10. **Verification:** Observe the UNIX OS Monitor metrics in the AutoPilot console.

4.5 Starting OS Monitor in UNIX

Using the `nsqosmon` program:

Usage: `nsqosmon -h`*`HostName`* `[-s`*`Port`*`] [-i`*`Interval`*`] [-f] [-c] [-r`*`SarInterval`*`] [-c`*`SarCount`*`] [-p] [-k] [-e] [-d] [-l`*`Syslog File`*`] [-a] [-b] [-U|-T] [-t] [-m`*`Mask`*`] [-R`*`sar_char`*`] [-M] [-D] [>`*`outfile`* `2>&1]`

Table 4-1. OS Monitor Start Options

Argument	Description
-h*HostName*	Host name or IP address of the node where the AutoPilot/OS Monitor expert (process wrapper) is deployed.
-s*Port*	Port number to which the metrics are sent. (The AutoPilot process wrapper receives the metrics and posts them to the facts board.) Default: 20000.
-i*Interval*	Generate OS metrics every Interval seconds. Default: 5
-f	Monitor the file system.
-c	Report on system configuration and monitor processor performance.
-r*SarInterval*	System Activity Report (SAR) sampling interval, in seconds. Default: 5
-c*SarCount*	SAR number of samples. Default: 1
-p	Monitor processes.
-k	**AIX only:** Monitor Kernel processes. **HP-UX, Solaris, Linux:** -p option includes Kernel processes.
-e	Clear the facts if the process terminates; otherwise, set process fact State = Stopped.
-d	Disable monitoring of the syslog file.
-l*Syslog file*	Name of the syslog to monitor versus using the default syslog.
-a	Activate all the monitors: file system, processes, syslog, SAR.
-b	Run the monitor as a background task. Default: runs on console.
-U | T	Communication protocol to use between OS Monitor and expert: -U=UDP, -T=TCP. Default: UDP
-t	Trace non-process fact messages.
-m*Mask*	Mask to trace process fact messages for process names that start with a mask pattern. Use '*' to trace all. Default: all Example: `-mov` to trace messages for all processes whose names start with characters 'ov'.
-R*sar_char*	**Linux only:** SAR disk statistics option character. Default: r On some Linux systems, this option value is 'b'. See installation procedure, step 8.
-M	**Linux only:** A flag to indicate that sar command with multiple options on this system generates a multi-line versus single line report. Default: single. You must determine if –M option should be used. See installation procedure, step 8.
-D	Debug/trace statements.
>*outfile* 2>&1	Redirect file for stdout and stderr.

	**NOTE:**	See the product release notes for any enhancements in usage or instructions, or type **`nsqosmon`** to view the usage display.

Startup Examples for Individual Monitors

- Activate process monitor only, disable syslog monitor, generate facts at 30 second intervals:
  ```
  nsqosmon -h11.0.0.114 -p –i30 -d
  ```
- Activate system configuration and processor performance monitor only:
  ```
  nsqosmon -h11.0.0.114 -c
  ```
- Activate file system monitor only:
  ```
  nsqosmon -h11.0.0.114 -f
  ```
- Activate syslog monitor only, with specific syslog:
  ```
  nsqosmon -h11.0.0.114 -l/tmp/syslog.log
  ```
- Activate all monitors with specific syslog, run in background:
  ```
  nsqosmon -h11.0.0.114 -a -l/tmp/syslog.log -b
  ```
- Activate the following: all monitors, the 'clear process facts' (-e) option, default syslog monitoring, use TCP protocol with listener on port 6001 (-T, -s), trace process facts for process names starting with 'ov' (-mov), monitor kernel processes (-k), trace non-process facts (-t), handle sar reports with multiple line output (-M), and sar option for disk I/O stats is -b; and use 60 sec sar samples.
  ```
  nsqosmon -h11.0.0.148 -s6001 -T -r60 -a -k -t -e -mov -M –Rb
  ```

Sample Output for OS Monitor Startup

	**NOTE:**	This example is from a Solaris 7 system.

```
$ bin/nsqosmon -h11.0.0.148 -s6030 -r60 -T -p -c -f -t -e -mov
```



```
Sysname("SunOS")
NodeName("sunboss")
Release("5.7")

VERSION("2.1.0003")
AutoPilot OS monitor expert host:          11.0.0.148
AutoPilot OS monitor expert port:          6030
OS monitor interval:                       5 secs
System Activity Report (sar) interval:     60 secs
Syslog to monitor:                         Default
Syslog monitor:                            Enabled
File system monitor:                       Enabled
System config and performance monitor:     Enabled
Process monitor:                           Enabled
Clear facts for terminated process:        Enabled
All monitor options:                       Disabled
Run in:                                    Console
Communication protocol:                    TCP
Trace non-process fact messages:           Enabled
Trace process fact messages mask:          ov
```

	**NOTE:**	The syslog file is monitored by default. To disable monitoring, use the –d option.

4.6 Stopping OS Monitor in UNIX

If the `nsqosmon` process runs on the console, type **ctrl-c** command at the prompt.

If the `nsqosmon` process runs in the background, type: **kill –term <*pid*>**

where *pid* is the process ID of the nsqosmon parent process.

Chapter 5: Using Performance Monitor

5.1 Deploy AutoPilot Performance Monitor Expert

The following steps are required to enable AutoPilot to receive the metrics facts generated by the Performance Agent.

	NOTE:	Ensure that the AutoPilot Domain Server and Managed Node(s) are started.

1. Start AutoPilot Console.
2. Click **Deployment Tool** to display *Directory Viewer* (if not the default) to display the AutoPilot Network.

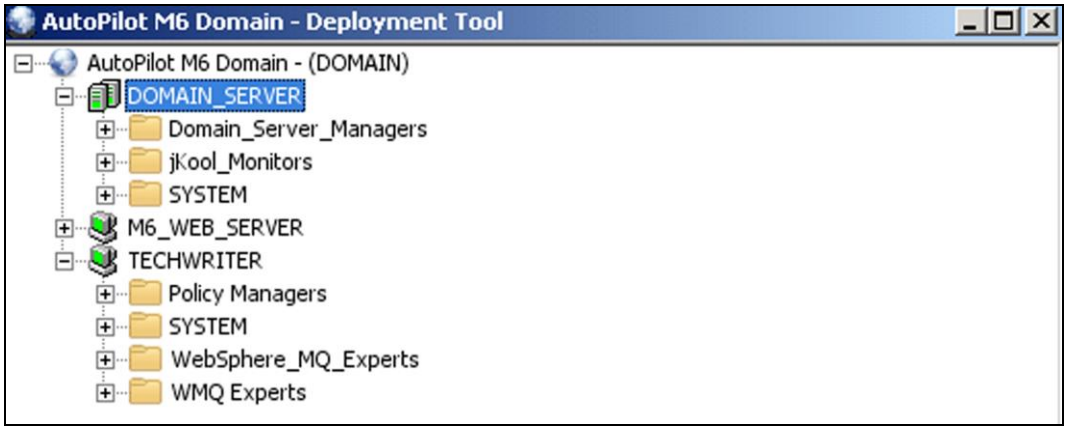

Figure 5-1. Deployment Tool and Networked Nodes

3. Right-click on the managed node and expand the node menu, click **Deploy Expert>>OS Monitors>>OS Performance Monitor** to initiate the Windows Monitor expert.

