

Comparison Article OpalisRobot vs. Windows 2000

- Performance Monitoring
- Scheduling
- Service Recovery

# Table of Contents

---

<b><u>Microsoft Windows 2000 and OpalisRobot</u></b> .....	<b>3</b>
<u>Objective</u> .....	3
<u>Why Compare Windows 2000 Features with OpalisRobot?</u> .....	3
<b><u>Performance Monitoring</u></b> .....	<b>4</b>
<u>Where OpalisRobot Excels in Performance Monitoring</u> .....	6
<b><u>Scheduling</u></b> .....	<b>8</b>
<u>Where OpalisRobot Excels in Task Scheduling</u> .....	9
<b><u>Service Recovery</u></b> .....	<b>11</b>
<u>Where OpalisRobot Excels in Service Recovery</u> .....	12

### **A Comparative Look at the Feature Enhancements**

#### **Objective**

This article serves to provide a comparative look at the feature enhancements in Windows 2000 in light of the functionality provided by OpalisRobot. What you will find is that OpalisRobot provides a robust solution to fulfill the directives of network administrators and other operators interested in monitoring, managing and automating their systems.

#### **Why Compare Windows 2000 Features with OpalisRobot?**

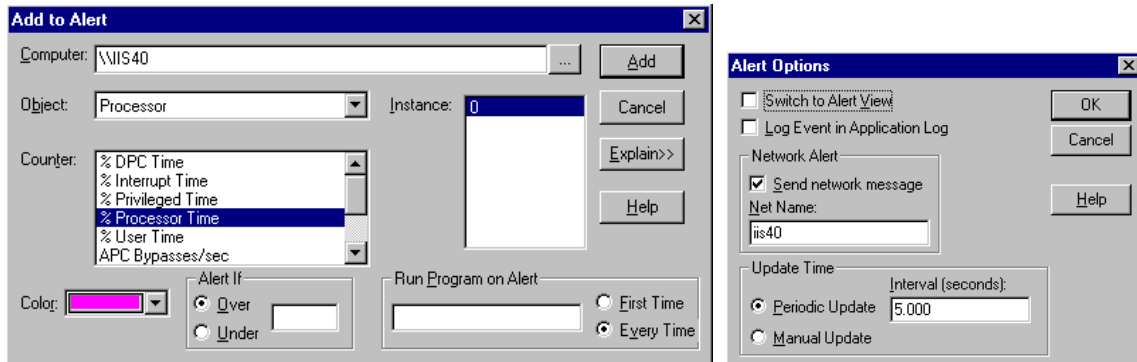
The improvements found in the Windows 2000 features covered in this article collectively evidence the growing desire for automation solutions. This is apparent in Windows 2000 feature enhancements that automatically detect performance problems and automate the launch of data logs, tasks based on a schedule or on the occurrence of particular events, or automatically recover failed services.

For individuals running on Windows 2000 and who have yet to experience the benefits of being an OpalisRobot user, the enhanced features in Windows 2000 may lead them to wonder if there exists a need to investigate an automation solution like OpalisRobot. Although the new features in Windows 2000 allows for the automation of some operations, many users will eventually find themselves looking for additional functionality and increased flexibility to satisfy a growing appetite for a more complete automation solution.

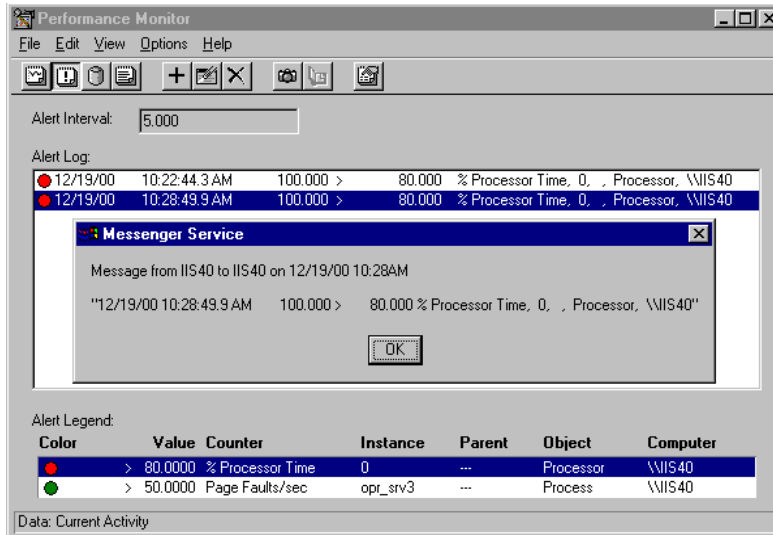
## Performance Monitoring

The Performance Monitor found in NT 4.0 was quite commonplace in the network administrator's arsenal of troubleshooting tools. To use the NT Performance Monitor was simple enough: you start by deciding which specific performance counters to monitor, and then continue to configure charts, logs, reports and alerts. It was a quick and easy way of troubleshooting various aspects of your system.

