



## **MonitorWare Agent 2.0**

© 2004 Adiscon GmbH



# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>4</b>
1 About MonitorWare Agent .....	4
2 Features .....	5
3 Components .....	8
Core Components .....	8
Add-On Components .....	8
4 System Requirements .....	9
<b>Part II Getting Started</b>	<b>10</b>
1 Setup .....	11
2 Creating an Initial Configuration .....	11
3 Installing Web Access .....	12
4 Obtaining a Printable Manual .....	12
5 MonitorWare Agent Tutorial .....	12
Filter Conditions .....	13
Ignoring Events .....	13
Logging Events .....	21
Time-Based Filters .....	25
Email Notifications .....	28
Alarming via Net Send .....	30
Starting Scripts and Applications in Response to an Event .....	32
Monitoring Hard Disk Space .....	35
Monitoring external Devices via PING .....	39
Monitoring External Devices via a PortProbe .....	43
<b>Part III Common Uses</b>	<b>45</b>
<b>Part IV Step-by-Step Guides</b>	<b>45</b>
<b>Part V Using Interactive Syslog Server</b>	<b>46</b>
1 Launching the Interactive Syslog Server .....	46
2 The Interactive Logging .....	47
3 Interactive Syslog Server Options .....	49
<b>Part VI Configuring MonitorWare Agent</b>	<b>50</b>
1 License Options .....	53
2 Debug Options .....	54
3 Services .....	56
Understanding Services .....	56
Syslog Server .....	57

SETP Server .....	58
Event Log Monitor .....	59
File Monitor .....	64
Heartbeat .....	67
Ping Probe .....	69
Port Probe .....	72
NT Services Monitor .....	75
Disk Space Monitor .....	77
SNMP Trap Receiver Service .....	79
<b>4 Filter Conditions .....</b>	<b>80</b>
Filter Conditions .....	80
Global Conditions .....	83
Operators .....	84
Filters .....	84
General .....	86
Date/Time .....	87
InformationUnit Type .....	89
Syslog .....	90
Event Log Monitor .....	92
NT Service Monitor .....	94
DiskSpace Monitor .....	95
SNMP Traps .....	95
Uptime .....	97
<b>5 Actions .....</b>	<b>98</b>
Understanding Actions .....	98
File Options .....	98
Database Options .....	102
Event Log options .....	105
Mail Options .....	106
Forward Syslog Options .....	111
Forward SETP Options .....	113
Start Program .....	114
Net Send .....	116
Set Status .....	116
Set Property .....	117
Call RuleSet .....	118
Discard .....	119
<b>Part VII Getting Help .....</b>	<b>119</b>
<b>Part VIII MonitorWare Concepts .....</b>	<b>121</b>
<b>Part IX Purchasing MonitorWare Agent .....</b>	<b>122</b>
<b>Part X Reference .....</b>	<b>122</b>
1 Property Replacer .....	123
System Properties .....	124
2 Complex Filter Conditions .....	124
<b>Part XI Copyrights .....</b>	<b>128</b>

---

<b>Part XII Glossary of Terms</b>	<b>128</b>
1 EventReporter .....	128
2 Millisecond .....	128
3 Monitor Ware Line of Products .....	129
4 Resource ID .....	129
5 SETP .....	130
6 SMTP .....	130
7 Syslog Facility .....	131
8 TCP .....	131
9 UDP .....	131
10 Upgrade Insurance .....	132
11 UTC .....	132
<b>Index</b>	<b>0</b>

# 1 Introduction

## 1.1 About MonitorWare Agent

### **MonitorWare is an integrated, modular and distributed solution for system management.**

Network administrators can continuously monitor their systems and receive alarms as soon as important events occur.

MonitorWare is a distributed and extensible system. At its very base is the MonitorWare Agent which includes all data gathering and real-time notification functions. This manual concentrates on the MonitorWare Agent.

The agent is run on the systems to be monitored and provides the base functionality. It can gather data from numerous sources, like the Windows Event Log, Syslog enabled devices (routers, firewalls ...) or text files to name a few. The MonitorWare Agent supports very flexible and powerful local filtering and processing of these events. Based on a powerful rule processor, events can be forwarded, acted on or discarded - all at the discretion of the system administrator. Given this engine, even a stand-alone MonitorWare Agent performs useful work. For example, in a small environment, it can generate alert emails at the occurrence of specific events.

Larger environments will consolidate all agent data in a central repository, for example the MonitorWare event database or combined log files. The database is the source of information for all reporting and analysis modules of the MonitorWare system. By default, it can be created with MySQL, Microsoft Access or Microsoft SQL Server (also available as cost-free MSDE). As standard SQL and ODBC are being used, it is easily adaptable to other database systems. For example, we know that many customers use it successfully with Oracle databases.

A number of different modules work on this consolidated database or the log files to achieve various activities. These modules include scheduled reporting facilities like [MonitorWare Console](#) for analysis, a web interface or [MoniLog](#) reporting.

Currently under development is an enterprise configuration manager, which facilitates configuration of the MonitorWare system on enterprise scope. With the MonitorWare Enterprise Manager, groups of configurations can be created (e.g. for Syslog servers, NT event log monitors, consolidation servers and the like). These function-focused groups can then be automatically applied to machine groups. So a whole MonitorWare system - no matter how large - can be administered from a single MonitorWare Enterprise Manager. If you are interested in the enterprise configuration system, please mail us at [support@adiscon.com](mailto:support@adiscon.com) to become enrolled in our beta program.

MonitorWare does also integrate with other management related Adiscon products like EventReporter and [WinSyslog](#). In fact, it uses common terms and methods wherever possible, so upgrading from these solutions to the full MonitorWare system is easy.

For a complete overview over the MonitorWare line of products, please visit [www.monitorware.com](http://www.monitorware.com).

## 1.2 Features

### Complete Windows Event Monitoring

Automatically monitor Windows Event Logs and application log files. All Event Logs – including the Windows 2000 specific extensions – are fully processed. Application log file monitoring provides support for virtually any application that logs to a text file. Examples are Web server log files or Oracle error logs. Even Windows itself stores some information not in the event log but application log files (like the DHCP log files).

### Active Network Probes

Ping and port probe services allow monitoring of both local and remote systems and services. These services are not restricted to Windows machines – virtually any existing service can be used with these probes. Good examples are LINUX based web and mail servers or firewalls. But our probes don't restrict you to an OS – even if you have a server running on a mainframe, MonitorWare can check its operational state.

Failing systems and services are detected and alert be generated.

### Monitor Windows' Services and Disk Space

The Windows service monitor and disk space monitor check the local machine. Failing services and low disk space are quickly detected and can be used to trigger notifications or even corrective actions before problems arise.

### External Events

Events are accepted via a standard Syslog server and hence all Syslog-enabled devices can be included in the MonitorWare system. This includes popular devices like routers and switches as well as printers and a large number of UNIX/Linux based systems and applications. Virtually all currently existing network devices support Syslog – so MonitorWare Agent can monitor all of them.

To reach an even broader device range, MonitorWare Agent not only supports standards compatible Syslog but also it supports popular extensions like Syslog over .

### Scalability

The MonitorWare system is modular and highly scalable. If a single server is to be monitored, MonitorWare Agent can provide all monitoring and alerting needs. However, multiple MonitorWare Agents in a complex, hierarchical network can talk to

each other and provide both local and central alerting and event archiving.

## **Event Archiving**

All incoming events – no matter what source they came from – can be stored persistently. Options include archiving in databases as well as log files.

## **Alerting**

All incoming events – no matter what source they came from – can be stored persistently. Options include archiving in databases as well as log files.

## **Powerful Event Processing**

MonitorWare Agent is powerful and flexible rule engine processes all events based on a configured set of actions. An unlimited number of rules and actions allows tailoring to the specific needs.

## **Zero-Impact Monitoring**

MonitorWare Agent has no noticeable impact on system resources. It was specifically written with minimal resource usage in mind. In typical scenarios, its footprint is barely traceable. This ensures it can also be installed on heavily loaded servers.

## **Robustness**

MonitorWare Agent is written to perform robust even under unusual circumstances. The reliability of the is proven since 1996.

## **Ease of Use**

MonitorWare Agent is easy to install and configure. Comprehensive step-by-step guides and wizards help administrators with setting up even complex systems.



---

## **Firewall Support**

Does your security policy enforce you to use non-standard ports? MonitorWare Agent can be configured to listen on any TCP/IP port for Syslog messages.

## **Syslog Support**

NT Event Messages can be forwarded using standard Syslog protocol. NT severity classes are mapped to the corresponding Syslog classes. Codes are fully supported.

## **Send Syslog Test Message**

The MonitorWare Agent client comes with "Send Syslog Test Message". This option enables us to check if Syslog Messages being sent properly to our destination or not.

## **SETP Support**

NT Event Messages can be forwarded using protocol. Windows Event Log are monitored successfully as well.

## **SNMP Trap Receiver**

A new key feature added in this version of Monitor Ware Agent. SNMP Trap Receiver allows you to receive SNMP messages.

## **Runs on large Variety of Windows Systems**

Windows 4.0, 2000, XP, 2003; Workstation or Server – MonitorWare Agent runs on all of them. We also have Compaq (Digital) ALPHA processor versions on platforms supporting this processor (service only, available on request).

## **Multi-Language Client**

The MonitorWare Agent client comes with multiple languages ready to go. Out of the box, English, French and German are supported. Other languages will be added shortly. Languages can be switched instantly. Language settings are user-specific; so multiple users on the same machine can use different languages.

## 1.3 Components

### 1.3.1 Core Components

#### **MonitorWare Agent Configuration Client**

The MonitorWare Agent Configuration Client – called "the client" - is used to configure all components and features of the MonitorWare Agent. The client can also be used to create a configuration profile on a base system. That profile can later be distributed to a large number of target systems.

However, the client can only configure one machine at a time and has no notation of machine or functional groups. For enterprise-wide administration, use the MonitorWare Configuration Manager, available as a separate product.

#### **MonitorWare Agent Service**

The MonitorWare Agent Service – called "the service" - runs as a Windows service and carries out the actual work.

The service is the only component that needs to be installed on a monitored system. The MonitorWare Agent service is called the product "engine". As such, we call systems with only the service installed "engine-only" installations.

The service runs in the background without any user intervention. It can be controlled via the control panel "services" applet or the "Computer Management" MMC under Windows 2000. The client can also be used to control service instances.

### 1.3.2 Add-On Components

There are a number of optional components available as free downloads.

All optional components work with the MonitorWare Common Database Format.

#### **Interactive Syslog Server**

The interactive Syslog server is helpful for quick analysis and troubleshooting. It displays incoming events in the interactive session.

Though it is not a core component, it is included in the MonitorWare Agent install set.

#### **MonitorWare Web Access**

Web access is a convenient facility to access MonitorWare gathered events over the

web. All major browsers are supported. Web Access is fully integrated with Microsoft's IIS, so multiple security layers can be used.

MonitorWare Web Access is included in the MonitorWare Agent install set. It gets installed automatically when IIS is present on the target machine. However, it is fully optional and need not be installed.

## MonitorWare Console

MonitorWare Console facilitates the Network Administrators to gather valuable information about their networks and offers them strong analytical abilities with which they can examine their network proficiently against countless problems including security breaches. Using the Views and Reporting Modules of MonitorWare Console, you can find the problematic areas in your network very efficiently and promptly. As a network administrator, you would not only like to find the problems but also their solutions. MonitorWare Console's Knowledge Base Module is exactly meant for this purpose. In short, MonitorWare Console is a very powerful tool that will facilitate the Network Administrators to scrutinize their networks from tip to toe and will give an in-depth perspective about what's going on in their system.

For further details please visit the MonitorWare Console website.

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

## 1.4 System Requirements

The MonitorWare Agent has minimal system requirements. The actual minimum requirements depend on the type of installation. If the client is installed, they are higher. The service has minimal requirements, enabling it to run on a large variety of machines – even highly utilized ones.

### Client

- The client can be installed on Windows NT 4.0 and above. This includes Windows 2000, Windows XP and the 2003 servers. The operating system variant (Workstation, Server ...) is irrelevant.
- The client uses XML technology. Unfortunately, operating system XML support is only available if at least Internet Explorer 4.01 SP1 is installed.
- The client requires roughly 8 MB RAM in addition to the operating system minimum requirements. It also needs around 10 MB of disk space.
- The client is available for Intel based systems, only.

### Service

- The service has fewer requirements. Most importantly, it does not need Internet Explorer to be installed on the system.
- It works under the same operating system versions.
- Additionally, it should perform well under NT 3.51, but as we have not yet received any request for supporting this operating system version, no tests have been conducted yet. This will be done upon request.
- The service also by design supports the Compaq/Digital APHA processor, but again has not been ported yet due to missing demand. If you are in need of such a version, please contact Adiscon at support@adiscon.com
- At runtime, the base service requires 5 MB of main memory and less than 2 MB of disk space. However, the actual resources used by the agent largely depend on the services configured.
- If the MonitorWare Agent shall just monitor the local systems event log, impact on the monitored system is barely noticeable, if at all visible.
- If the MonitorWare Agent acts as a central Syslog server receiving hundreds of messages per second, it will need many more resources. Even then, the actual load is depending on the actions carried out. Storing the messages to text files is much less performance intense than writing them to a database table – especially if the database engine is located on the same machine. As such, there is no single guideline for hardware sizing. It needs to be adapted to the expected workload. We have created an article on [performance optimization for syslog server operations](#), which you may want to visit.
- Please note, however, that the service is specifically optimized to handle high throughput including message bursts (for example received via syslog).
- If you expect high volume burst and carry out time consuming actions (for example database writes), we highly recommend adding additional memory to the machine. Even 64 MB additional memory will do nicely. A typical Syslog message (including overhead) will take roughly 1.5 KB. With 64 MB, you can buffer up to 50,000 messages in 64 MB.
- Please note, MonitorWare Agent is capable of storing such bursts temporarily in memory even if the machine would otherwise be too slow to process the messages.

## MonitorWare Web Access

- MonitorWare Web Access requires Microsoft Internet Information Server (IIS) version 3 or higher to be present on the machine where Web Access is to be installed.
- Please note that Web Access can be installed on a machine different from the service as long as that machine can access the MonitorWare database.

## 2 Getting Started

MonitorWare Agent can be used for simple as well as complex scenarios. This chapter provides a quick overview of the agent and what can be done with it. Most importantly, it contains a tutorial touching many of the basic tasks that can be done with MonitorWare Agent as well as pointer on how to setup and configure.

Be sure to at least briefly read this section and then decide where to go from here - it will definitely be a worth time spent.

## 2.1 Setup

Installing the MonitorWare Agent is simple and easy. A standard setup program installs the application.

There are a number of different download versions of the product available. The main difference is whether or not a current version of the Microsoft Windows Installer program is included. If you use recent software (e.g. Windows XP or Windows 2003 Server), you can typically use the smallest install set. Install sets have different names. Those ending in "max" are typically the version for older operating systems without a current installer. If in doubt, use an install set whose name ends in "max". All files are direct install sets, so there is no need to unzip them to find a setup.exe or such.

Depending on the download directory, the setup program may also be supplied in a ZIP file.

## 2.2 Creating an Initial Configuration

MonitorWare Agent actually consists of five products in one. MonitorWare Agent can work as

### **Data gatherer**

Here, it gathers event data from important sources like Windows event logs, text files, ping and port probes and the like.