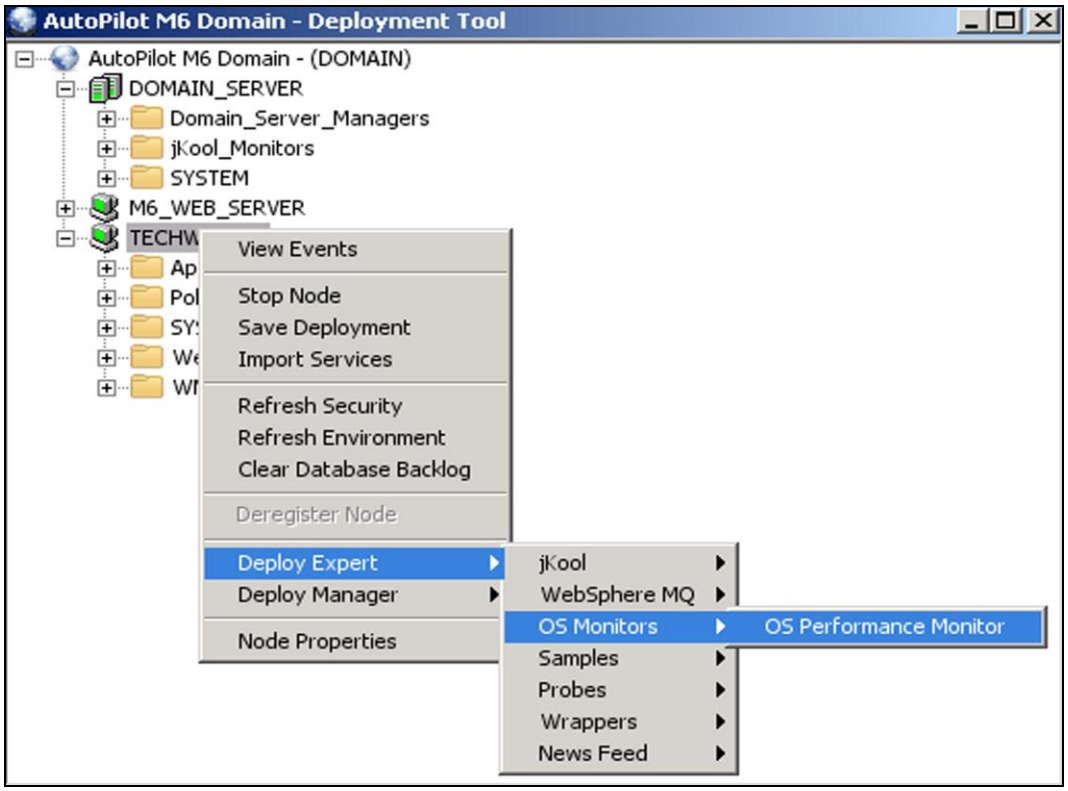

Figure 5-2. Windows Performance Monitor

4. Update fields and/or enable functions as needed. Under *General* properties, at a minimum, update the *Description* and *Name* to define your Windows Monitor expert.

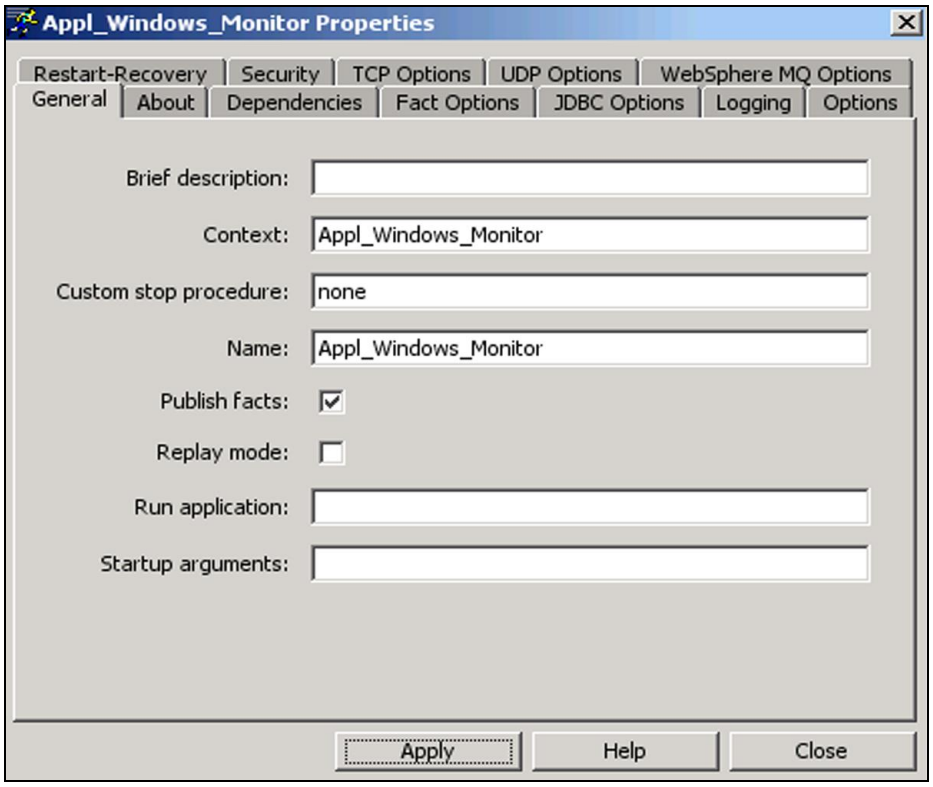

Figure 5-3. Modify OS Monitor Expert Configuration

Table 5-1. Windows Monitor: General	
Property	**Description**
Brief description	A short, user defined, description of the service.
Context	A user define category that will be registered with the domain server. Default is OS_Monitors.
Custom stop procedure	Application or script that shuts down started application gracefully. Default is none.
Name	Name that uniquely identifies the service in the domain. Default is OS_Monitor.
Publish facts	Disable/enable publishing of facts. Default is enabled.
Replay mode	Emulate fact source and replay all received facts.
Run application	Fully-qualified name of the application/script/process to run.
Startup arguments	Startup parameters that are passed to the application.

5. Click the *About* tab. The information in this table is not accessible to the user and is available for information purposes only.

Table 5-2. Common Properties: About	
Property	**Description**
Package Title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Package version as assigned by the vendor.

6. Click the *Dependencies* tab if required, identify, and format dependencies as defined in the table. These parameters are common to all experts.

Table 5-3. Common Properties: Dependencies

Property	Description
Platform Dependencies	Dependencies on OS platforms, comma separated list.
Service Dependencies	Dependencies on other services, comma separated list.

7. Click the *Fact Options* tab. Edit properties described in the table below. These parameters are common to all experts.

Table 5-4. Common Properties: Fact Options

Property	Description
Exclude Fact Filter	Comma separated list of fact paths to exclude during publishing.
Expire facts (ms)	User-defined time in which facts that have not been updated within a specific time automatically expire (in milliseconds).
Fact History Size	Automatically maintain specified number of samples for each published fact in memory.
Include Fact Filter	Comma separated list of fact paths to include during publishing.

8. Click the *JDBC Options* tab. Edit properties described in the table below.

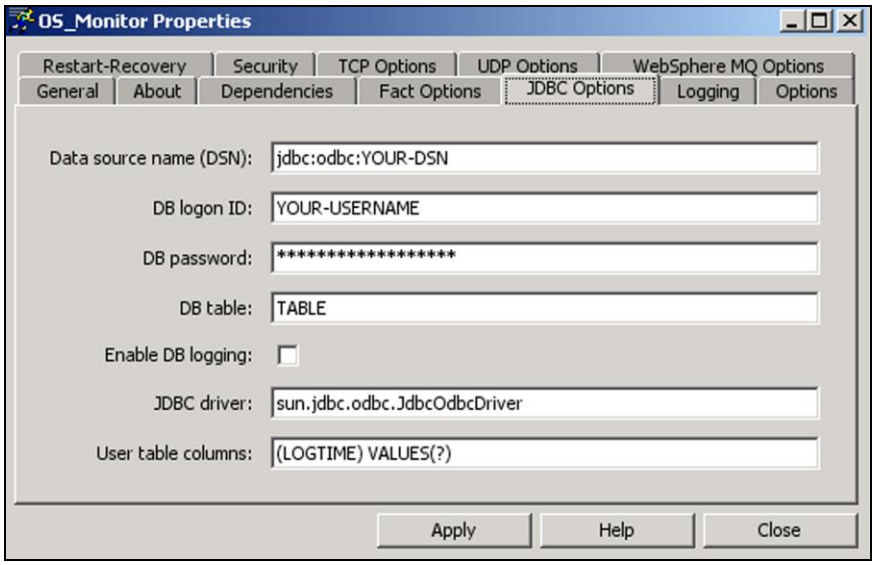

Figure 5-4. OS Monitor JDBC Options

Table 5-5. Windows Monitor: JDBC Options

Property	Description
Data source name (DSN)	Logical data source name that points to the physical database.
DB logon ID	Login ID to access database if required.
DB password	Password to access database if required.
DB table	Name of the physical database table (example: CPU_TABLE)
Enable DB logging	Enable/Disable database logging.
JDBC driver	Class name of the JDBC driver.
User table columns	(LogTime) VALUES (?)