The following screenshots provide a quick look at performance alerts in NT 4.0. Here you see the dialog used to select a performance counter, set an alert threshold, and specify the course of action to take:

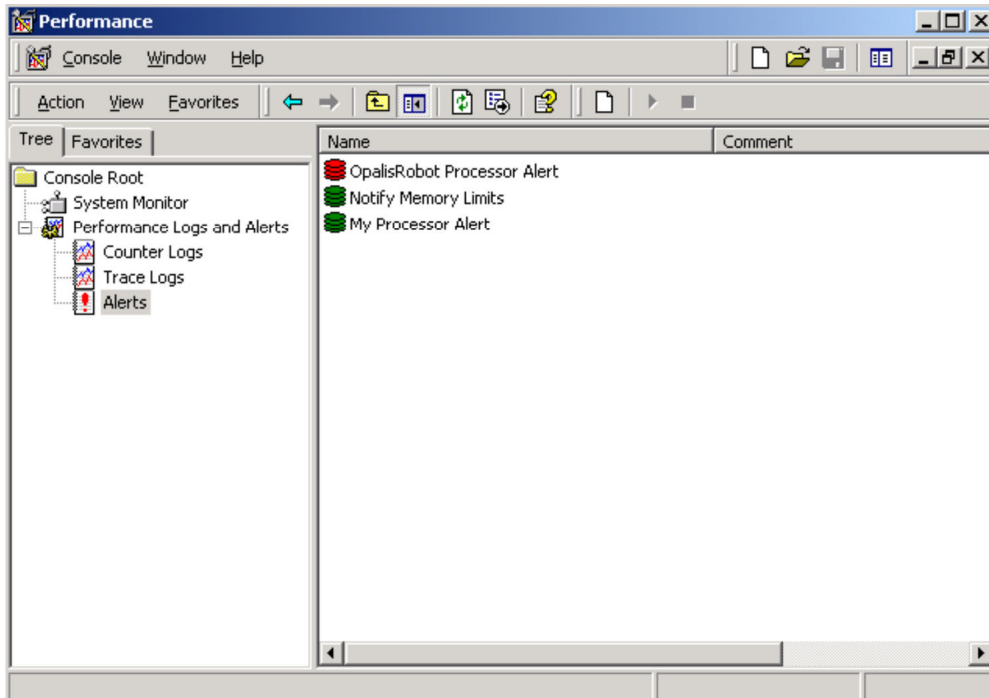


Here you see the list of configured performance alerts and the result of an alert being triggered:



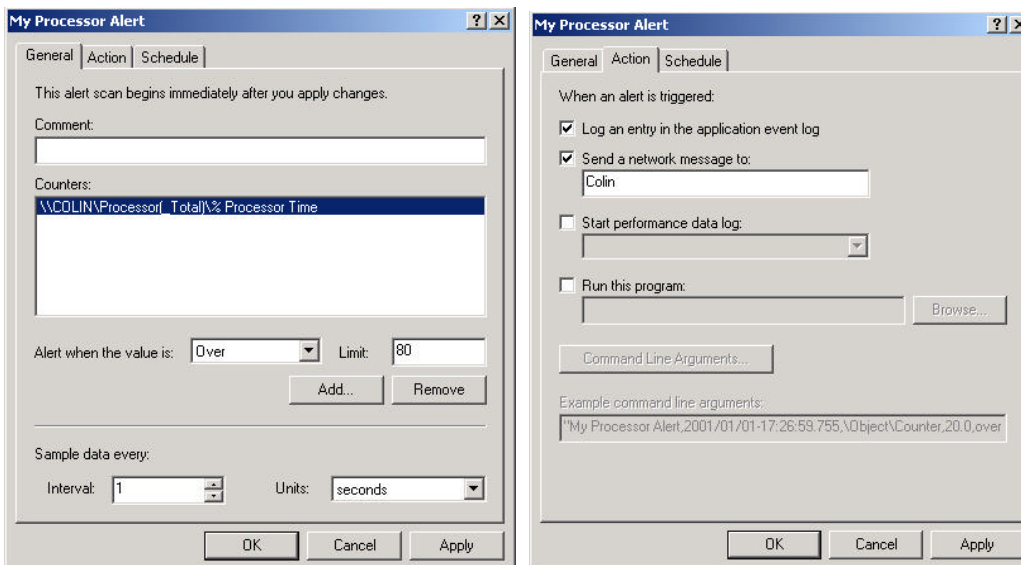
The Performance Monitor improvements in Windows 2000 makes it easier to manage logs and alerts it also provides some additional functionality. Aside from the Performance Monitor being an MMC snap-in, the new interface for the Performance Monitor (actually called System Monitor in Windows 2000) allows you to open multiple instances of performance charts and provides a consistent interface for logs and alerts.