### **Real Time Alerter**

Alert conditions can be detected in real time and alerts be issued. Alerts can be sent via email and various other means. Alerts based on data gathered by the data gatherers configured.

### **Automatic Admin Actions**

Depending on certain events, administrative actions can be automatically initiated, for example the deletion of temporary files in a low-disk space condition.

### **Relay Server**

MonitorWare Agent can be used to build, highly scalable, complex systems with relay servers between locations or networks. As a relay server, it will forward incoming events to another instance of MonitorWare Agent or a Syslog daemon.

## Event Repository

All gathered event data can be stored in a repository. The repository is a database providing the base for all other MonitorWare products. Events can also be stored in text files. With a specific configuration, a secure log repository can be created for auditing purposes.

MonitorWare Agent can perform any mix of the five functions on a given machine. There are no limits inside the product. Right after installation, however, it is not configured for any of the above functions. So in order to have it do some useful work, it needs to be configured.

## 2.3 Installing Web Access

MonitorWare Web Access is installed if Microsoft IIS is present on the target machine. In that case, a web "WebAccess" is created.

After setup, Web Access is present, but needs to be configured. With this release, configuration is done by editing the ConfigSettings.asp file inside the Web Access directory. This can be done with any plain text editor like notepad (do not use Word or any other text processor!). ConfigSettings.asp contains comments on which parameters can and need to be changed. Most notable, the database connection needs to be updated.

Adiscon is currently working on a new, php-based web interface. Please email [support@adiscon.com](mailto:support@adiscon.com) if you are interested in trialing this.

## 2.4 Obtaining a Printable Manual

A printable version of the manual can be obtained at <http://www.monitorware.com/en/Manual/>

The manuals offered on this web page are in PDF format for easy browsing and printing. The version on the web might also include some new additions, as we post manual changes – including new samples – frequently and as soon as they become available.

## 2.5 MonitorWare Agent Tutorial

The goal of this tutorial is to provide a rough overview over the MonitorWare Agent as well as some typical uses. It is in no way complete, but should help in understanding MonitorWare Agent and how it can be configured to suit your needs.

In the tutorial, we start by describing and focusing on the filter conditions, as these are often needed to understand the usage scenarios that follow below.

MonitorWare Agent gathers network events – or "information units" as we call them – with its services. Each of the events is then forwarded to a rule base, where the event is serially checked against the different rule's filter conditions. If such a condition evaluates to true ("matches"), actions associated with this rule are carried out (for example, storing the information unit to disk or emailing an administrative alert).

### 2.5.1 Filter Conditions

For every rule, filter conditions can be defined in order to guarantee that corresponding actions are executed only at certain events.

These filter conditions are defined via logical operators. Boolean operators like "AND" or "OR" can be used to create complex conditions.

If you are not so sure about the Boolean operators, you might find the following brush-up helpful:

**AND** – All operands must be true for the result to be true. Example: AND (A, B): Only if both A and B are true, the result of the AND operation is true. In all other cases, it is false.

**OR** – if at least one of the operands is true, the end result is also true. Example: OR (A, B): The end result is only false if A and B are false. Otherwise, it is true.

**NOT** – negates a value. Example: NOT A: If A is true, the outcome is false and vice versa. There can only be a single operand for a NOT operation.

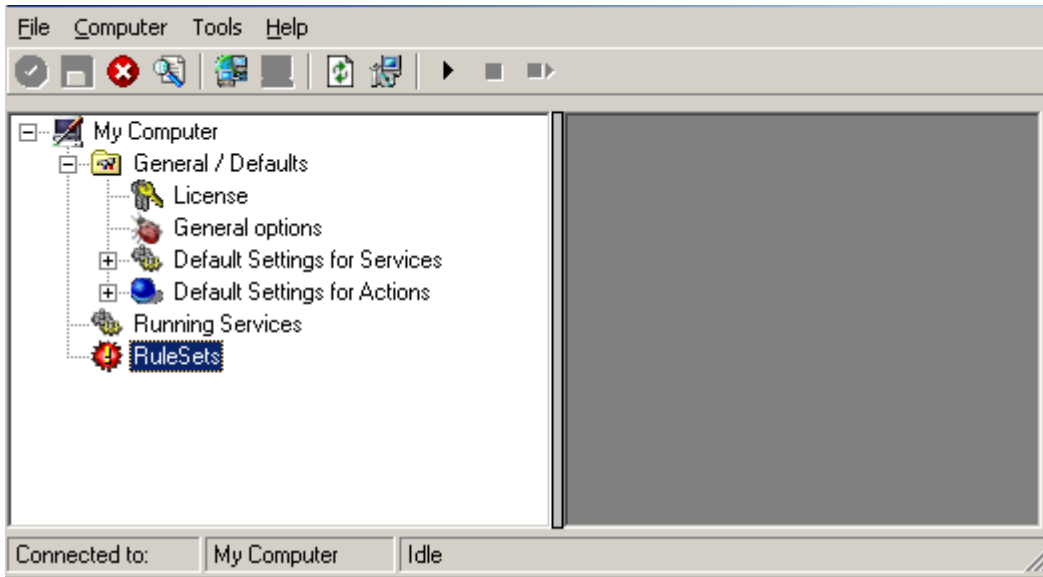
### 2.5.2 Ignoring Events

In most cases, there are some events that we would like to ignore. Events we know to occur often and we also know to be of no interest for what we try to accomplish. Most often, there are events that we do not want to store in our log files and that should also not cause any other action.

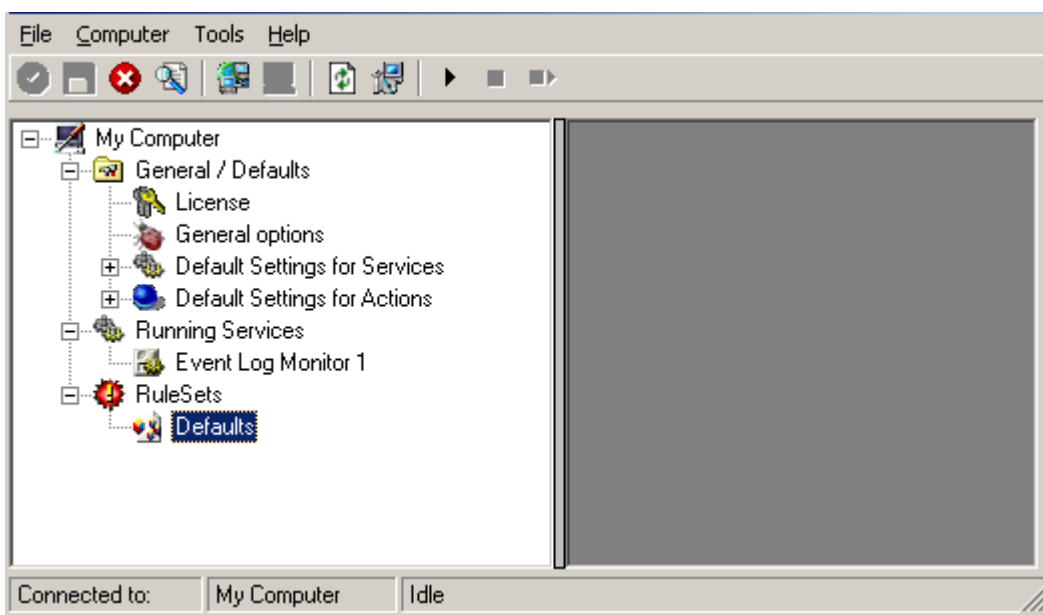
We handle these events on top of our rule set. This ensures that only minimal processing time is needed and they are discarded as soon as possible.

In this tutorial, we define a filter that discards such events. In our example, we assume that Events with the ID105, 108 and 118 are not required. Please note that for simplicity reasons we only filter based on the event ID. In a production environment, you might want to add additional properties to the filter set.

In this sample, no service or rule set is yet defined. It is just a "plain" system right after install, as can be seen in the following screen shot:

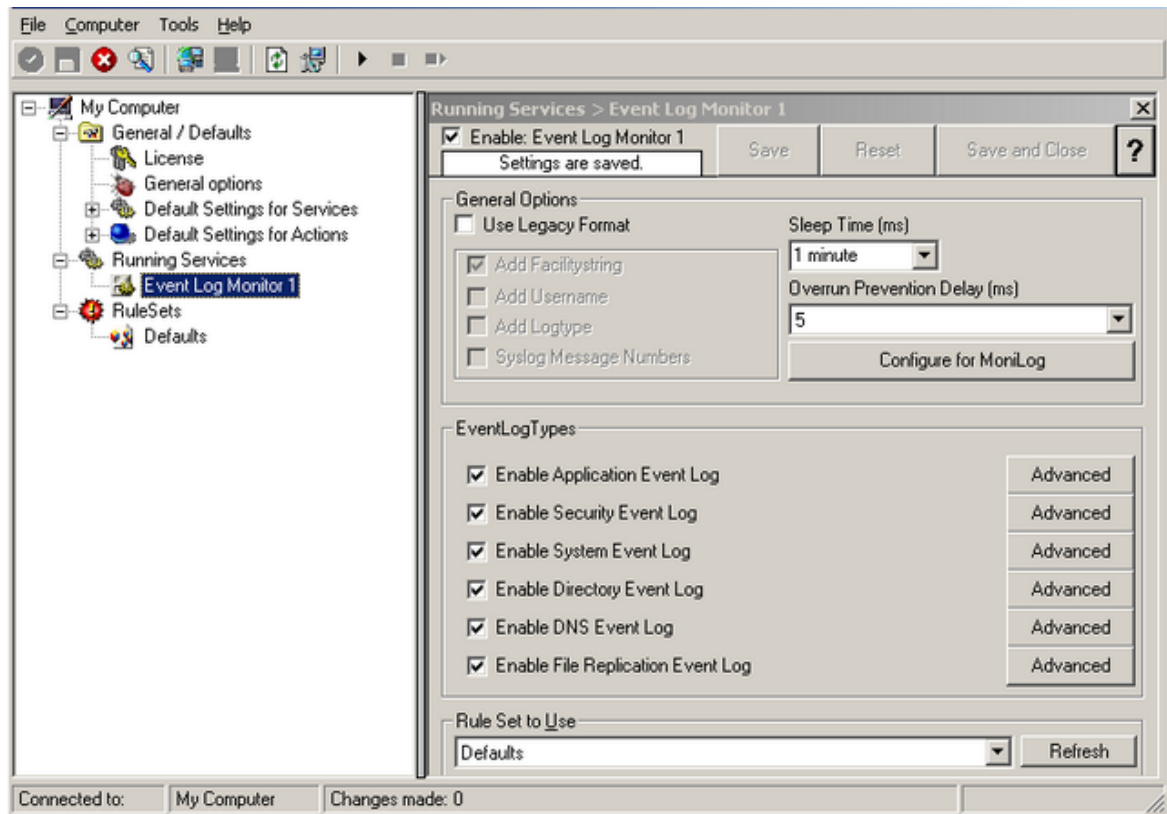


We begin by defining a rule set. Right-click on "RuleSets" and choose "Add RuleSet" from the context menu. Type in a name of your choice. In this tutorial, we use the name "Defaults". Click on "Next". Leave all as is in the next dialog. Click "Next", then "Finish". As can be seen in following screen shot, the rule set "Defaults" has been created but is still empty.



Of course we can only use a rule if we configure a corresponding service. To do so, right-click on "Running Services" and choose "Service" in the context menu. Then select "Add Services" and "Event Log Monitor". Provide a name of your choice. In our sample, we call the service "Event Log Monitor". Leave all defaults and click "Next", then "Finish". Now click on "Event Log Monitor" under "Running Services". Your screen should look as follows:

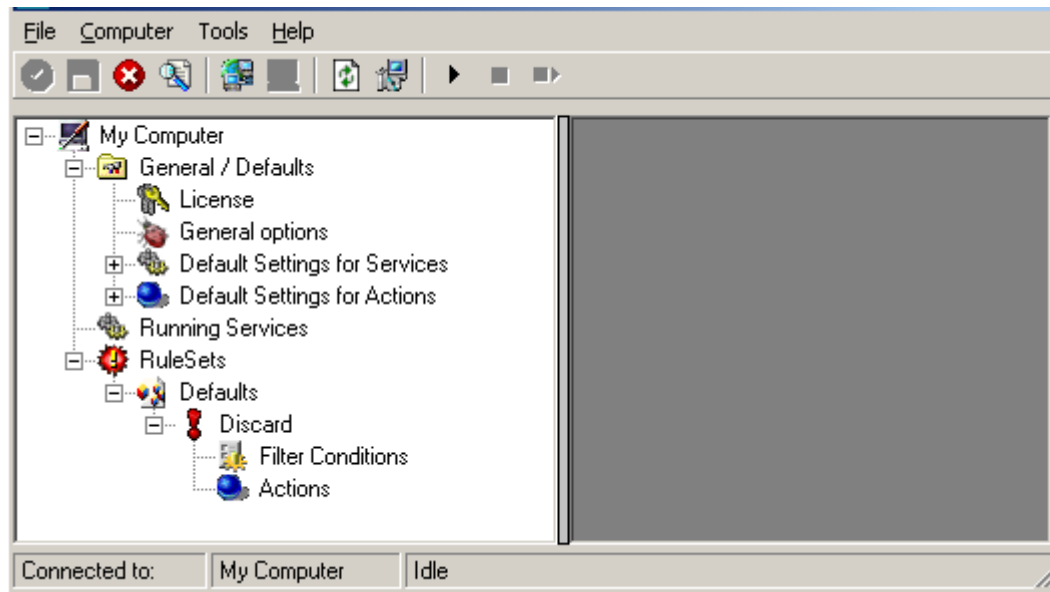




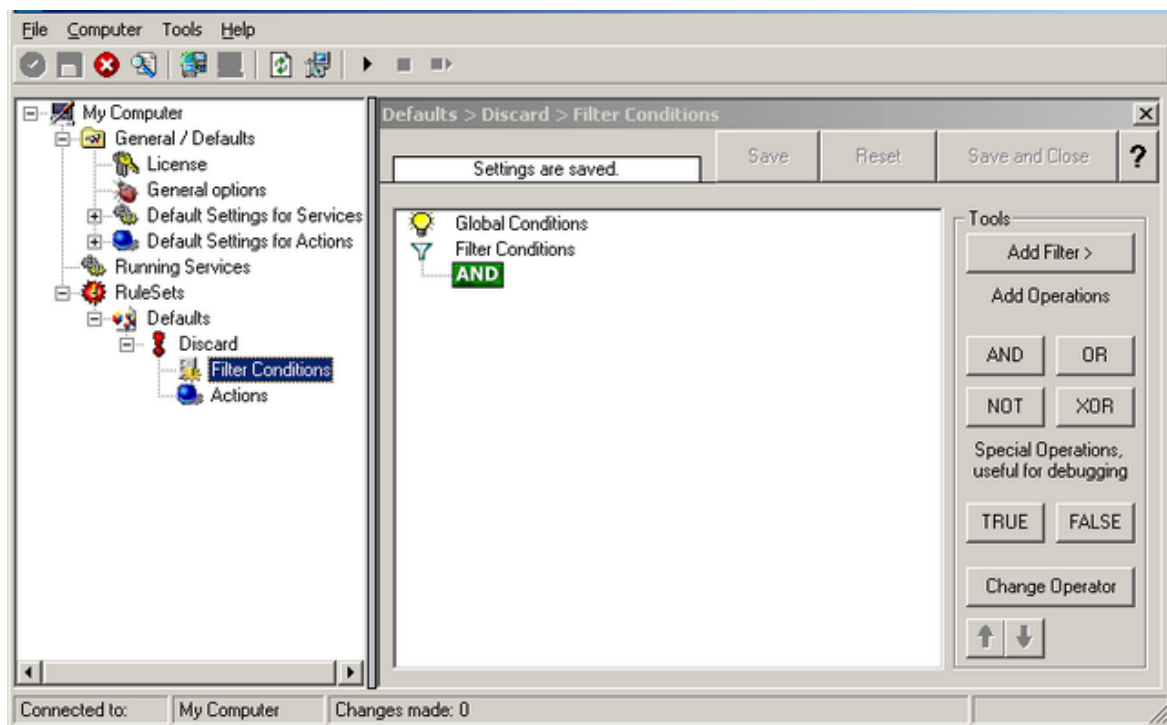
Because there we created the "Defaults" rule set initially, it is shown as the rule set to use for this service. For our purposes, that is correct. To learn more on the power of rule set assignments, see other sections of this manual.