9. Click the *Logging* tab. Identify and format logging requirements as defined in the table. These parameters are common to all experts.

Table 5-6. Common Properties: Logging	
Property	**Description**
Audit	Disable/enable service audit trace.
Log Name	Log name associated with the service.
Log Service Activity	Disable/enable service activity trace.
Log size (bytes)	Log size in bytes. Real log size is the maximum value of `server.log.size` and log size.
Record Facts	Enable/Disable recording for this service. Managed node records all facts produced by this service into an `.fct` file which can be replayed back using the apfact utility. Recording occurs only when managed node is started with –logfacts option or environment variable *property server.service.fact.logging=true* is defined in global.properties or node.properties.

10. Click the *Options* tab. If required, edit properties described in the table below.

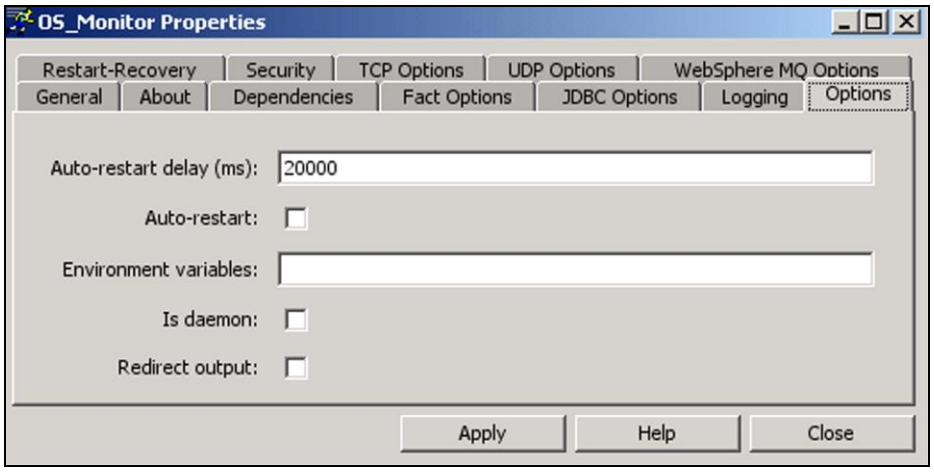

Figure 5-5. OS Monitor Options Tab

Table 5-7. Windows Monitor: Options	
Property	**Description**
Auto-restart delay (ms)	Delay before restarting stopped applications.
Auto-restart	Enable/Disable to specify whether application should be restarted when stopped.
Environment variables	Comma separated list of environment variables required for the application (Example: PATH=/opt.nastel,TEMP=/temp)
Is daemon	Enable/Disable to specify whether application is long running (daemon) or terminates after execution.
Redirect output	Enable/Disable to redirect program output to private event log.

11. Click the *Restart/Recovery* tab if required, disable/enable requirements as defined in the table. These parameters are common to all experts.

Table 5-8. Common Properties: Restart/Recovery

Property	Description
Automatic Start	Disable/enable automatic start.
Save in registry	Persistent services are saved in registry.
Synchronous Control	Disable/enable synchronous service initiation.

12. Click the *Security* tab, disable/enable requirements as defined in the table. These parameters are common to all experts.

Table 5-9. Common Properties: Security

Property	Description	
Inherit Permission from Owner	Always inherit permission from owner's permission masks.	
Owner:	User that owns the object.	
Permission	Permissions for users of the same group and others. Enable/disable as required.	
	Group	**Others**
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, disable.	Others may execute control actions such as start, stop, disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

13. Click the *TCP Options* tab. If required, edit properties described in the table below.

Table 5-10. Windows Monitor: TCP Options

Property	Description
Accept TCP facts	Disable/enable to accept facts published to the specified TCP port.
TCP port	Unique port on which this service will listen on for incoming facts.

14. Click the *UDP Options* tab. If required, edit properties described in the table below.

Table 5-11. Windows Monitor: UDP Options

Property	Description
Accept UDP facts	Disable/enable to accept facts published to the specified UDP port.
UDP port	Unique port on which this service will listen on for incoming facts.

15. Click the *WebSphere MQ Options* tab. Edit properties as described in the table below.

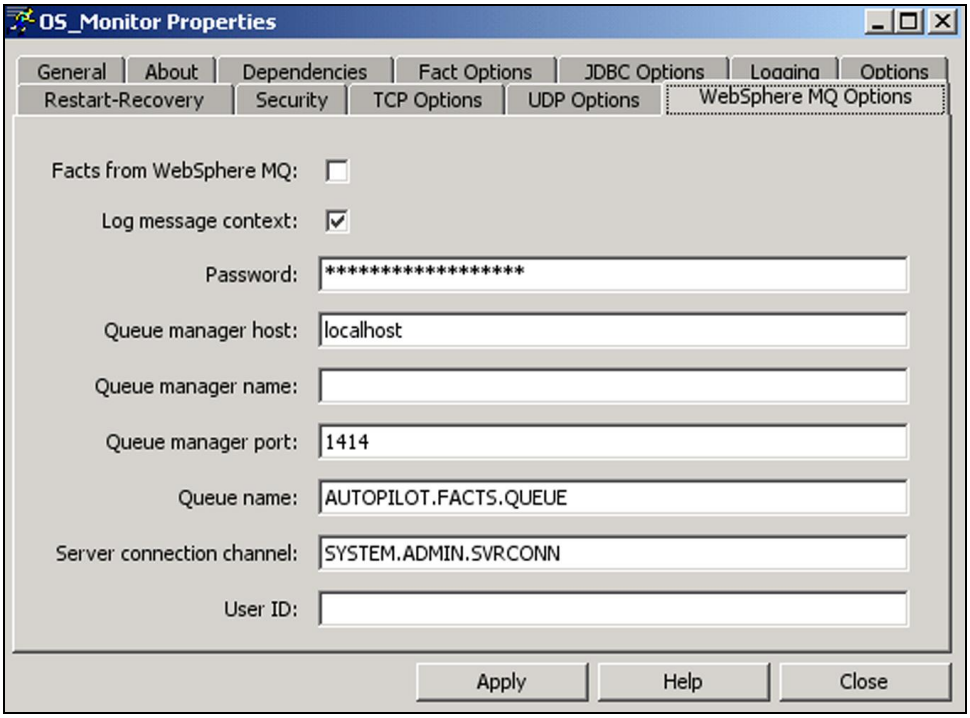

Figure 5-6. OS Monitor WebSphere MQ Options Tab

Table 5-12. Windows Monitor: WebSphere MQ Options

Property	Description
Facts from WebSphere MQ	Enable/Disable reading facts from WebSphere MQ.
Log message context	Enable/Disable to log WebSphere MQ descriptor such as type and persistence.
Password	Password to be used when connecting to the queue manager.
Queue manager host	Name of the host where the queue manager is defined.
Queue manager name	Name of the local queue manager.
Queue manager port	Channel listener TCP port for the queue manager.
Queue name	Name of the queue where the facts are being published.
Server connection channel	Name of the SVR CONN channel on the queue manager.
User ID	User ID to be used when connecting to the queue manager.