The following screenshot shows the difference in the Alerts interface:

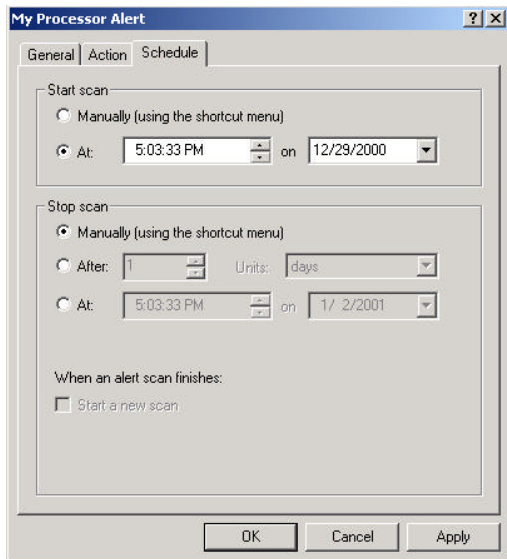


Alerts can now be given a specific name by the user and are no longer identified simply by the colour assigned to it and a description as its default name. You can also have multiple monitor instances of the same performance counter, whereas NT would modify an existing performance alert if one already exists with the same counter. Every individual alert can also be started or stopped.

A nice feature in the Windows 2000 Performance Monitor is the ability to automate a specific performance data log. This was not available in Windows NT and allows the user to automate data collection upon an alert trigger.



You can also schedule your alerts to run only during specific activity periods:



The improvements made to the Performance Monitor in Windows 2000 serve to reduce the amount of interaction required by the user assigned to monitor the system and to automatically simulate how that user would monitor and react to particular problem situations. Also note the new addition of a service on the Windows 2000 system named: Performance Logs and Alerts.

## Where OpalisRobot Excels in Performance Monitoring

OpalisRobot excels in general because it can act as a central point for the configuration and management of all automation implemented on a system. The object-oriented nature of OpalisRobot also allows automated tasks to interact with each other and will allow each object to employ the power of any other object. For instance, all events and tasks in OpalisRobot can use any of the notification objects available to send a variety of alerts and is not limited to the notification options provided for by each individual utility. OpalisRobot users will find that this object-oriented design provides a very high level of functionality, flexibility and scalability.

Here we look specifically at performance monitoring and what operators will realize when using only the Windows 2000 Performance Monitor. To provide automated responses to performance and system problems, a threshold is used to determine when a particular performance counter is indicative of potential problems. Windows 2000 allows users to set this threshold as either over or under a particular value. OpalisRobot provides users with additional threshold conditions that will trigger if the counter value is *different than* a particular value, if it is *equal to* a particular value, or if it is *equal to or lower/higher than* a particular value.

When an alert is triggered, the Windows 2000 Task Scheduler is limited to the following responses:

- Log an entry in the application event log
- Send a network message
- Start a performance data log
- Run a program

OpalisRobot's object-oriented nature will allow users to go beyond these limitations and react in a multitude of ways. This means that with OpalisRobot you have additional responses available such as sending e-mails, paging individuals, stopping services, writing to text files, etc. More important is that the Windows 2000 Performance Monitor does not allow users to assign dependency on these responses. For instance, if a response to an alert is to run a program and the program no longer exists on the system, Robot can be configured to send a message only when the designated program fails to launch.