Now we will do something with the data that is generated by the event log monitor. To do so, we must define rules inside the rule set.

In the tree view, right-click "Defaults" below "RuleSets". Then, click "Add Rule". Choose any name you like. In our example, we call this rule "Discard". Then, expand the tree view until it looks like the following screen shot:



Click on "Filter Conditions" to see this dialog:



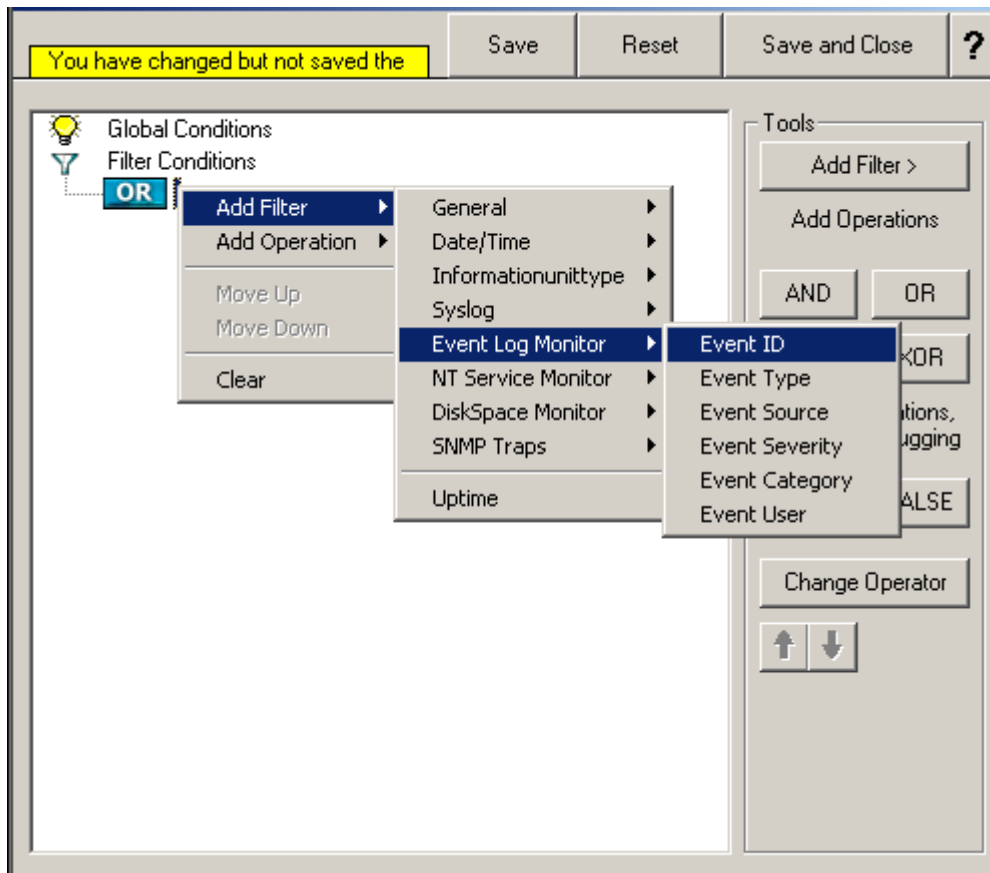
In that dialog, we will define our filter. Remember: we are about to filter those events, which we are **not** interested in. As we would like to discard multiple events, we need the Boolean "OR" operator in the top-level node, not the default "AND".

Thus, we need to change the Boolean operator.

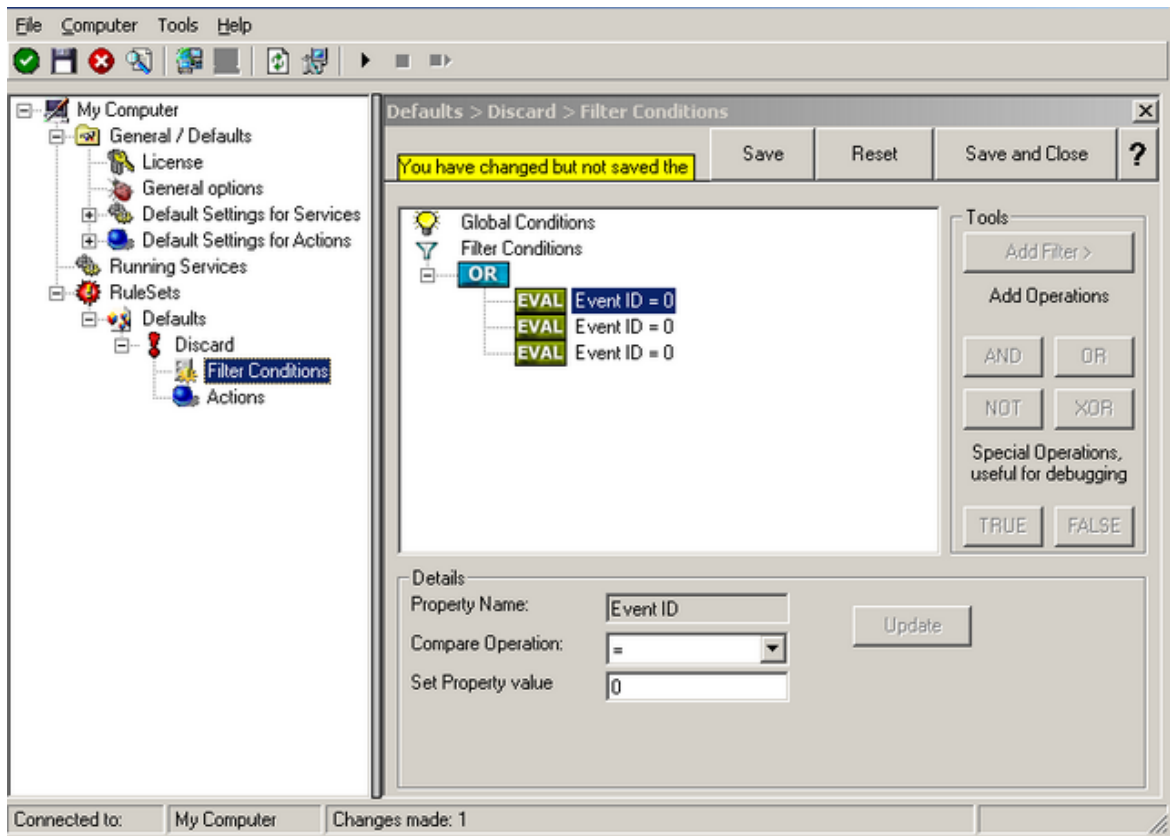
There are different ways to do this. Either double-click the "AND" to cycle through the supported operations or select it and click "Change Operator". In any way, the Boolean operation should be changed to "OR".

We filter out "uninteresting" events via their event id. Again, there are different ways

to do this. In the sample, we do it via right clicking the "OR" node and selecting "Add Filter" from the pop up menu. This can be seen in the screen shot below:

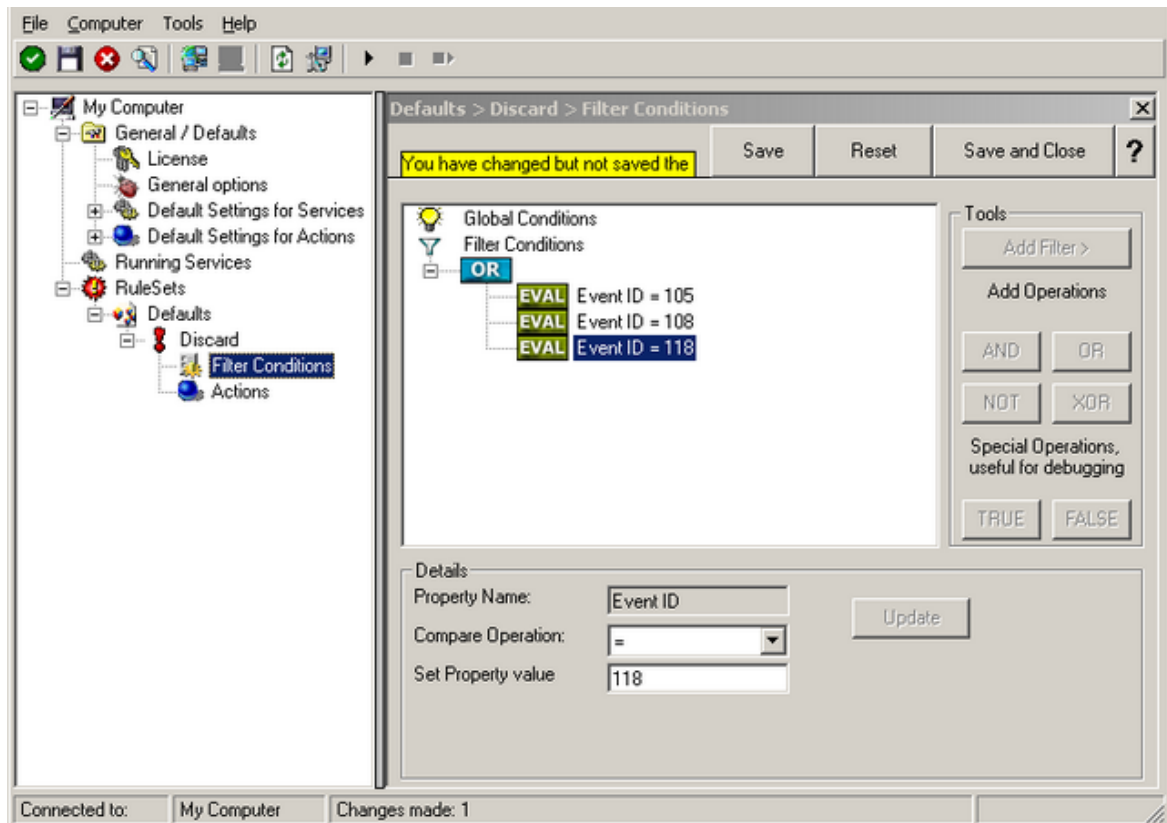


I prefer to add all three-event id property filters first and later on change the event id to the actual value I am looking for. When you have added them, it should look as follows:

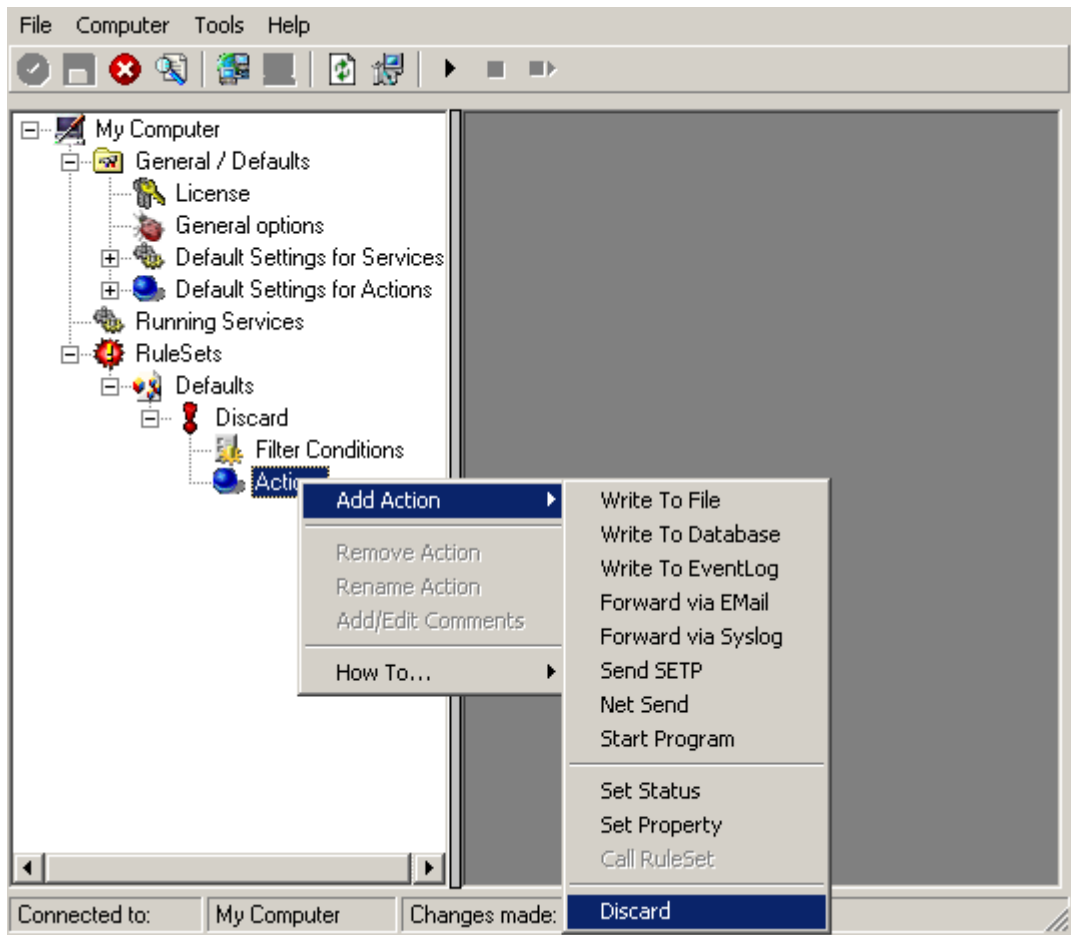


In order to enter the actual values, select each of the three filters. A small dialog opens at the bottom of the screen. There you enter the values you are interested in. In our sample, these are IDs 105, 108 and 118. As we are only interested in exactly these values, we do a comparison for equality, not one of the other supported comparison modes.