16. Click **Deploy**. The deployment message will confirm the name and location of the expert. Click **OK**.

Figure 5-7. Service Deployed

OR

17. Click **Deploy On**. Select the unique name and node where you want to deploy your expert. Click **Deploy.**

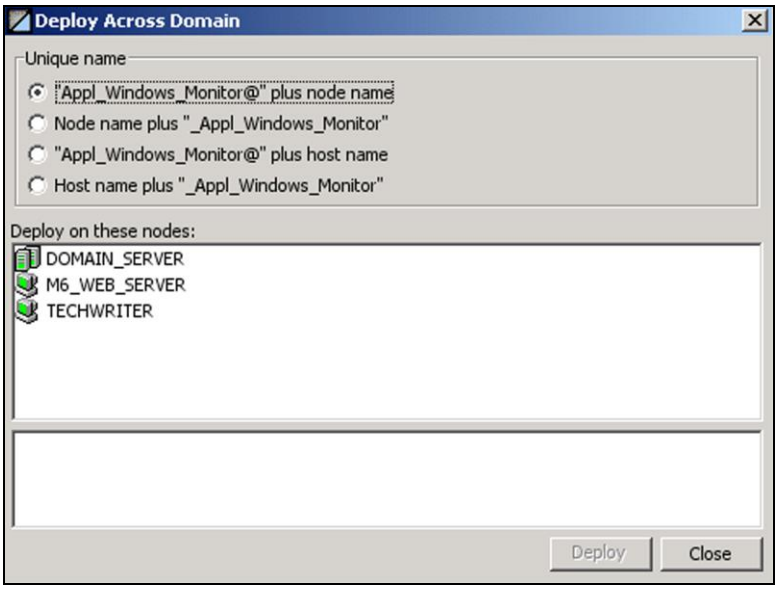

Figure 5-8. Deploying Expert Across Domain

5.2 Starting the Performance Monitor Expert on AP Console

Start the Performance Monitor Expert as follows:

1. Select the Windows Expert to be started. Stopped experts will display a red rectangular icon.
2. Right click on the expert and select the **Start** option from the pop-up menu. Select **Auto Start** if required. The icon will change to a green triangle when the expert is running.

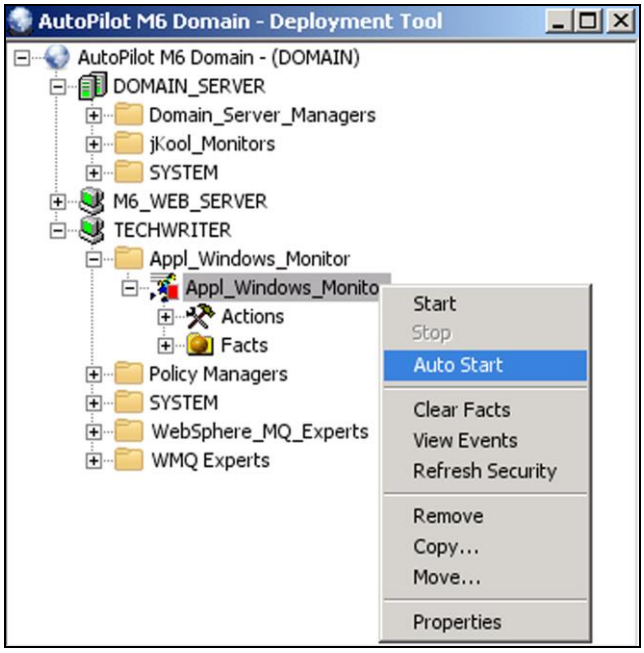

Figure 5-9. Starting the Windows Expert

3. Drill down into the expert to review the facts available. The facts displayed are based on what was selected during configuration. Facts for monitored services and system information are standard for all systems/machines.

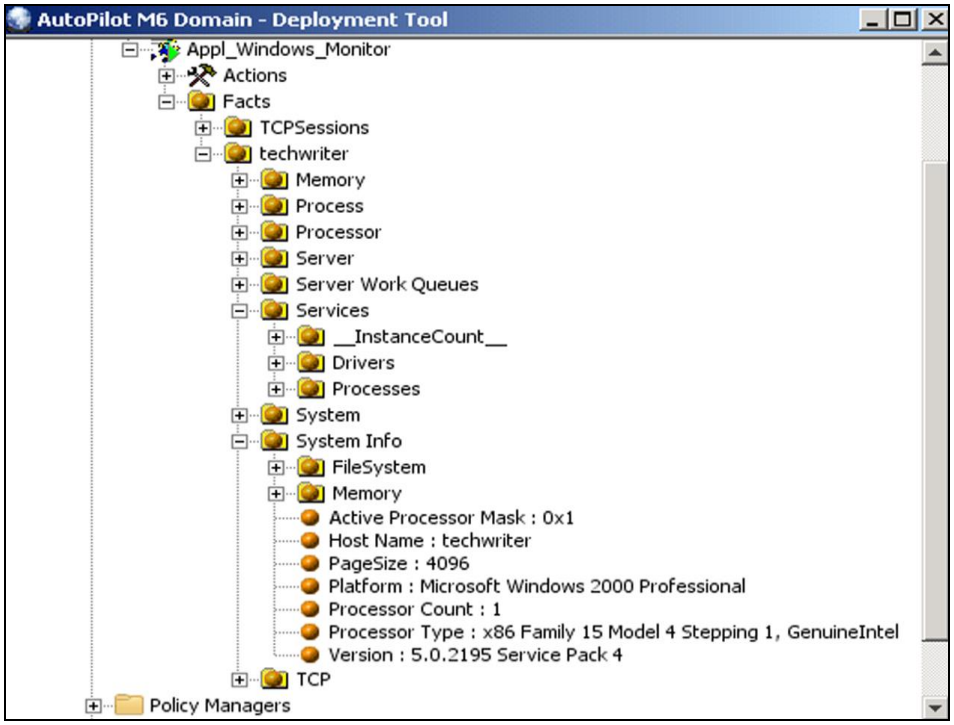

Figure 5-10. Example of Windows Monitor Metrics

Chapter 6: Configuration Metrics

6.1 Windows Performance Monitor Metrics

The Configuration Utility is used to define the specific Windows performance counters that the performance monitor is to collect. Each available counter (fact) produced by the Performance Monitor for Windows is described on the P*erformance Counter Browser* by doing the following:

1. Display the Configuration Utility by navigating to **Start>Programs>Nastel AutoPilot Performance Monitor>Agent Configuration** or by running `apperfmoncfg.exe`. (See Figure 3-1.).
2. Click **Add**.

   The *Performance Counter Browser* is displayed.
3. Highlight the counter and click **Explain**.

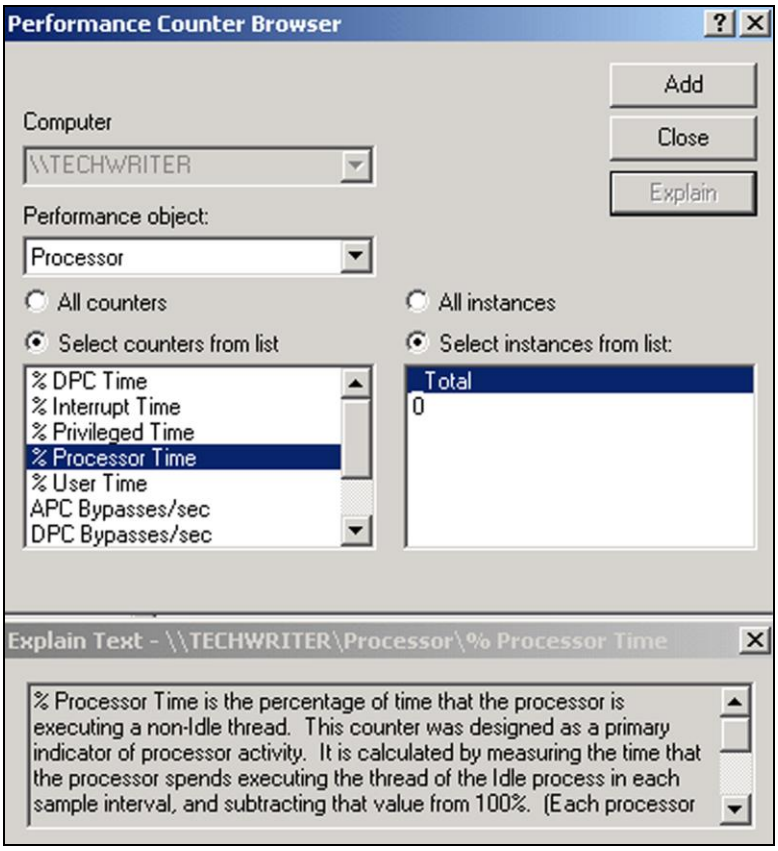

Figure 6-1. Describing Selected Counter

6.1.1 Process Monitor Metrics

In addition to the Windows performance counters, the process monitor can also be configured to monitor all system services and some general system information.