When looking at how performance scans are scheduled, users will find that they have only a single start and stop option. With OpalisRobot, users can assign very specific schedules for when the scans are to be active (e.g. only Monday to Thursday during office hours excluding holidays) and have these activity schedules saved as templates for reuse in other objects.

Finally, when performance counters meet thresholds and trigger alerts in Windows 2000, this may not be useful enough to operators without some sort of statistical analysis. OpalisRobot allows users to configure more specific performance scans based on the data analysis automated by the use of statistics counters.

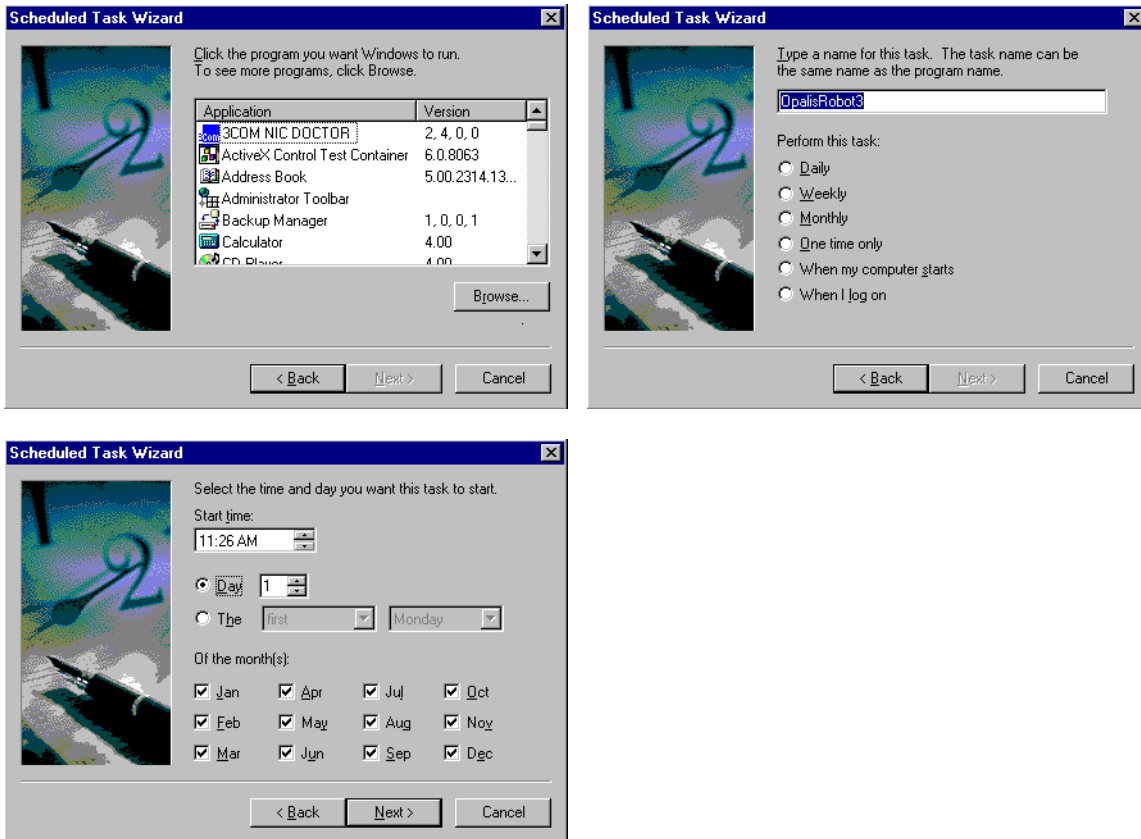
For example, ten alerts sent during the course of a day alerting that CPU usage has passed a specified threshold is not very useful for the user. With OpalisRobot, operators can use an event for performance monitoring in conjunction with statistical counters to react only to evidence that the CPU usage has maintained extremely high levels during specific time samples. Many statistical tests can be performed on a variety of different performance counters to react to specific scenarios and be recorded for future analysis by the user interested in discovering trends and causal relationships.

## Scheduling

Windows 2000 comes with a built-in task scheduler. It is used to schedule any program or task to launch when the computer starts, after a user logs onto the system, or according to a specific date/time setting. Opening the Control Panel and double-clicking on the Task Scheduler applet is one way of accessing the Task Scheduler.