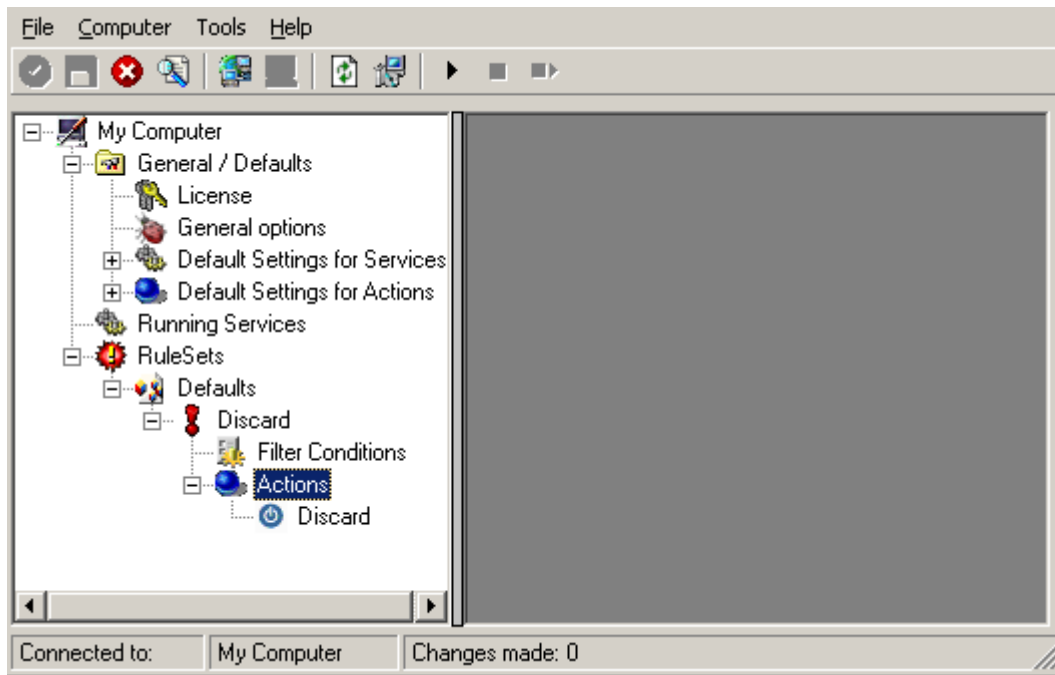
When you have made the updates, your screen should look as follows:



Save the settings by clicking the (diskette-like) "Save" button. We have now selected all events that we would like to be discarded. In reality, these are often far more or a more complicated filter is needed. We have kept it simple so that the basic concept is easy to understand – but it can be as complex as your needs are. Now let us go ahead and actually discard these events. This is done via an action. To do so, right-click on "Actions" and select "Discard."



Again, name the action as you like in the following dialog. We use "Discard" as this is quite descriptive. Select "Next" and then "Finish" on the next page. Your screen should look like follows:



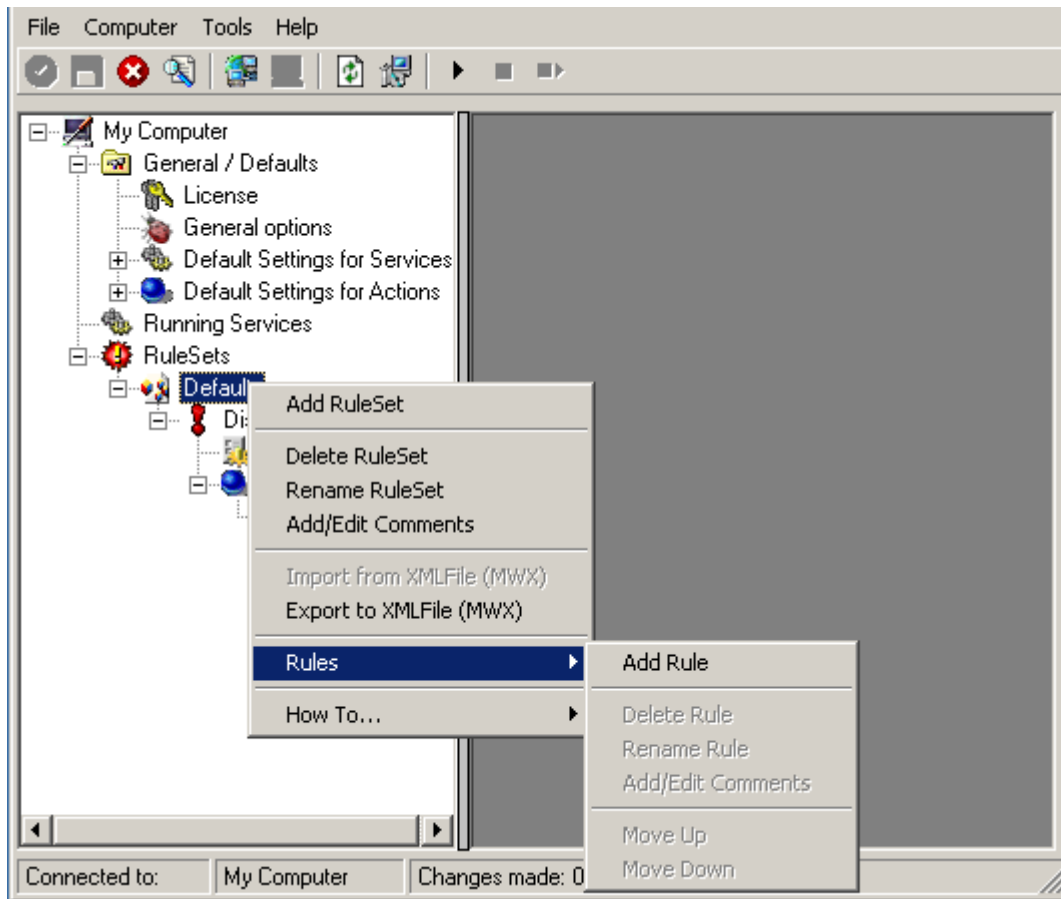
This concludes the definition of our first rule.

If we would start MonitorWare Agent service now, all events with IDs 105, 108 and 118 would be handled by this rule and thus be discarded. All other events will not cause the filter condition to evaluate to true and thus those would be left untouched. Consequently, only these other events will flow down to rules defined behind the "Discard" rule. Obviously, our configuration effort is not yet completed. We just finished a first step, excluding those events that we are not interested in. And of course, in reality you need to decide which ones to discard in a real rule set.

### 2.5.3 Logging Events

Often, a broad range of events (or information units as we call them) needs to be stored persistently so that you can review and analyze them if there is need. As such, we are in need of a rule that persists the events. In our sample, we choose to work with a text log file (not a database, which we also could use). We will now create a rule to store all those events not discarded by the previous rule.

To do so, right click the "Defaults" rule set as shown below. Then, select "Rule Sets" and "Add Rule":



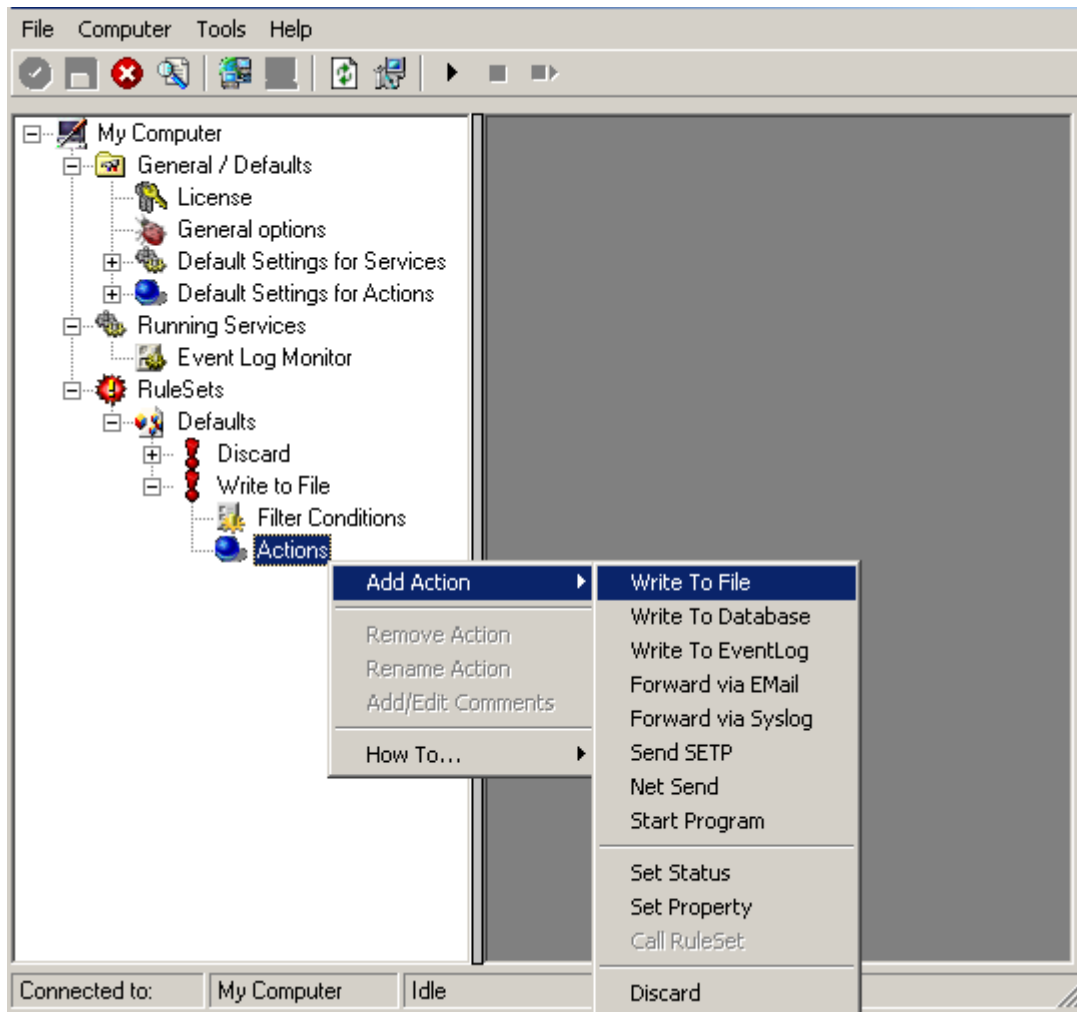
Use a name of your choosing. In our sample, we call this rule "Write To file".

This rule should process **all** events that remained after the initial discard rule. As such, we do not need to provide any filter condition (by default, the filter condition matches always).

Since we want to store all still open Events with help of this rule, we do not require any filter rules here. However, a corresponding action must be defined. Therefore, we just need to define the action:

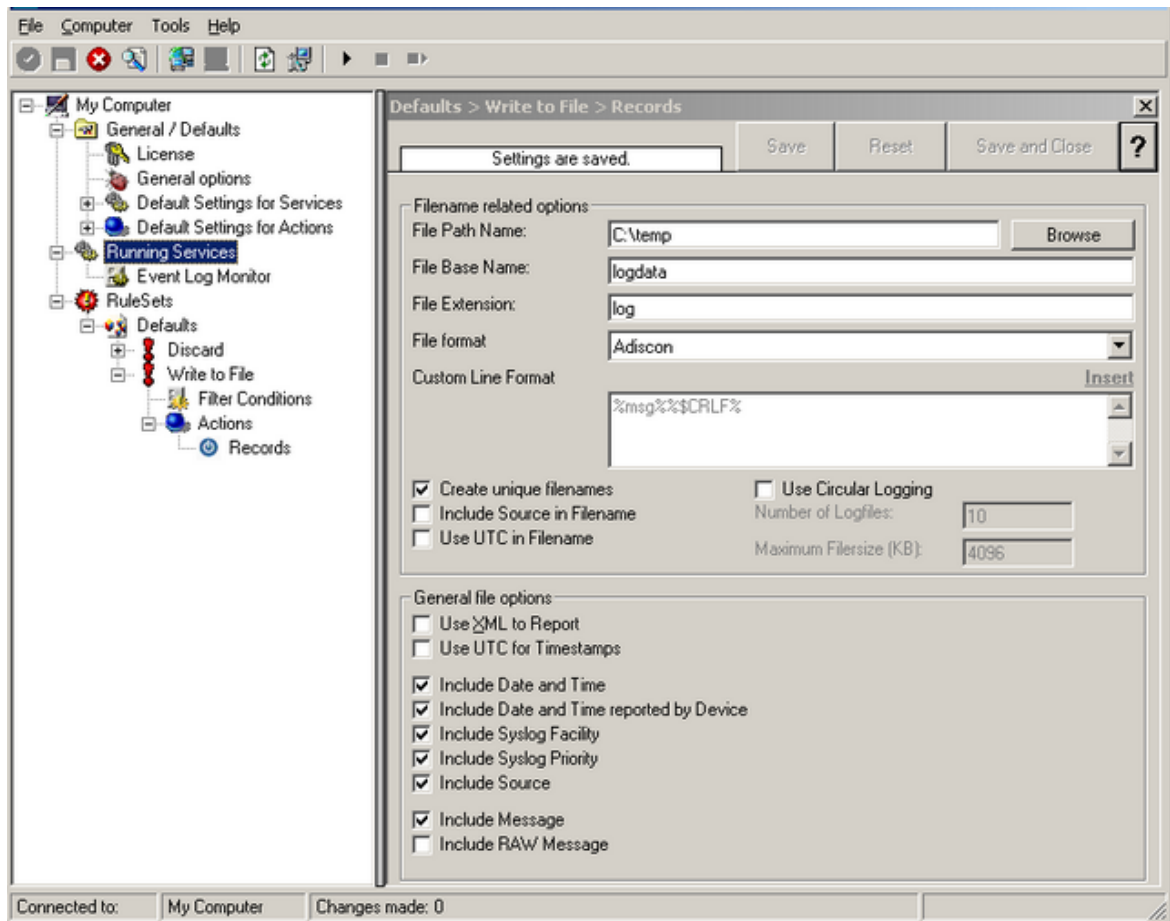
To do so, expand "Write To file" and right-click "Actions". Select "Add Action", then "Write To file" as can be seen below:





Again, choose a name. Do not modify the defaults. In our sample, we call this action "Records". Click "Next", then "Finish."

Now the tree view contains a node "Records", which we select:



### Important

make sure that the folder specified exists! If it does not exist, MonitorWare Agent will not write the log file. MonitorWare Agent will also **not** create the folder by itself. So if the folder does not exist, be sure to either create it or select a different (existing) one. In our sample, we also change the file base name to "logdata". This was just done out of personal preference. There is no need to do so, but it may be convenient for a number of reasons.

### Summary

What did we do so far? All events from the Windows event log are passed through our rule engine and rule filters. Certain events are discarded and the remaining events are stored to a text file on the local disk (for later review or post-processing). We can now do a quick test: Start MonitorWare Agent by hitting the start button seen below:



The log file should be created in the path you have specified. Open it with notepad. You should see many events originating from the event log. When you re-open the log

file, new events should appear (if there were any new events in the Windows event log). The file is not easily readable. Most probably, you have created it for archiving purposes or to run some external scripts against it. For review, we recommend using either the web interface or the upcoming MonitorWare Console add-on. Please note that the current date is appended to the log file. This facilitates file management and archiving. The format is "logdata-YYYY-MM-DD.log". You have now learned to define rules and actions. The following chapters thus will not cover all details of this process. If in doubt, refer back to these chapters here.