The services information is grouped by service type class (process, driver, and adapter). The following table explains the metrics published for services.

Table 6-1. Services Metrics

Metric	Description
Display Name	Name used by service control programs to identify the service.
Description	Description of Service.
Service Type	
Standalone Process	The service runs in its own process.
Shared Process	The service shares a process with other services.
Device Driver	The service is a device driver.
File System Driver	The service is a file system driver.
State	
Running	The service is running.
Stopped	The service is completely stopped.
Starting	The service is beginning.
Stopping	The service is stopping.
Continue Pending	The service is in a pending state.
Pause Pending	The service has paused the pending state.
Paused	The service is paused.
Start Type	
Boot	Started by the system loader (for Drivers only).
System	Started by the IoInitSystem function (for Drivers only).
Automatic	Started by the service control manager automatically during system startup.
Manual	Started by the service control manager when a process calls the StartService function.
Disabled	Can no longer be started.
Executable Path	Path to process to run when starting a service, along with any arguments.
Process ID	Process Identification of the service.
Runs in System Process	Whether it runs in system process (only published if service is running). Data value is either *Yes* or *No*.
List of service control commands as a ":" separated list	List of service control commands accepted. Possible values are: • Stop • Pause and Continue • Shutdown • Parameter Change • Network Binding • Hardware Profile Change • Power Event • Session Change

Table 6-1. Services Metrics	
Metric	**Description**
Error Control	
Ignore	The startup (boot) program logs the error but continues the startup operation.
Normal	The startup program logs the error and displays it in a message box pop-up. Continues the startup operation.
Severe	The startup program logs the error. If the last-known good configuration is being started, the startup operation continues. Otherwise, the system is restarted with the last-known-good configuration.
Critical	The startup program logs the error, if possible. If the last-known good configuration is being started, the startup operation fails. Otherwise, the system is restarted with the last-known good configuration.
Load Order Group	The load ordering group of which this service is a member. The startup program uses this list to load groups of services in a specified order with respect to the other groups in the list. This is not published if the service is not a member of a load order group.
Tag ID	Specifies a unique tag value for this service in the load order group. This is used for ordering service startup within a load order group. This is not published if the service is not a member of a load order group.
Dependencies	Names of services or load ordering groups that must start before this service.
Logon User	The account name that the service process will be logged on as when it runs. This is only published for process class services.
Driver Object	The driver object name that the input and output (I/O) system uses to load the device driver. This is only published for driver class services.
Instance Count	
Total	Total number of services.
Total for each service type	Total number of services for each type of service.

The following table explains the metrics published for general system information.

Table 6-2. System Metrics

Metric	Description
Host Name	DNS host name of the local computer.
Windows Platform	Family of Windows installed.
Windows Version	Specific version of Windows installed.
Processor Type	Identifies the type and class of processor installed.
Processor Count	Number of processors installed.
Active Processor Mask	A bit-mask identifying the active processors (if processor 0 is active, the bit 0 is set).
Page Size	Size of a virtual memory page.
Memory	
% Load	Current memory utilization as a percentage of total memory.
Physical Total Bytes	Total size of physical memory.
Physical Available Bytes	Amount of physical memory available.
Page File Maximum Bytes	Total possible size of the paging file.
Page File Available Bytes	Amount of space available in paging file.
File System **Drive** **Drive Type**	
Fixed	The disk cannot be removed from the drive.
Removable	The disk can be removed from the drive.
Network	The drive is a remote (network) drive.
CD-ROM	The drive is a CD-ROM drive.
RAM	The drive is a RAM disk.
Total Space Bytes	Total number bytes on the disk.
Free Space Bytes	Number of available bytes on the disk.

6.1.2 Error Status

For each group (and instance if the group supports instances), the following error status items are published, which indicate if there were any errors retrieving the metrics for the group or instance:

- **_FailedOperation** – the operation that failed.
- **_ErrorCode** – numeric error code returned by the operation.
- **_ErrorDesc** – textual description of the error code.

6.2 UNIX OS Monitor Metrics

	**NOTE:**	Metrics apply to all UNIX platforms unless specific platform(s) is displayed in parenthesis after the description.

6.2.1 UNIX Configuration Metrics

Below is a table defining the collected configuration metrics. The configuration metrics hierarchy in AutoPilot console is: `Facts\`*`hostname`*`\Configuration\`*`metric`*

Table 6-3. UNIX Configuration Metrics

Metric	Type	Description
Machine	String	Machine model or architecture. Examples: i686 (on Linux) 6005006F4 (on AIX) sun4u (on Solaris) 9000/800 (on HP-UX)
Memory_Buffers_MB	Integer	Total number of memory buffers. (AIX)
Num_Cpu	Integer	Number of CPUs physically on the system. (AIX, HP-UX, Solaris)
Num_Disk	Integer	Number of disks configured in the system. (HP-UX)
Os_Name	String	Name of the operating system.
Os_Version	String	Version of the operating system.
Os_Release	String	Current release of the operating system.
Physical _Memory_KB	Integer	Amount of physical memory (KB) in the system.
System_Id	String	Network node hostname of the system.
Uptime	String	Date and time system was last rebooted. Displayed as: day mmm dd hh:mm:ss yyyy Example: Tue Jan 6 09:30:44 2008
Virtual_Memory_MB	Integer	Amount of virtual memory in megabytes. (AIX)

6.2.2 UNIX File System Metrics

Below is a table defining the collected file system metrics. The file system metrics hierarchy in AutoPilot console is: `Facts\hostname\File_System\Fs_Dir_Name\metric`

where [*Fs_Dir_Name]* is the path name of the mount point of the file system (example: opt, export/home)

For Linux, AIX: Forward slash (/) character in mount point names are replaced by the underscore (_) character. Example: **/opt** appears as **_opt**. In addition, the root file system (/) appears as **root**.

For Solaris, HP-UX: Forward slash (/) characters are deleted.

Table 6-4. UNIX File System Metrics

Metric	Type	Description
Fs_Device_Name	String	Path name string of current device (examples: `/dev/hda/`, `/dev/vg001/lvl3`)
Fs_Inodes_Avail	Long integer	Number of non-super user I-nodes available for this file system. (≤ FS_Inodes_Max).
Fs_Inodes_Free	Long Integer	Available I-nodes for this file system.
Fs_Inodes_Max	Long integer	Maximum number of I-nodes for this file system.
Fs_Inodes_Used	Long integer	Number of l I-nodes used for this file system.
Fs_Inodes_Util_Percent	Double	Percent of I-nodes for this file system that were in use during the interval.
Fs_Block_Size	Integer	Average size of a block in this file system.
Fs_Blocks_Avail	Integer	Total blocks available in this file system. (≤ FS_Blocks_Max).
Fs_Blocks_Free	Unsigned long integer	Free blocks available in this file system.
Fs_Blocks_Max	Unsigned long integer	Total number of blocks in this file system.
Fs_Blocks_Used	Integer	Number of blocks used in this file system.
Fs_Blocks_Util_Percent	Integer	Percent of blocks utilized in this file system.
Fs_Type	String	The file system types on Unix systems typically are: hfs - hierarchical file system ufs - hierarchical UNIX file system nfs - network file system cdfs - CD-ROM file system vxfs - Veritas file system tmpfs - temporary file storage on many UNIX systems which uses virtual memory instead of persistent storage. ext3 - third extended file system; a journalled file system currently used by Linux. usbdevfs - USB device file system.