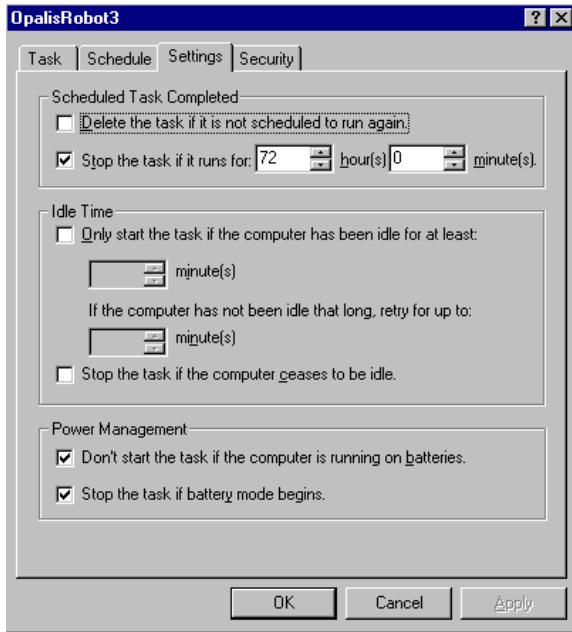
To schedule a task, the user can either use the “Add Scheduled Task” shortcut from the interface, or right-click on the interface itself and specify the creation of a new scheduled task. Both methods will begin the Scheduled Task Wizard.

The following screenshots demonstrate some of the steps contained in the Scheduled Task Wizard:

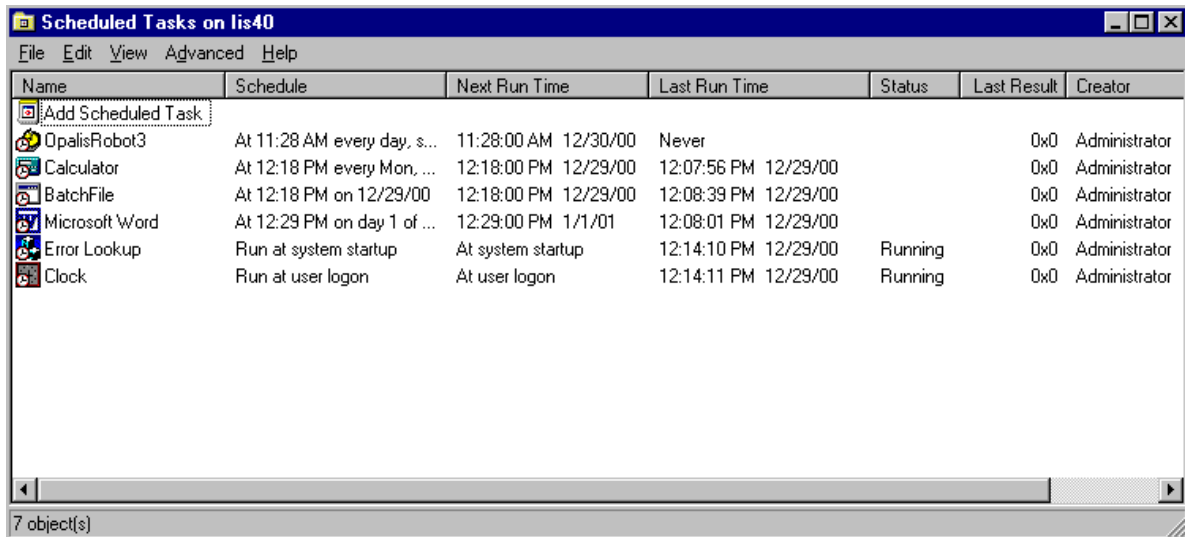


Note the various advanced settings available for each scheduled task as seen in the screenshot below. Users can set other options such as automatic task cleanup, automatically determine when tasks should conditionally stop running, and conditions based on power management. There is also a security tab where permissions, auditing and ownership can be set.





Here you see the Windows 2000 Task Scheduler interface which contains the Add Scheduled Task shortcut and various tasks that have been scheduled (including the example shown in the screenshots above):



If a detailed view is selected (as shown above), the user can see from a single interface various details regarding each scheduled task, including the schedule assignment, the next run time, the last run time, the status of the task, the last execution result (0x0 indicating success) and the creator. The user also has the option of right clicking on any task in the list and triggering it manually.