## 2.5.4 Time-Based Filters

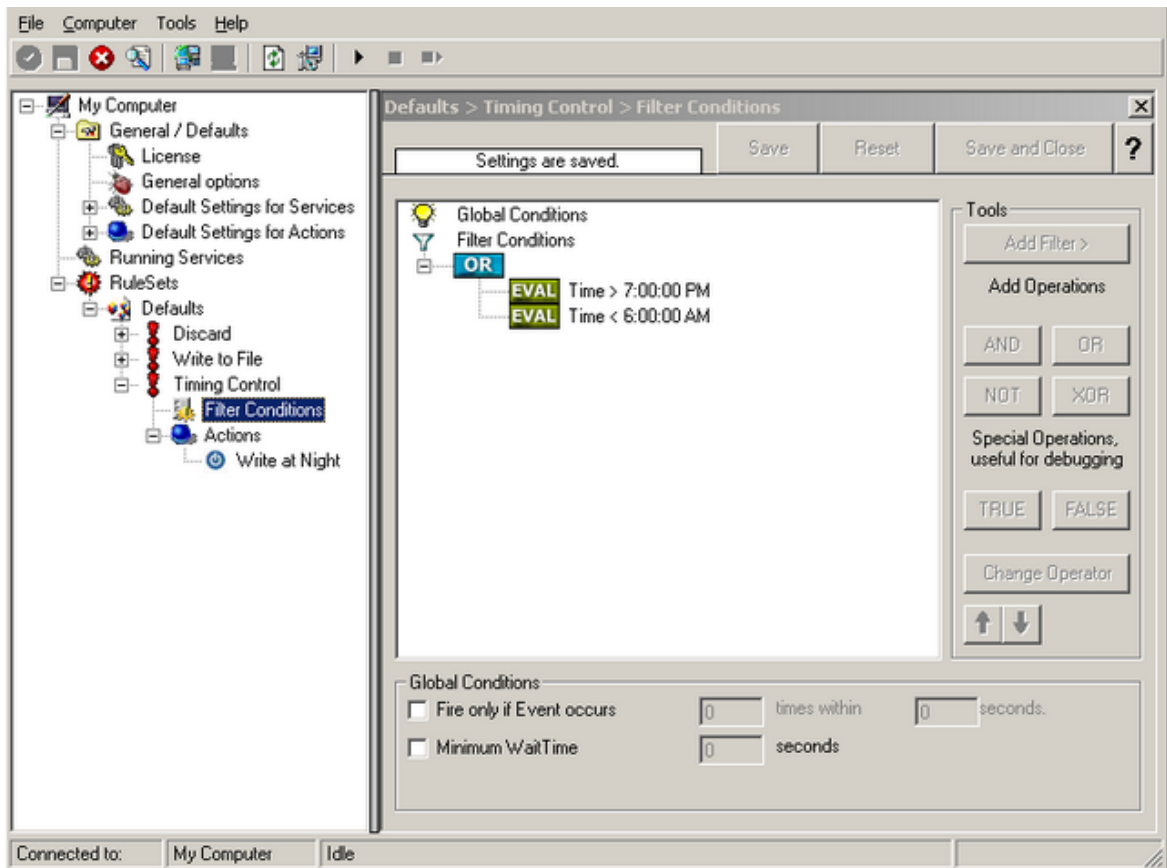
Time based filters are especially useful for notifications. For example, a user login is typically a normal operation during daytime, but if there are no night shifts, it might be worth generating an alert if a user logs in during nighttime. Another example would be a backup run that routinely finishes during the night. If we see backup events during the day, something might be wrong.

Similarly, there are a number of other good reasons why specific actions should only be applied during specific time frames. Fortunately, MonitorWare Agent allows defining complex time frames. In this tutorial, though, we focus on the simple ones.

Let us first define a sample time-based filter that applies a nightly time frame. In fact, there are many ways to do this. We have used the method below, because it is straightforward and requires the least configuration work.

To make matters easy, we use this filter condition just to write nightly event log data to a different log file. In reality, time based filters are often combined with other conditions to trigger time based alerts. However, this would complicate things too much to understand the basics.

In the sample below, an additional rule called "Timing Control" has been added. It includes a time-based filter condition. Only if that condition evaluates to "true", the corresponding action is executed. This action can be "Write to Database" or "Write to File". Here we had selected "Write to File" action and renamed it as "Write at Night". Please note: we use the 12-hour clock system below.



All events generated by services binding to our rule set "Defaults" will now also be passed along the "Timing Control" rule set. If these events come in nighttimes between 07:00:01 PM and 5:59:50 AM, the action "Write at Night" is executed. Please note that the use of the "OR" operator is important because either one of the time frames specified does apply. This is due to the midnight break.

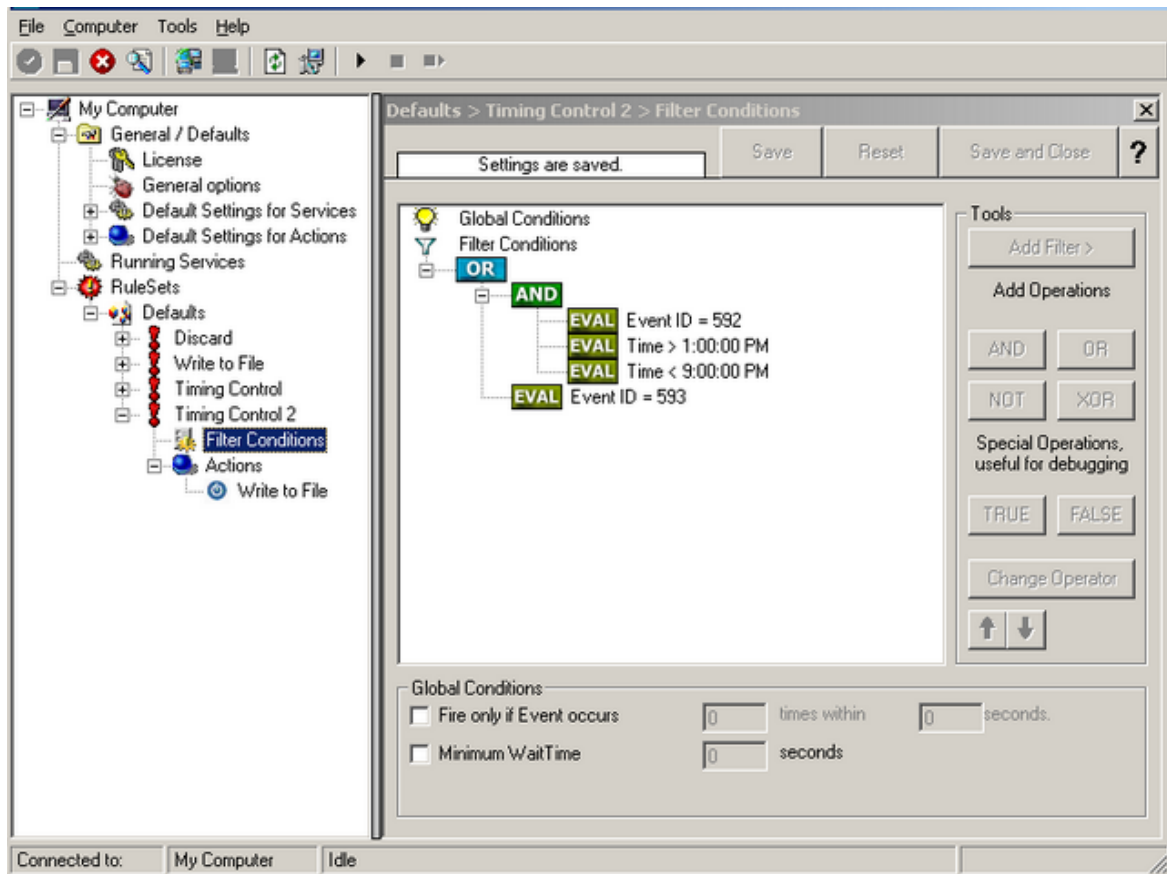
If an event comes in at 08:00:00 AM in the morning, the action will not be called – it is outside of the specified time frame:

08:00:00 AM > 07:00:00 PM = *false*  
 08:00:00 AM < 06:00:00 AM = *false*

If the very same event comes in at 08:00:00 PM in the filter condition evaluates to true and the action will be executed.

08:00:00 PM > 07:00:00 PM = *true*  
 08:00:00 PM < 06:00:00 AM = *false*

As stated earlier, time frames are most often used in combination with other filters. Here is a more complex example:



In this example, we will call the configured actions if events with ID 592 occur between 01:00:01 PM and 08:59:59 (roughly 9 PM). We will also execute the configured actions if event ID 593 occurs. Please note that in the case of 593 events, the time filter does not apply due to the used Boolean operations.

In this sample, you also notice that we use an "AND" condition to build the time frame. The reason is that there is no implicit midnight boundary for our time frame as was in the first sample. As such, we need to employ "AND" to make sure the events are WITHIN the specified range.

Now let us look at some sample data:

We receive a 592 event at 07:00:00 AM sharp:

```

Event ID = 592           = true
07:00:00 AM > 01:00:00 PM = false
07:00:00 AM < 09:00:00 PM = false
"AND" Branch           = false
Event ID = 593         = false
  
```

In all, the filter condition is false.

Now, the same event comes in at 02:00:00 PM:

```

Program start ID = 592   = true
  
```

```
Event ID = 592           = true
02:00:00 PM > 01:00:00 PM      = true
02:00:00 PM < 09:00:00 PM      = true
"AND" Branch             = true
Event ID = 593           = false
```

This time, the time frame is correct, yielding to an overall true condition from the "AND" branch. That in turn yields to the filter condition as whole to evaluate to true.

In this example still is another Event ID. All events with event ID 593 is grasped. This happens independently from the timing control when grasping the Events 592.

One last sample. At this time, event 593 comes in at 07:00:00 AM:

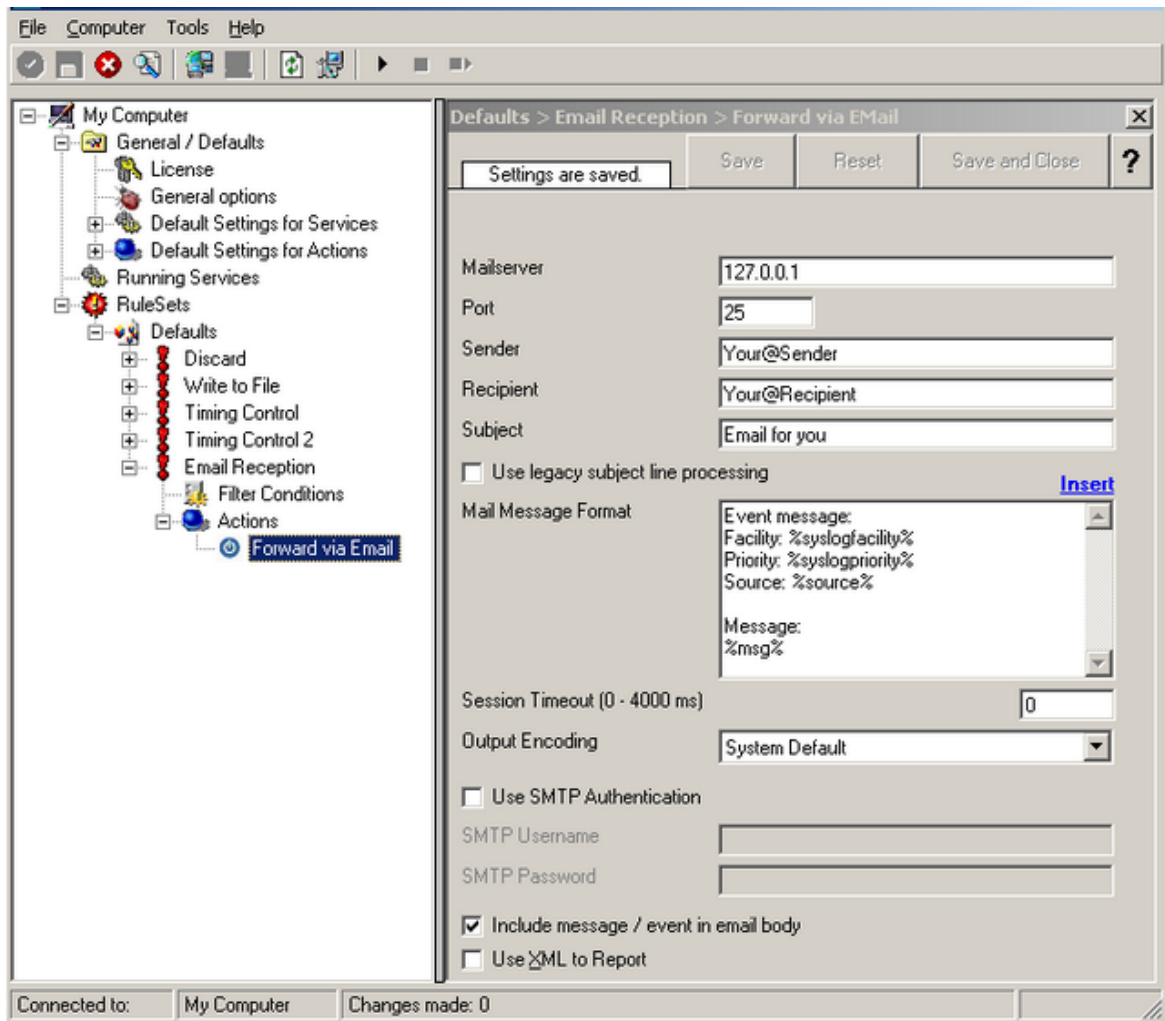
```
Program start ID = 592     = true
Event ID = 592           = false
07:00:00 AM > 01:00:00 PM      = false
07:00:00 AM < 09:00:00 PM      = false
"AND" Branch             = false
Event ID = 593           = true
```

This time the filter condition evaluates to true, too. The reason is that the (not matched) time frame is irrelevant as the other condition of the top-level "OR" branch evaluates to true (Event ID = 593).

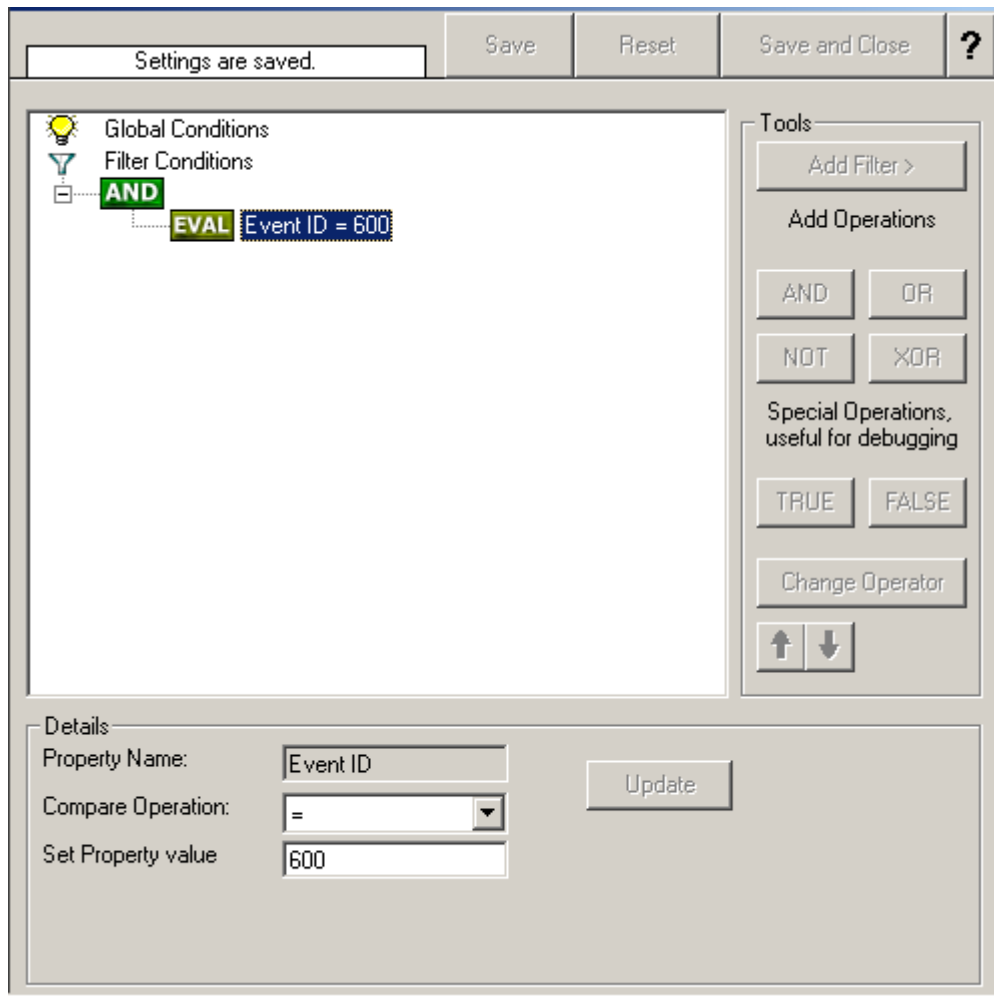
## 2.5.5 Email Notifications

In this example, we would like to receive email notifications when certain events happen.