6.2.3 UNIX Performance Parameters Metrics

The table below defines the collected processor performance metrics. The performance metrics hierarchy in AutoPilot console is: `Facts\hostname\Performance\metrics`

Table 6-5. UNIX Performance Parameters

Metric	Type	Description
Context Switch Rate	Integer	Rate of context switches per second.
Cpu_Idle_Percent	Integer	Percent of time the CPU was idle during this interval with no outstanding disk I/O requests.
Cpu_Nice_Percent	Integer	Percent of time the CPU is in the 'nice' state, wherein processes are running with 'nice' values other than 0. Processes with negative values will have higher priority, with -20 being the highest. Processes with positive 'nice' values will have a lower priority, with +19 being the lowest. They are nice, unselfish tasks that allow other tasks to get a greater share of CPU time. (Linux)
Cpu_Non-Idle _Percent	Integer	Percent of time the CPU was not idle during this interval in seconds. (user + system time) (Linux, Solaris)
Cpu_Sys_Mode_Percent	Integer	Percent of time that the CPU was in system state during the interval.
Cpu_Total_Percent	Integer	Percent of time the CPU was in system state and/or user state. (AIX, HP-UX)
Cpu_User_Mode_Percent	Integer	Percent of time that the CPU was in user mode during the interval.
CPU_Wait_IO_Percent	Integer	Percent of time the CPU is idle and waiting for I/O completion. (HP-UX, Solaris)
Currently_Logged_Users	Integer	Number of users currently logged in at this time.
Disk_Block_Read_Rate	Integer	Disk read rate in blocks per second. (Linux)
Disk_Block_Write_Rate	Integer	Disk write rate in blocks per second. (Linux)
Disk_Read_Transfers_Rate	Integer	Disk read transfer requests per second. (Linux)
Disk_Tranfers_Rate	Integer	Disk total I/O transfers per second. (Linux)
Disk_Write_Transfers_Rate	Integer	Disk write transfer requests per second. (Linux)
Disk_Physical_Byte	Long Integer	Number of bytes transferred from and to the disks during the interval. (Solaris)
Disk_Physical_Io	Long Integer	Number of data transfers (read and writes) from and to disks. (Solaris)
File_Handles_Alloc	Long Integer	Number of allocated file handles (limit is File_Handles_Max). (Linux)
File_Handles_Free	Long Integer	Number of free handles. (Linux)
File_Handles_Max	Long Integer	Maximum number of file handles that can be allocated. (Linux)
File_Handles_Used	Long Integer	Derived Fact: File_Handles_Alloc - File_Handles_Free. (Linux)
File_Table_Avail	Long Integer	Maximum size of the file table. (AIX, HP-UX, Solaris)
File_Table_Used	Long Integer	Current size of the file table. (AIX, HP-UX, Solaris)
Free_Memory_MB	Long	Free memory available during the interval. (AIX, HP-UX)
Inodes_Alloc	Long	Number of i-nodes the system has allocated. **NOTE:** This can be slightly more than i-Inodes_Max because Linux allocates i-nodes a page at a time. (Linux)
Inodes_Free	Long	Number of free i-nodes. (Linux)

Table 6-5. UNIX Performance Parameters		
Metric	**Type**	**Description**
Inodes_Max	Long	Maximum number of M-memory i-nodes. **NOTE:** This metric may not be available on all Linux systems. (Linux)
Inodes_Used	Long	Number of i-nodes used. (Linux)
Inode_Table_Avail	Long	Maximum size of the i-node table. (AIX, HP-UX, , Solaris)
Inode_Table_Used	Long	Current size of the i-node table. (AIX, HP-UX, Solaris)
IPC_Message_Primitives_Rate	Float	Call rate of interprocess communication (IPC) using message primitives msgrcv( ). (AIX, HP-UX)
IPC_Semaphore_Primitives_Rate	Float	Call rate of interprocess communication using semaphore primitives semop( ) (AIX, HP-UX)
Load_Averages	String	Load averages in the last 1, 5, and 15 minutes (xx.xx, xx.xx, xx.xx)
Messages_Primitives	Float	Number of System V IPC message receive and sent calls per second during the interval. (Solaris)
Proc_Table_Avail	Long	Maximum size of the process table. (AIX, HP-UX, Solaris)
Proc_Table_Used	Long	Current size of the process table. (AIX, HP-UX, Solaris)
Processes	Integer	Number of processes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Each process represents the running of a program.
Ram_Memory_Free_KB	String	Total amount of free memory expressed in Kilobytes. (Linux, Solaris)
Ram_Memory_Used_KB	String	Total amount of used memory expressed in Kilobytes. (Linux, Solaris) NOTE: Ram_Memory_Free_KB + Ram_Memory_Used_KB = Physical_Memory_KB. See Table 6-3.
Real_Memory_Active	String	Amount of physical memory in use during this interval. (HP-UX)
Real_Memory_Total	String	Maximum amount of physical memory available during this interval. (HP-UX)
Semaphores_Primitive_Rate	Float	Number of semaphore operations per second during this interval. Semaphore primitives (create, use, destroy) (Solaris)
Swap_Memory_Free_KB	Integer	Available swap memory. (AIX, Linux, Solaris)
Swap_Memory_Used_KB	Integer	Used swap memory. (AIX, Linux, Solaris)
Swapin_Page_Rate	Float	Number of page swapins per second. (Linux)
Swapout_Page_Rate	Float	Number of page swapouts per second. (Linux)
Swapin_Process_Rate	Float	Number of process swapins per second. (HP-UX, Solaris)
Swapout_Process_Rate	Float	Number of process swapouts per second. (HP-UX, Solaris)
System_Calls_Rate	Integer	Rate of system calls of all types per second. (AIX, HP-UX, Solaris)
System_Execs_Rate	Float	Rate of exec system calls per second during this interval. (AIX, HP-UX, Solaris)
System_Fork_Rate	Float	Forks and vfork system calls per second during the interval.
System_Read_Rate	Integer	Number of read and readv system calls per second. (AIX, HP-UX, Solaris)
System_Select_Rate	Integer	Number of select ( ) calls per second. Related to message and semaphore activities. (HP-UX)

Table 6-5. UNIX Performance Parameters		
Metric	**Type**	**Description**
System_Write_Rate	Integer	Rate of write and writev system calls per second. (AIX, HP-UX, Solaris)
Virtual_Memory_Active	String	Amount of active virtual memory that is in use by recently run processes. (HP-UX)
Virtual_Memory_Total	String	Maximum amount of virtual memory. (HP-UX)

6.2.4 UNIX Process Metrics

Below is a table defining the collected file process metrics. The file process metrics hierarchy in AutoPilot console is:

`Facts\`*`hostname`*`\Process\`*`processName_pid`*`\`*`metric`*

`Facts\`*`hostname`*`\Process\`*`processName`*`\Instance`

Table 6-6. UNIX Process Parameters

Metric	Type	Description
Args	String	Command line arguments for the process (max 80 characters).
Cpu_Percent	Double	Percent of CPU utilization during the interval.
Instance	String	Number of instances of this process that are running .
Memory_Percent	Integer	Percent of memory used by this process.
Nice	Integer	Nice value (priority computation). See UNIX man page ps (1), nice (1).
Page_Faults	Integer	Number of page faults during the interval. (AIX, Linux, Solaris)
Pid	Integer	Process identification number.
Priority	Integer	Priority of the process.
Size_Image_KB	Integer	Image size of the process during the interval. This is calculated as Size = data size + text size + stack size.
Size_Resident_KB	Integer	Resident set size (KB) of the process.
State	String	Current status of the process. Some of the possible status values of the process are: Sleeping, run, stop, start, cpu, swap, zombie, other, idle. (AIX: sleeping, static, runable, idle, zombie, stopped, running, swapped.) (HP-UX: sleeping, running, stopped, zombie, other, idle.)
Thrds	Integer	Number of threads running in the process. (AIX, HP-UX, Solaris)
Time_Secs	Integer	Time spent on execution. Calculated as Time = user time + system time.
Uid	Integer	User ID of the process.
User	String	Name of owner of the process.