### Where OpalisRobot Excels in Task Scheduling

Users will uncover various limitations with the Windows 2000 Task Scheduler, especially when they attempt to use it for more advanced scheduling operations.

The Windows 2000 Task Scheduler does not handle task scheduling that goes beyond a frequency of minute intervals. This means that the highest frequency you can schedule a task to launch is every minute. With OpalisRobot, you can schedule tasks to launch with a frequency as high as every 5 seconds.

When launching tasks based on the occurrence of events, the Windows 2000 Task Scheduler is limited to only two events: when a computer starts and when a user logs on. OpalisRobot can automatically launch tasks based on a number of additional event occurrences, such as when disk space is low, when a remote computer is stopped (or no longer reachable), or when specific files are detected.

Although the Windows 2000 Task Scheduler allows users to assign more than one schedule to a task, users will complain about several inconveniences. The Scheduled Tasks listing will not show the multiple schedules assigned to a single task, showing only the message "Multiple Schedule Times" under the 'Schedule' column. This means that users with a large number of tasks, each being scheduled at specific times during various days, will need to open and view the properties of each individual task and examine its details.

With OpalisRobot, every task that is scheduled can have multiple Date/Time objects linked to it, but more importantly, every Date/Time object can be used with a number of different tasks. That means that a user can create a single schedule, and then assign it to a large number of tasks very easily. Along with Activity Templates, multiple scheduling of a large number of tasks with OpalisRobot is much easier to configure and manage.

Where OpalisRobot really excels becomes more apparent when users go beyond simply scheduling the launch of a task. For more advanced scheduling operations, OpalisRobot will allow users to schedule a task that will in turn launch other tasks in sequence. All successive tasks that are launched are not limited to programs. They can be performing actions such as sending e-mails, stopping services, or even making a telephone call and playing a voice-recorded message (with the Call Add-on). This is all done while implementing dependency, conditional logic and dynamic data.

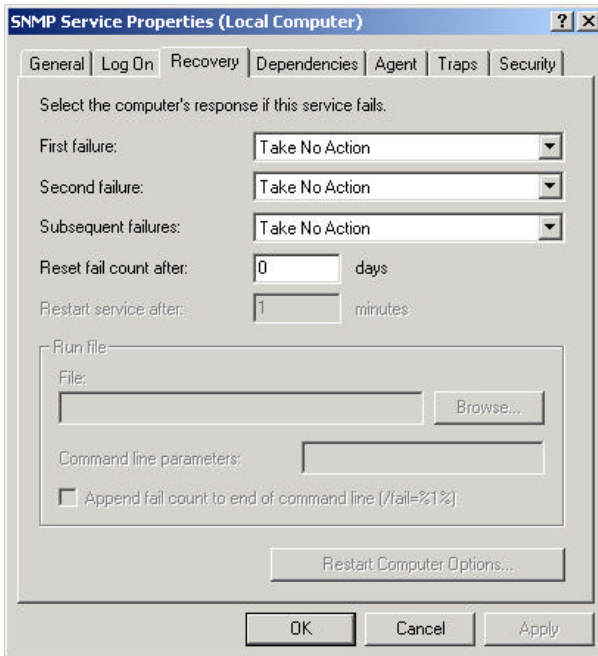
## Service Recovery

---

The final feature improvement covered in this article is Service Recovery. When Windows NT 4.0 users viewed individual NT service properties from the services listing, they were able to change the service startup type and assign a user account to the service:



In Windows 2000, users have several additional options. The most notable improvement when compared to the previous NT 4.0 version is the addition of Service Recovery:



Here the user can define an automatic response when a particular service fails, such as restarting the service, running a file, or rebooting the computer. The user can also set different responses to subsequent response failures.

## **Where OpalisRobot Excels in Service Recovery**

The addition of automated service recovery in Windows 2000 is long overdue, but the limitations again are quite evident. Users have only three service recovery options:

- Restart the service
- Run a file
- Reboot the computer

The object-oriented nature of OpalisRobot will allow recovery to consist of any combination of objects that can be used for system actions, notification, program execution, logging, etc. This means additional options for recovering services, but in hindsight of what has been covered in our feature comparison above, it also means that the other OpalisRobot benefits empower service recovery far beyond the possibilities that can be seen here using the Windows 2000 features. It is this extensive functionality, flexibility and scalability that makes using OpalisRobot the most robust way of automating a system.