So let us create an additional rule for that purpose: Right-click the "Defaults" rule set and select "Rule Sets", "Add Rule" from the pop up menu. Provide a name. We will call it "Email Reception" in this example. Then, add a "Forward via Email" action. In the action details, be sure to configure atleast the mail server, recipient and subject properties. Please note that many mail servers also need a valid sender mail address or otherwise will deny delivery of the message.



Then, select the filter conditions. Let us assume we are just interested in events of ID 600. Then the filter conditions should look as can be seen below:



When you have finished these steps, be sure to save the configuration and re-start the MonitorWare Agent service. After the restart, the newly extended rule set will be executed. In addition, the rules defined so far, the new one will be carried out, emailing all events with ID 600 to the specified recipient.

### 2.5.6 Alarming via Net Send

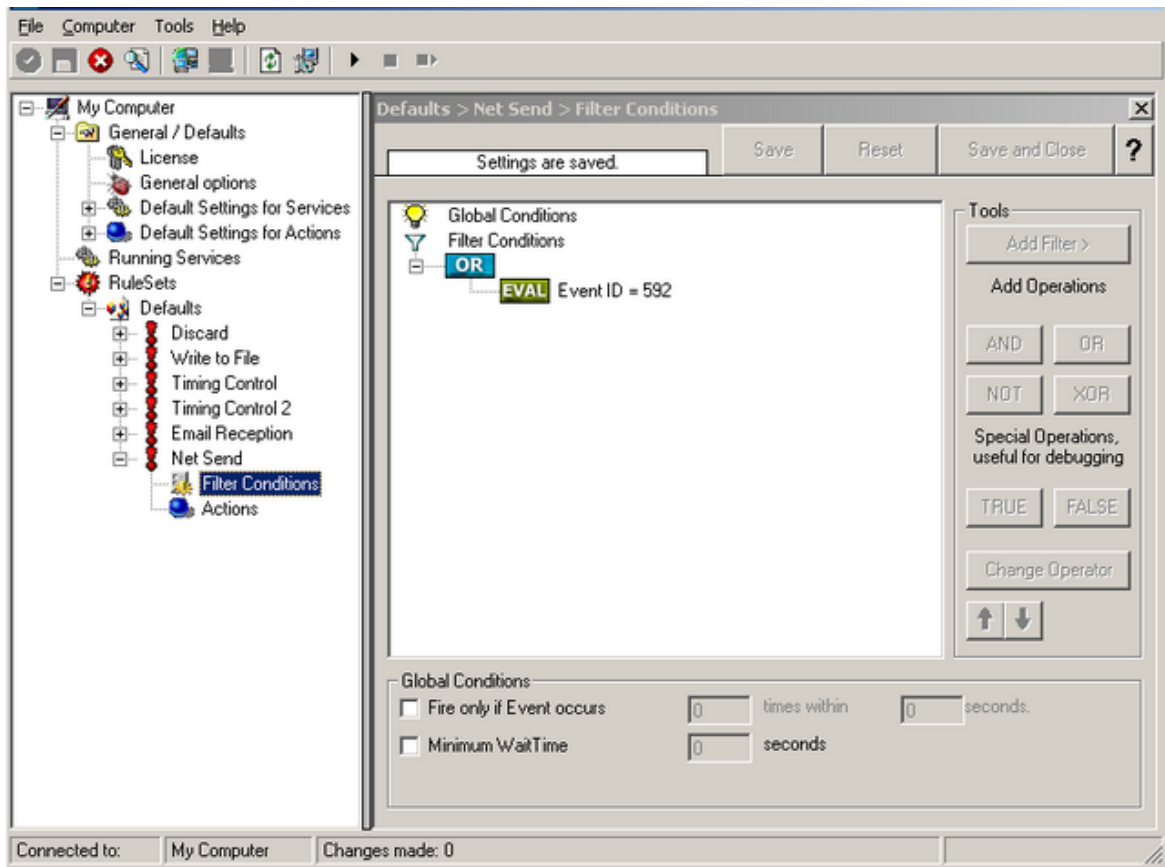
Again, we add another rule to our rule set. This time, we would like to receive notification via the Windows messenger service (aka "net send").

Please bear in mind that the Windows messenger service is not the instant messaging service that many people nowadays associate with it. The messenger service is meant for administrator notifications. If a windows workstation (or server) receives a message via that service, a message box pops up on that workstation and the user needs to press an "OK" button to continue. No interaction is possible.

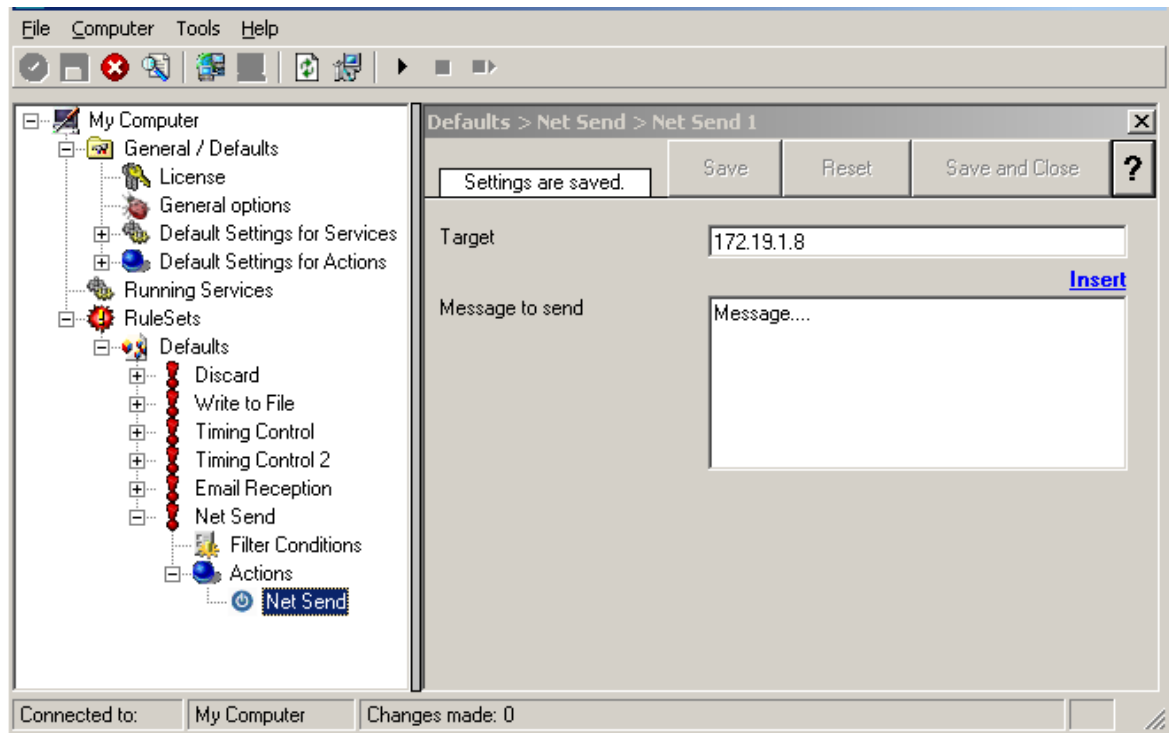
We create a new rule in our rule set "Defaults". In this case, we assume that we will receive messenger notifications for all events with event id 592. In a real use case, you will make sure that this is a real important event, or chances are good you will



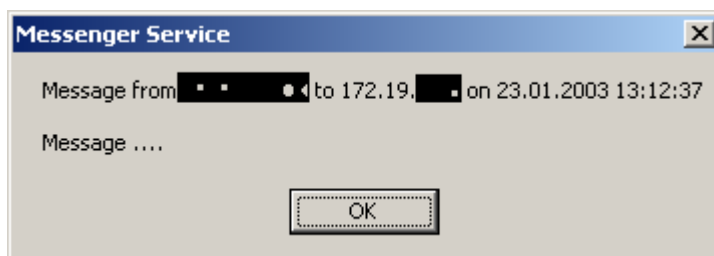
become overwhelmed with messaging windows. A better example could be a filter that checks for a server running low on disk space (using the disk space monitor).



This time, we use the "Net Send" action as can be seen below. The target field holds either the name or IP-Address of the workstation this message should be sending to. The message text itself goes into "Message to send".



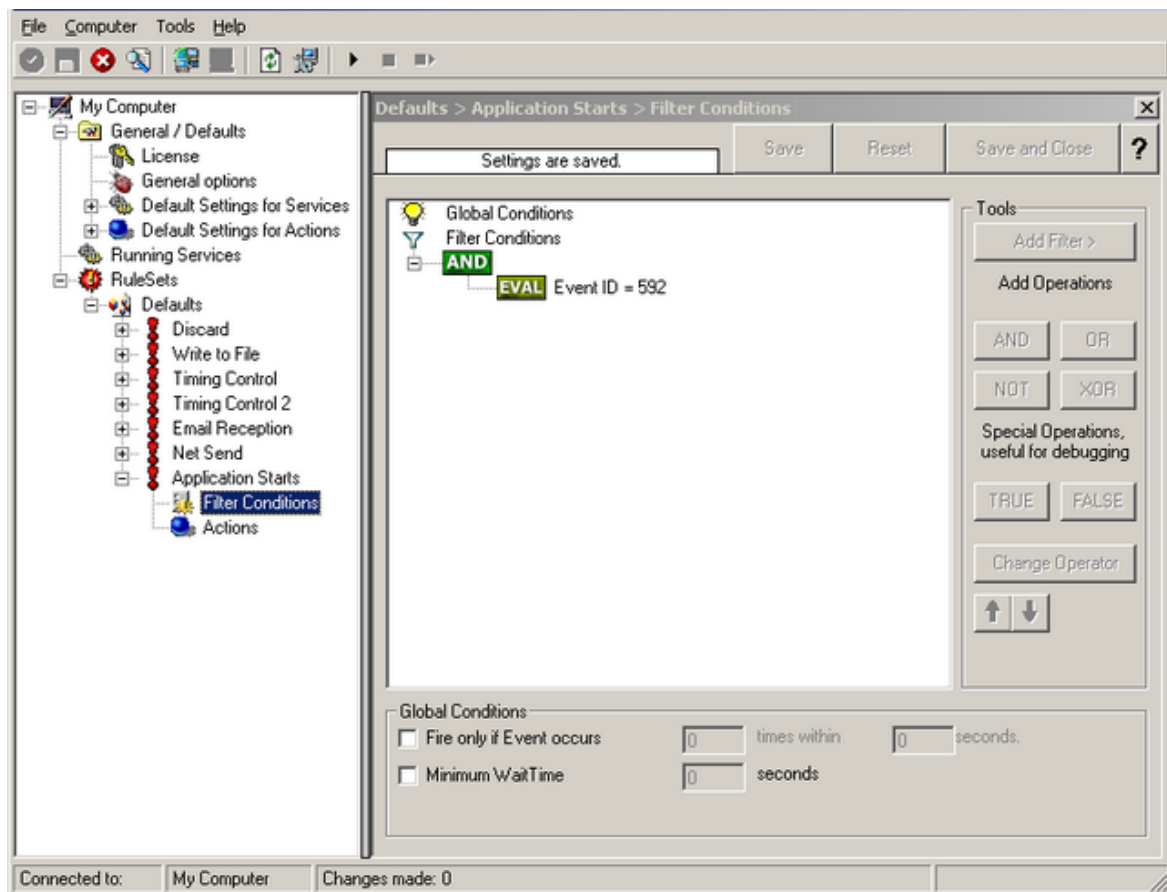
After saving the configuration and restarting the MonitorWare Agent, we will receive notifications if the filter condition evaluates to true. A sample message might look like this (slightly obscured in this sample):



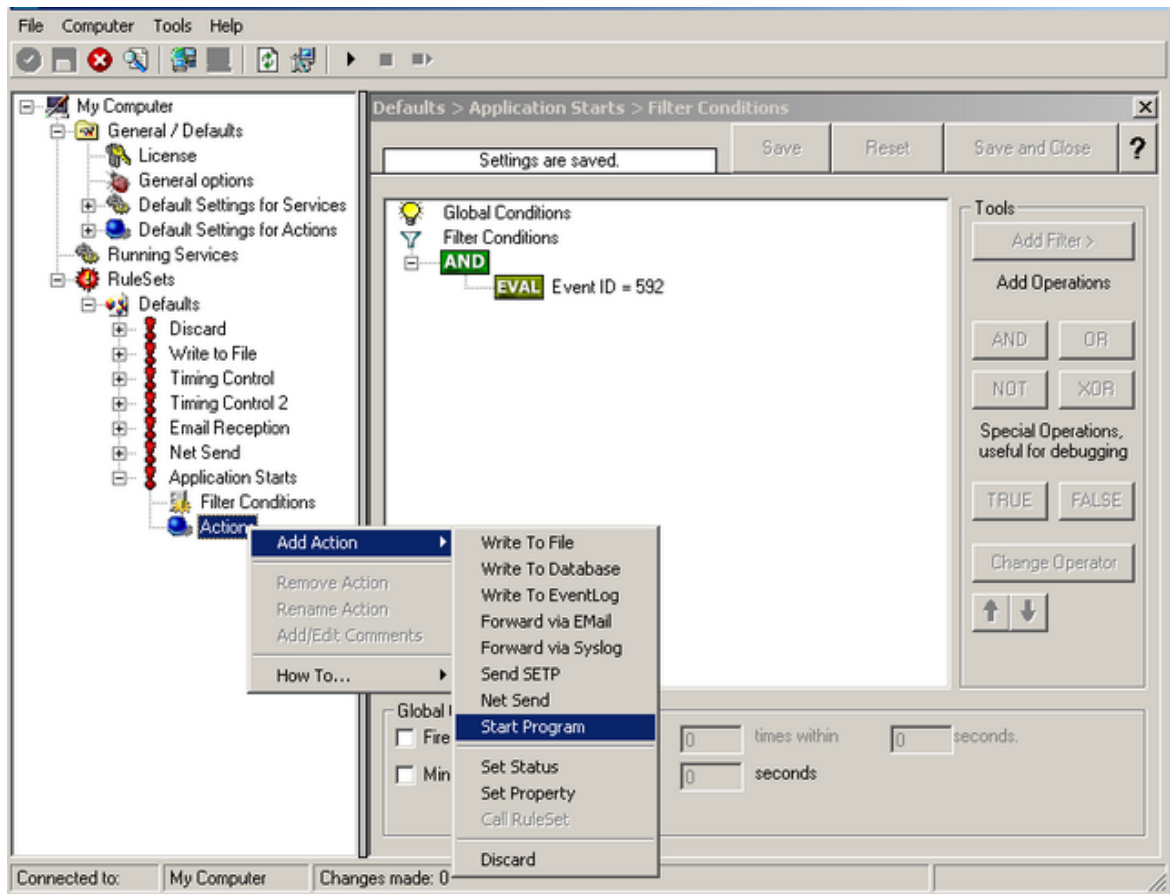
### 2.5.7 Starting Scripts and Applications in Response to an Event

We now want to start an application or a script when certain events occur. Typically, this is done to start administrative scripts or corrective action. For example, if a disk runs low on space, you could start a script that deletes temporary files, or if a service fails, a script could restart it.