6.2.5 UNIX Syslog Metrics

Below is a table defining collected syslog metrics. The syslog metrics hierarchy in AutoPilot console is:

`Facts\`*`hostname`*`\Syslog\`*`Category`*`\`*`metrics`*

Table 6-7. UNIX Syslog Metrics

Metric	Type	Description
Syslog	String	Top level fact folder for syslog; literal.
Category	String	Syslog facility generating the message. Example: user. See table 6-8.
Date	String	Time stamp. Displayed as: mmm dd hh:mm:ss Example: Jan 6 16:10:44
Message	String	Message string. Example: bootpd 26172:IP address not found: 11.0.0.23
Severity	String	Severity level: Notice, Information, Alert, Critical Error, Warning, Emergency, Debug, None, etc. See man (syslogd), man (syslog). See table 6-9.

6.2.6 UNIX Syslog Category

Each syslog message is generated by an application category or facility, which are listed in the table below.

Table 6-8. UNIX Syslog Message Facilities

Numeric Code	Category (Facility)
0	kernel messages
1	user-level message
2	mail system
3	system daemons
4	security/authorization messages (see note 1)
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon (see note 2)
10	security/authorization messages (see note 1)
11	FTP daemon
12	NTP subsystem
13	log audit (see note 1)
14	log alert (see note 1)
15	clock daemon (see note 2)
16	local use 0
17	local use 1
18	local use 2
19	local use 3
20	local use 4
21	local use 5
22	local use 6
23	local use 7

NOTES:

1. Various operating systems have been found to utilize facility codes 4, 10, 13, and 14 for security/authorization, audit, and alert messages, which seem to be similar.
2. Various operating systems have been found to utilize facility codes 9 and 15 for clock (cron/at) messages.

6.2.7 UNIX Syslog Severity Levels

Each syslog message has a decimal severity level indicator. These are described in the table below:

Table 6-9. UNIX Syslog Message Severities

Numeric Code	Severity	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical condition
3	Error	Error condition
4	Warning	Warning condition
5	Notice	Normal but significant condition
6	Informational	Informational messages
7	Debug	Debug-level messages

6.2.8 UNIX Performance Overhead

The following table shows the CPU usage of the OS Monitor collecting metrics at different intervals using the **top** utility on HP-UX 11.00 A 9000/800 machine with 128MB RAM. The numbers indicate the monitor does not add an overhead that would affect other processes.

Table 6-10. Performance Tuning

Polling Frequency	CPU Idle Time	CPU Usage of Monitor
5 interval seconds	97.75	0.31
10 interval seconds	97.29	0.30
15 interval seconds	97.34	0.30

NOTE: The metrics vary depending on system configuration and system usage.

Appendix A: References

A.1 Nastel Documentation

Table A-1. Nastel Documentation	
Document Number (or higher)	**Title**
M6/INS 600.007	*AutoPilot M6 Installation Guide*
M6/USR 600.009	*AutoPilot M6 User's Guide*
M6WMQ 600.002	*AutoPilot M6 Plug-in for WebSphere MQ*
M6WMQ/ADM 620.002	*AutoPilot M6 for WebSphere MQ Administrator's Guide*
M6WMQ/INS 620.002	*AutoPilot M6 for WebSphere MQ Installation Guide*
M6WMQ/SM 600.001	*AutoPilot M6 for WebSphere MQ Security Manager User's Guide*

A.2 Java™

http://java.sun.com/products/JavaManagement/reference/docs/index.html
http://www.hp.com/products1/unix/java/infolibrary/index.html
http://developer.java.sun.com/developer/technicalArticles/Servlets/corba/

A.3 MS Windows

http://www.microsoft.com/windows/default.mspx

A.4 UNIX

http://www.unix.org/

A.5 Solaris

http://www.sun.com/software/solaris/

A.6 HP-UX

http://welcome.hp.com/country/us/en/welcome.html

A.7 Linux

http://www.linux.org/

This Page Intentionally Left Blank

Appendix B: Conventions

B.1 Typographical Conventions

Table B-1. Typographical Conventions

Convention	Description
Blue/Underlined	Used to identify links to referenced material or websites. Example: support@nastel.com
Bold Print	Used to identify topical headings, glossary entries, and toggles or buttons used in procedural steps. Example: Click **EXIT**.
Italic Print	Used to place emphasis on a title, menu, screen name, or other category.
`Monospaced Bold`	Used to identify keystrokes/data entries, file names, directory names, etc.
`Monospaced Italic`	Used to identify variables in an address location. Example: [`C:\`*`AutoPilot_Home`*`]\documents`, where the portion of the address in the brackets [ ] is variable.
`Monospaced Text`	Used to identify addresses, commands, scripts, etc.
Normal Text	Typically used for general text throughout the document.
Table Text	Table text is generally a smaller size to conserve space. 10, 9, and 8 point type is used in tables throughout the AutoPilot product family of documents.

B.2 Naming Conventions

In the redesign of AutoPilot M6, we have defined many elements within the AutoPilot M6 product line.

Table B-2. AutoPilot M6 Related Naming Conventions

Old Name	New Name	Abbreviated As	Link
Nastel AutoPilot	Nastel AutoPilot M6	M6	NA
AutoPilot Web	M6 Web Server	M6 Web Server	http://host:8080
AutoPilot Web Portal	M6 Web Console	M6 Web Console	http://host:8080/m6console
AutoPilot Reports	AutoPilot M6 Reports	M6 Reports	http://host:8080/m6reports
Nastel AutoPilot Business Dashboard	Nastel AutoPilot M6 Business Dashboard	M6 Dashboard	See product documentation.

Glossary

AutoPilot M6: Nastel Technologies' Enterprise Application Management Platform. AutoPilot M6 monitors and automates the management of *e*Business integration components such as middleware application, application servers and user applications.

AP-WMQ: Nastel Technologies' WebSphere MQ management solution. Re-designated as AutoPilot M6 for WebSphere MQ with release 6.0. Abbreviated as AP/WMQ and AP-WMQ.

BSV: *see* Business View.

Business View (BSV): A collection of rules that define a desired state of an *e*Business environment. Business Views can be tailored to presents information in the form most suited to a given user, as defined by the user.

Client: Any programming component that uses the AutoPilot infrastructure; for example, the AutoPilot M6 Console.

Common Object Request Broker Architecture (CORBA): An object that can be invoked from a Web browser using CGI scripts or applets.

Console: The console acts as the graphical interface for AutoPilot M6.

Contacts: A subordinate to a given Manager or Expert.

CORBA: *See* Common Object Request Broker Architecture.

Data Source Name (DSN): The logical name that is used by Open Database Connectivity (ODBC) to refer to the drive and other information that is required to access data. The name is use by Internet Information Services (IIS) for a connection to an ODBC data source, (For Example: Microsoft SQL Server database). The ODBC tool in the Control Panel is used to set the DSN. When the ODBC DSN entries are used to store the connection string values externally, you simplify the information that is needed in the connection string. This makes changes to the data source completely transparent to the code itself.

Dependent WebSphere MQ Node: WebSphere MQ nodes that are not directly managed by M6-WMQ. Because dependent nodes do not run an MQ WMQ Agent, they must be managed by proxy.

Deploy: To put to use, to position for use or action.

Domain Server: A specialized managed node that maintains the directory of managed nodes, experts etc. The domain server is also capable of hosting experts, managers etc.

DSN: *See* Data Source Name.

EVT: Event Log file extension (e.g.: *sample.evt*)

Event: An *Event* is something that happens to an object. Events are logged by AutoPilot M6 and are available for use by AutoPilot M6 Policies or the user.

Expert: Services that monitor specific applications such as an applications server, web-server or specific components within the applications (Example: channels in WebSphere MQ.) Experts generate facts.

Fact: Facts are single pieces of data that have a unique name and value. One or more facts are used to determine the health of the object, application or server.

Graphical User Interface (GUI): A type of environment that represents programs, files, and options by means of icons, menus, and dialog boxes on the screen. The user can select and activate these options by pointing and clicking with a mouse or, often, with the keyboard. Because the graphical user interface provides standard software routines to handle these elements and report the user's actions (such as a mouse click on a particular icon or at a particular location in text, or a key press); applications call these routines with specific parameters rather than attempting to reproduce them from scratch.

GUI: *See* Graphical User Interface.

Independent WebSphere MQ Node: A WebSphere MQ node that runs a WMQ Agent and which is managed directly by an MQ Workgroup server. Independent nodes can be used as proxy nodes for managing dependent nodes.

IIS: *See* Internet Information Services.

Internet Information Services (IIS): Microsoft's brand of Web server software, utilizing HTTP to deliver World Wide Web documents. It incorporates various functions for security, allows CGI programs, and also provides for Gopher and FTP services.

Java: A platform-independent, object-oriented programming language developed and made available by Sun Microsystems.

Java Developer's Kit (JDK): A set of software tools developed by Sun Microsystems, Inc., for writing Java applets or applications. The kit, which is distributed free, includes a Java compiler, interpreter, debugger, viewer for applets, and documentation.

JDBC: *See* Java Database Connectivity.

Java Database Connectivity (JDBC): The JDBC API provides universal data access from the Java programming language. Using the JDBC 2.0 API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built. The JDBC *Test Tool* that was developed by Merant and Sun Microsystems may be used to test drivers, to demonstrate executing queries and getting results, and to teach programmers about the JDBC API.

Java Server Pages (JSP): JSP technology enables rapid development of web-based applications that are platform independent. Java Server Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. Java Server Pages technology is an extension of the Java Servlet technology.

Java Virtual Machine (JVM): The "virtual" operating system that JAVA-written programs run. The JVM is a hardware- and operating system-independent abstract computing machine and execution environment. Java programs execute in the JVM where they are protected from malicious programs and have a small compiled footprint.

JDK: *See* Java Developer's Kit.

JRE: JAVA Run-time Environment. The minimum core JAVA required to run JAVA Programs.

JSP: *See* Java Server Pages.

JVM: *See* Java Virtual Machine.

M6 for WMQ: Nastel Technologies' WebSphere MQ management solution. Re-designated as M6 for WMQ with release 6.0, prior releases retain the AP-WMQ or MQControl trademark.

M6 Web: A browser-based interface that provides monitoring and operational control over managed resources and applications.

Management Information Base (MIB): A specification that describes the properties and behavior of a network device. Network managers use MIBs to interact with SNMP-compatible devices. Each MIB is part of a directory structure that specifies where objects are found on the network.

Manager: Managers are the home or container for policies. All business views must reside on managers, and manager must be deployed prior to deploying a business view or policy.

Message Oriented Middleware (MOM): A client/server infrastructure that increases the interoperability, portability, and flexibility of an application by allowing the application to be distributed over multiple heterogeneous platforms.

Message Queue Interface (MQI): A method of program-to-program communication suitable for connecting independent and potentially non-concurrent distributed applications. A part of IBM's Networking Blueprint.

MIB: *See* Management Information Base.

MOM: *See* Message-Oriented Middleware.

MQControl: Nastel Technologies' MQSeries management product. Re-designated as AP-WMQ with release 4.0 and M6 for WMQ with release 6.0. Prior releases retain the MQControl trademark.

MQI: *See* Message Queue Interface.

MQSC: *See* WebSphere MQ Commands

MQSeries: IBM's message queuing product. Renamed by IBM as WebSphere MQ.

Managed Node: A container that can host any number of AutoPilot M6 services such as experts, managers, policies, etc. Unlike managed nodes, it is a physical process.

Naming Service: A common server records "names" of objects and associates them with references, locations and properties.

Object Request Broker (ORB): A piece of middleware software that allows programmers to make program calls from one computer to another via a network.

ORB: *See* Object Request Broker.

Orbix: CORBA product distributed by IONA Technologies.

Package Manager: The command line utility that allows users to list, install, uninstall, verify and update AutoPilot M6 installation on any Managed Node.

PCF: *See* Programmable Command Format.

PKGMAN: *See* Package Manager.

Policy/Business Views: Business views are a collection of one or more sensors. Business views are used to visually present the health and status of the different systems as well as automatically issue remedial actions.

Programmable Command Format (PCF): A set of programmable commands that M6-WMQ uses to manage WebSphere MQ. PCF includes data definitions for items such as integers, strings, and lists. The commands can be submitted directly to a queue manager. PCF is comparable to MQSC, except for the fact that MQSC cannot be programmed.

Proxy Management: The indirect management of MQ objects by an intermediate entity. For example, a proxy queue manager might be used to handle another queue manager.

QSG: *See* Queue Sharing Group.

Queue Sharing Group (QSG): In z/OS, a group of queue managers in the same sysplex that can access a single set of object definitions stored in the shared repository, and a single set of shared queues stored in the coupling facility. The shared queue is a type of local queue. The messages on the queue are stored in the coupling facility and can be accessed by one or more queue managers in a queue-sharing group. The definition of the queue is stored in the shared repository.

Sensor: A rule that is used to determine the health of an object or application based on one or more facts. Actions can then be issued, based on health. Sensors are definable in AutoPilot business views by use of the sensor wizard.

Simple Mail Transfer Protocol (SMTP): A TCP/IP protocol for sending messages from one computer to another on a network. This protocol is used on the Internet to route e-mail. *See also* communications protocol, TCP/IP. *Compare* CCITT X series, Post Office Protocol.

Simple Network Management Protocol (SNMP): A de facto standard for managing hardware and software devices on a network. Each device is associated with a Management Information Base (MIB) that describes its properties and behavior.

SMTP: *See* Simple Mail Transfer Protocol.

SNMP: *See* Simple Network Management Protocol.

SNMP Master Agent: An implementation of the SNMP protocol. It includes a definition of the standard MIB. The master agent routes SNMP requests from subagent to subagent.

SNMP Subagent: The implementation of an MIB for a particular device. The MIB describes the device's desired behavior; the SNMP subagent carries it out.

TCP/IP: *See* Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP): A protocol developed by the Department of Defense for communications between computers. It is built into the UNIX system and has become the de facto standard for data transmission over networks, including the Internet.

Virtual Machine: Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *See* Java Virtual Machine.

WAP: *See* Wireless Application Protocol.

WebSphere MQ: IBM's message queuing product. Formally known as MQSeries.

WebSphere MQ Commands: A command-line language used to configure WebSphere MQ.

Websphere_MQ_Manager: A specialized manager capable of hosting one or more MQSeries specific policies, apart from the regular policies.

Workgroup Server (WS): Monitors WebSphere MQ nodes. A workgroup server consists of two agents, M6 managed node and workgroup.

Wireless Application Protocol (WAP): An open global specification that is used by most mobile telephone manufacturers. WAP determines how wireless devices utilize Internet content and other services. WAP enables devices to link diverse systems contents and controls.

WS: *see* Workgroup Server.

z/OS: *see* Z Series Operating System.

Z Series Operating System (z/OS): IBM architecture for mainframe computers and peripherals. The zSeries family of servers uses the z/Architecture. It is the successor to the S/390 and 9672 family of servers.