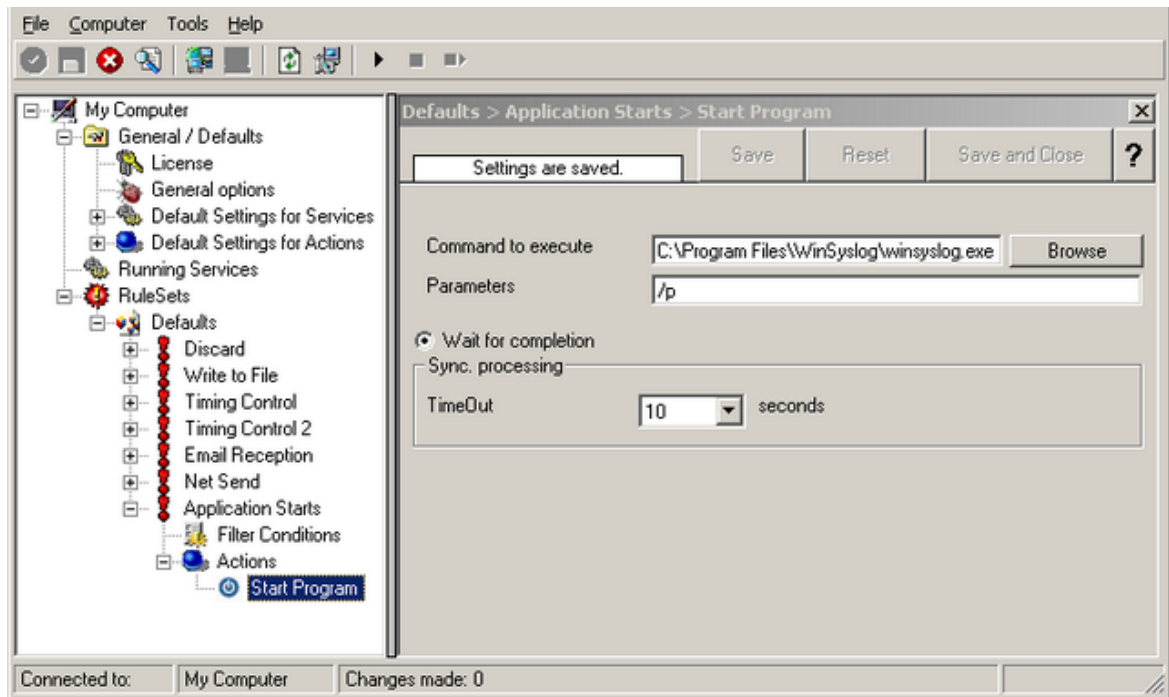
Our sample, on the other hand, is kept quite simple again. We just show how to generically start an exe file. To do so, we define a new rule, name "Application starts" below. Again, we use the imaginary event 592 as a filter condition. Therefore, the application will start whenever event 592 comes in.



The start program action is just a "normal" action:



In the "Start Program" action's parameters, select the file to run as well as all parameters to be supplied to it (if any):



Once this configuration is done, the program will be executed as soon as an event matching the filter condition comes in.

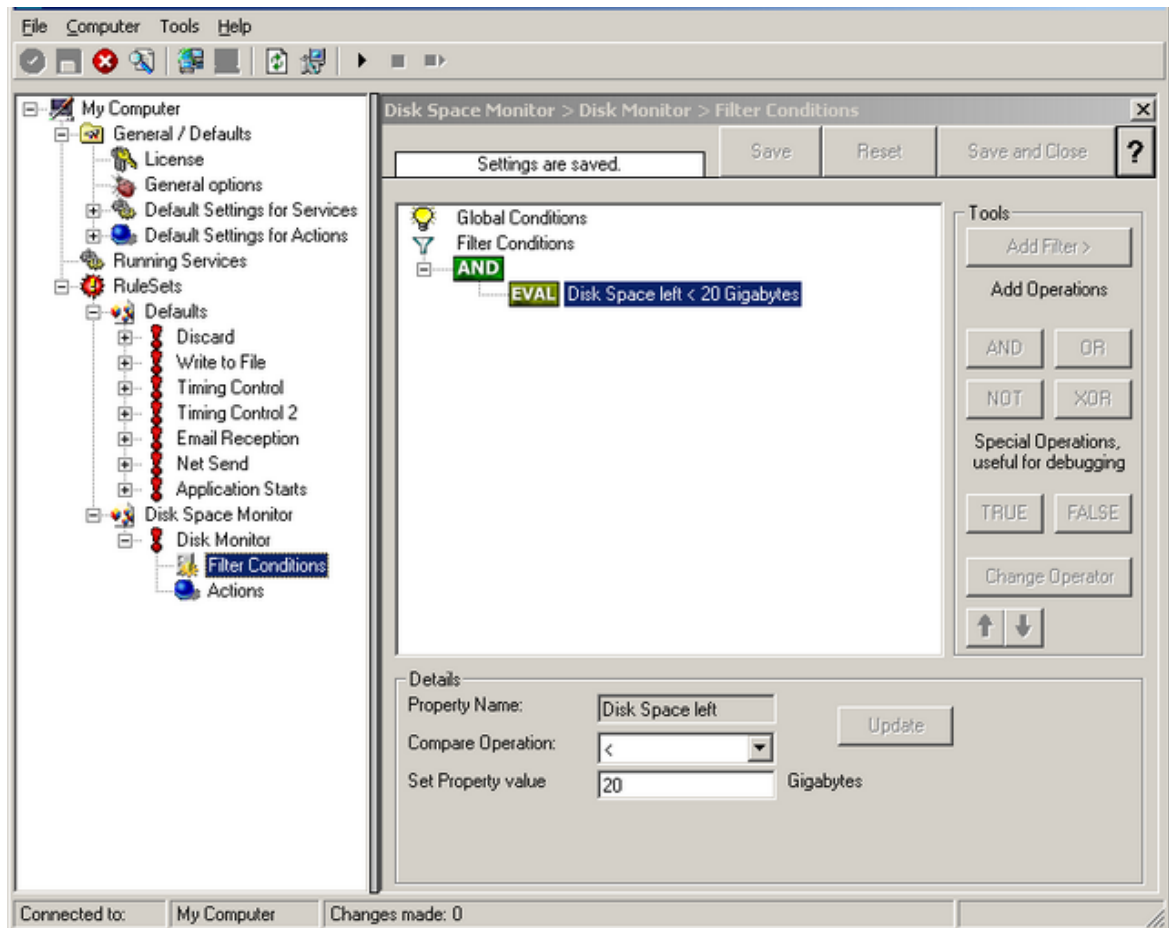
## 2.5.8 Monitoring Hard Disk Space

Monitoring hard disk space solves at least two purposes: it can be used to generate alerts or trigger corrective actions if a system runs out of free space. It can also be used as a statistical tool to monitor disk space utilization over time.

In our tutorial, we configure a simple disk space monitor and define a rule that stores the results into a text file that can later be analysed. Of course, we could have added trigger conditions for alerts and such. We have not done this, to keep things simple.

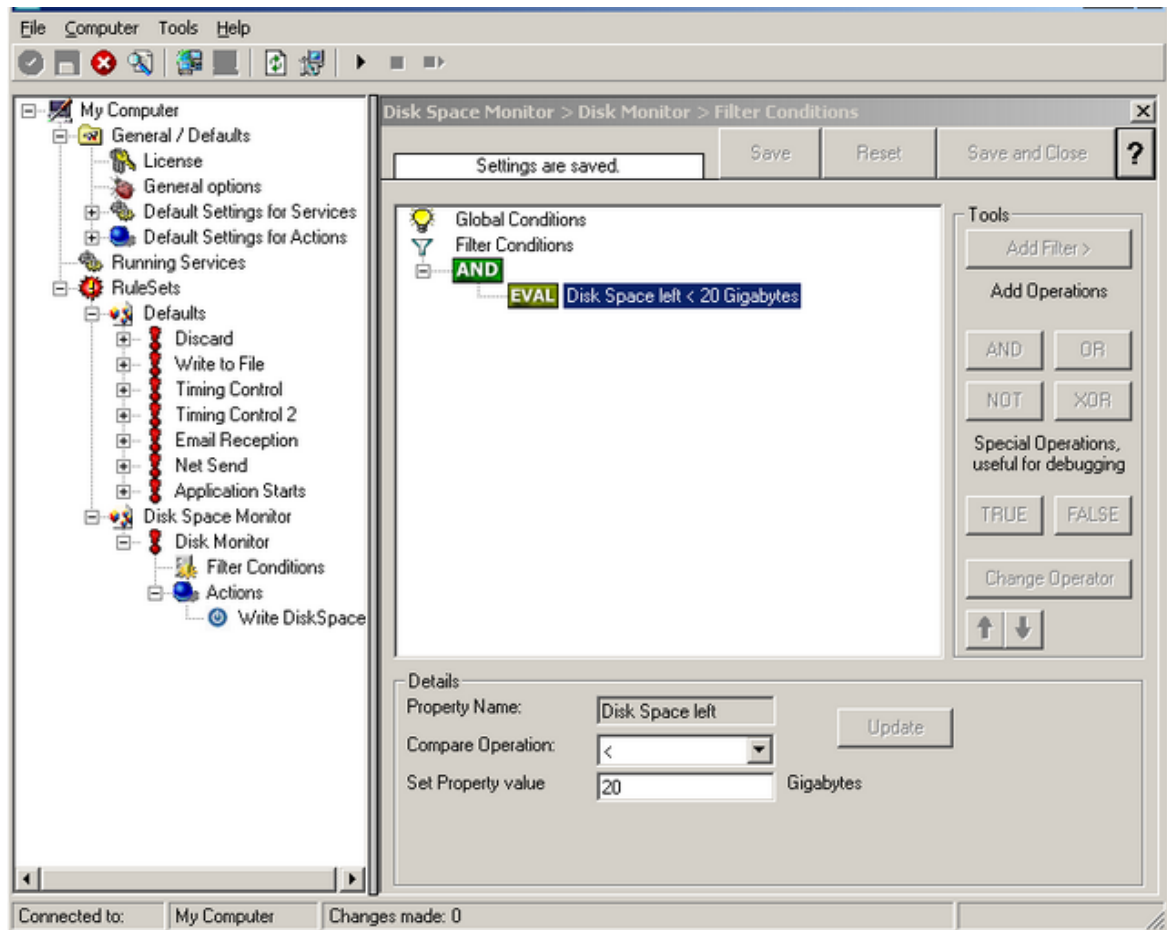
As always, we create the needed rule set first. In our sample, we call it "DiskSpace". Please note that this time we actually create a rule **set**, not just an additional rule in the "Defaults" rule set. The reason is that for our purposes it is much easier to define a specialised rule set and then bind this specialised rule set to the disk space monitor. If we would use the generic "Defaults" rule set, we had to make sure that our filter conditions would only match when an event of type disk space monitor would come in. In addition, it would require more processing time, as all rules and condition filters would be processed – a process that is not needed as we deal with a specific case. As such, it is more appropriate to define a specific rule set, which is then only used for the disk space monitor. What is appropriate in your environment depends on your needs. There is no general rule.

Inside the new rule, we create a filter condition that evaluates to true only if the report has less than 20 gigabytes of free space. So we will log date only when we potentially have constrained disk space. The filter looks as follows:



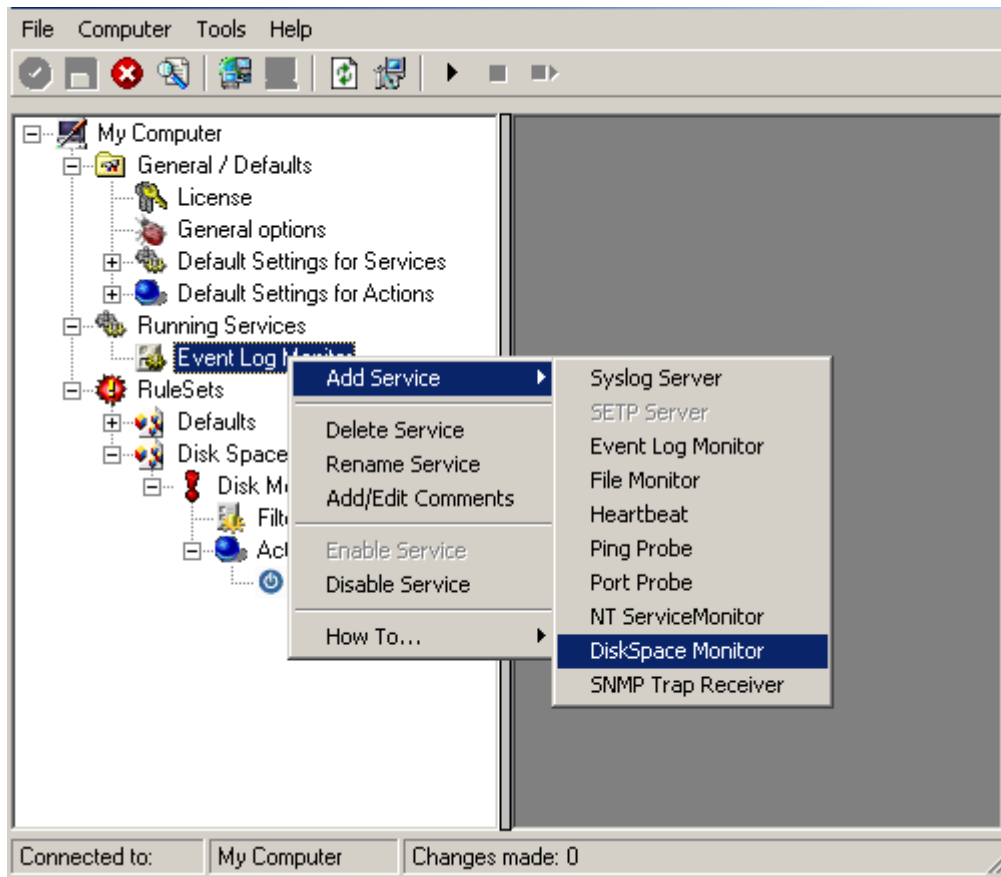
To create this filter, select "Disk Space Monitor", then "Disk Space Left" when pressing the "Add Filter" button.

As I said initially, we use the "Write to File" action in this sample. The action is called "Write DiskSpace" as can be seen below. We could also have used other actions, including emailing, to alert an administrator or start a script to delete temporary files.

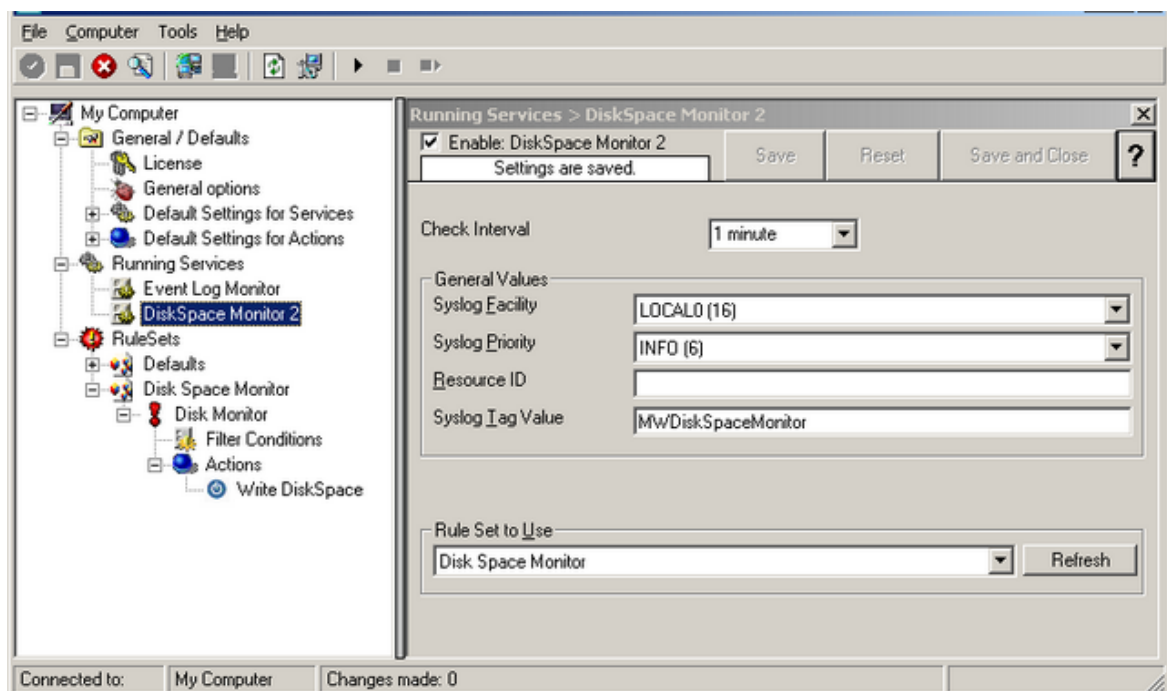


Please note: you should make sure that the base name is different from other "write to file" actions. Otherwise data might get mixed up in the files.

Having created the new rule set, we now need to create the disk space monitor service itself. It is the part of the software that actively goes out and monitors the disk space. To create it, right-click "Running Services" and select "Add Service", then "DiskSpace Monitor" as seen below:



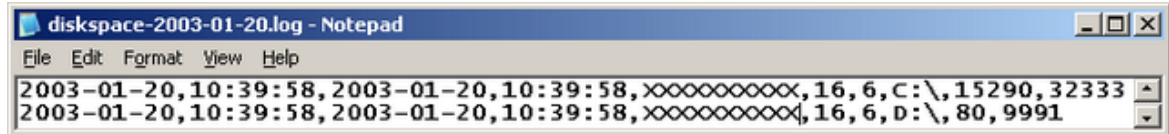
When the wizard starts, you need to name the new service. We use "DiskSpace Monitor" in our sample. Leave the default settings and click "Next" and "Finish".





When you select the new service, it is typically bound to the "Defaults" rule set (as seen above). We need to change this, as we have created the specific "Disk Space Monitor" rule set. Change the "Rule Set to Use" to update it to the new binding.

Save the configuration and restart the service. After a few moments, the disk space log file should fill up (**if there is less than 20 GB of free space on the monitored system**). In notepad, it looks like follows:

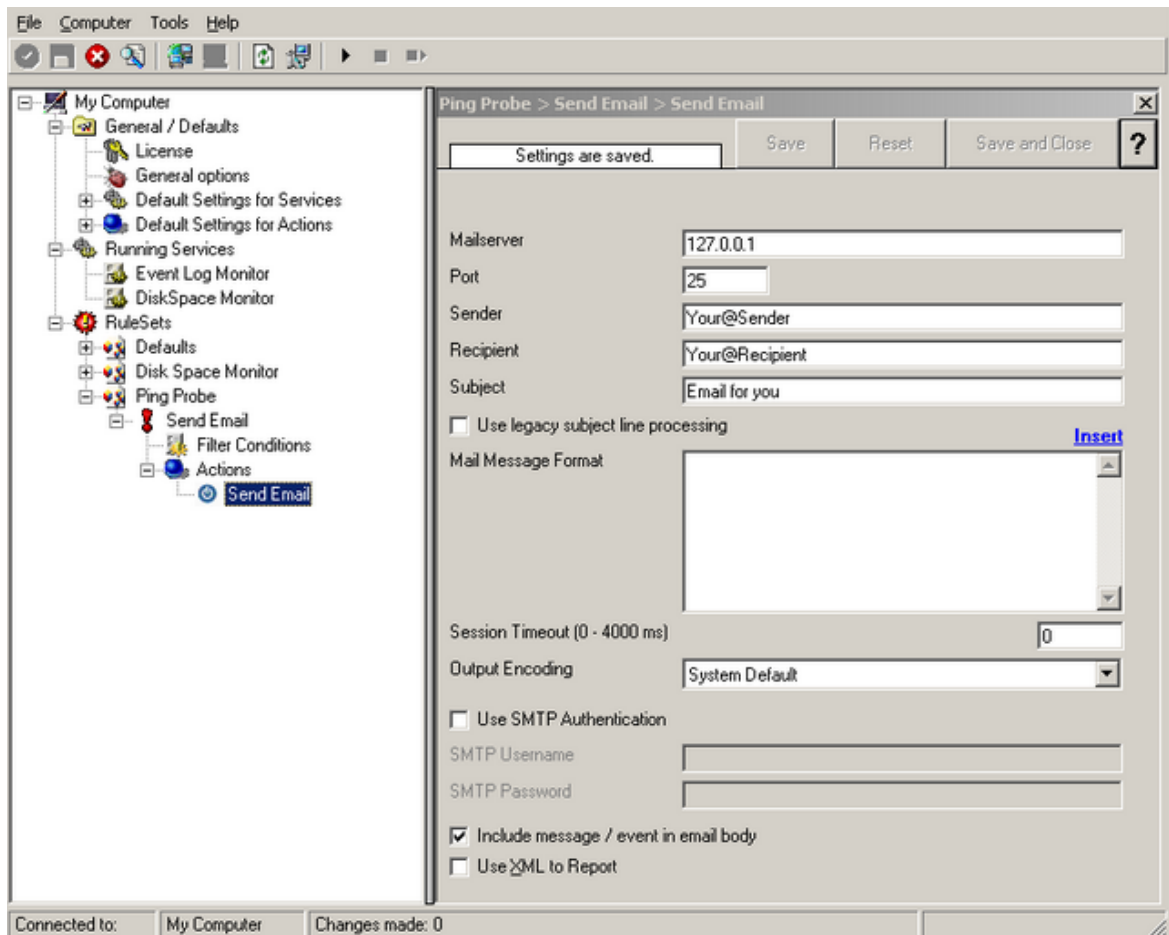


```
diskspace-2003-01-20.log - Notepad
File Edit Format View Help
2003-01-20,10:39:58,2003-01-20,10:39:58,XXXXXXXXXX,16,6,C:\,15290,32333
2003-01-20,10:39:58,2003-01-20,10:39:58,XXXXXXXXXX,16,6,D:\,80,9991
```

### 2.5.9 Monitoring external Devices via PING

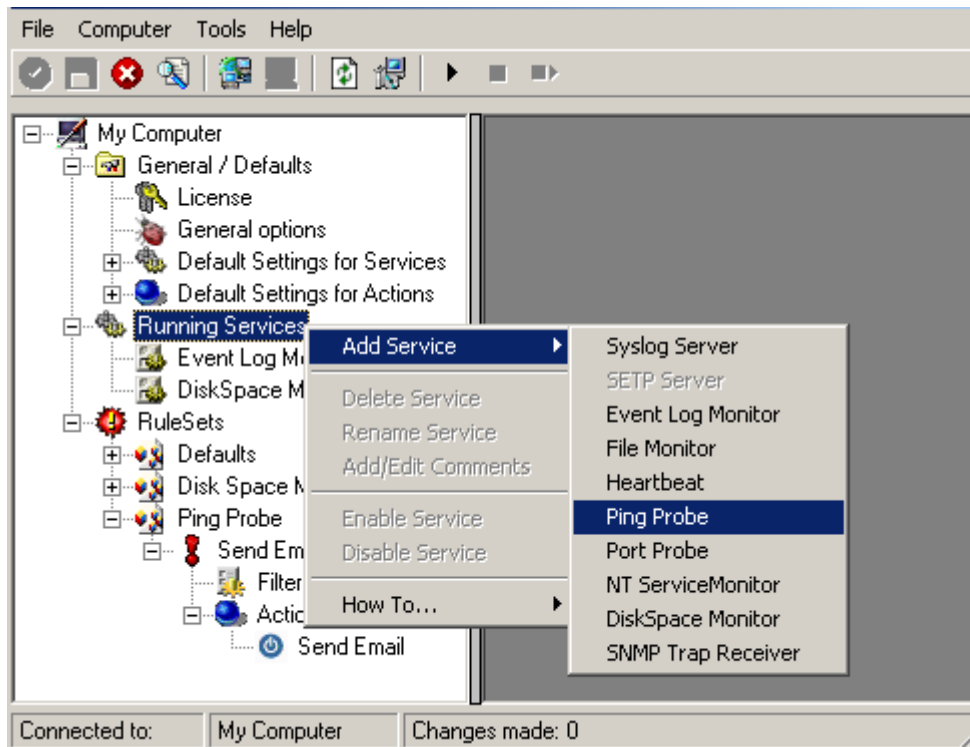
In this sample, we use the ping probe to monitor the availability of external devices. The ping probe issues a standard IP "PING". Each system that is "pinged" will provide a reply to the system initiating the ping. When the reply comes back, the initiator knows that the pinged system is up and running. Please note that this does not imply that all services on that machine are running. To check this, a port probe must be used. At least the ping probe can detect failing systems. It can also be used in any case, whereas the port probe can only be used with TCP based services.

As first step, we create a new rule set. Please see the previous example for the reasoning of doing so. We call the new rule set "Ping Probe". We would like to receive email notifications if the ping probe fails. So we add a "Send Email" action. After doing so, the screen looks as follows:



We do not customize the Send Email action properties in this sample. In your environment, you need to use some meaningful settings.

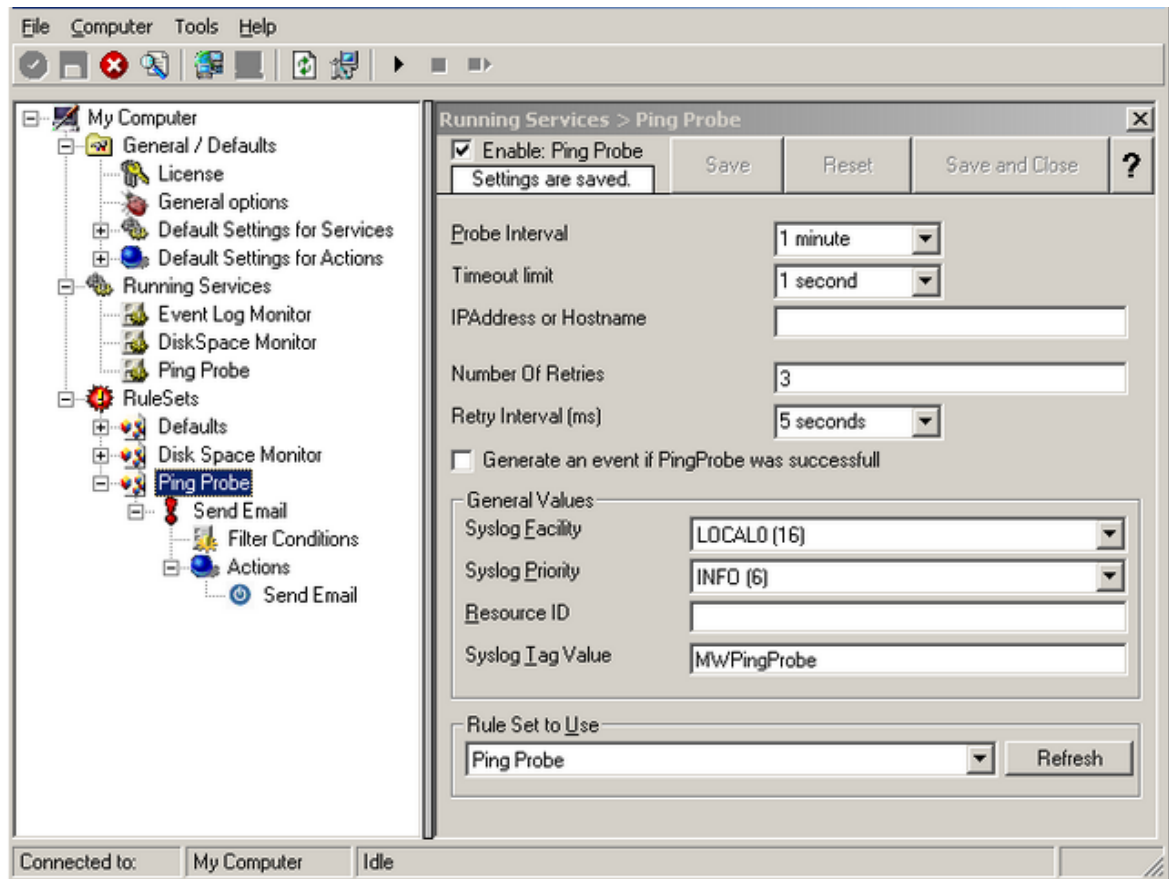
Now that we have defined the rule set, we need to create the corresponding service. To do so, right-click "Running Services" and follow the screen shot below:



Use a name of your choosing, leave the defaults as is and click "Next" and then "Finish". We have used the name "Ping Probe" in our sample.

Click the newly created service. We need to uncheck the "Generate an event if Ping Probe was successful" check box. If it is checked, an event is generated every time. If unchecked, it is generated only when the ping fails. As we are just interested in failed systems, we uncheck it. Therefore, we do not need to apply any other filters. If you forget to uncheck this option, you will receive multiple emails – one each time the ping probe runs and probes the configured system.

Your screen should now look as follows:



Now save the settings and restart the service.

Whenever the ping probe fails, you will receive mail. This mail looks as follows:

Event message:

Facility: 16

Priority: 6

Source: 172.19.1.18

Message:

PingProbe Status="error" remoteip="172.19.1.18" PingStatus="11003" ErrorMessage="Destination Host Unreachable"

A ping probe service can monitor a single device in this version of MonitorWare Agent. Therefore, if you would like to monitor multiple devices, you need to create multiple ping probe services.

## 2.5.10 Monitoring External Devices via a PortProbe

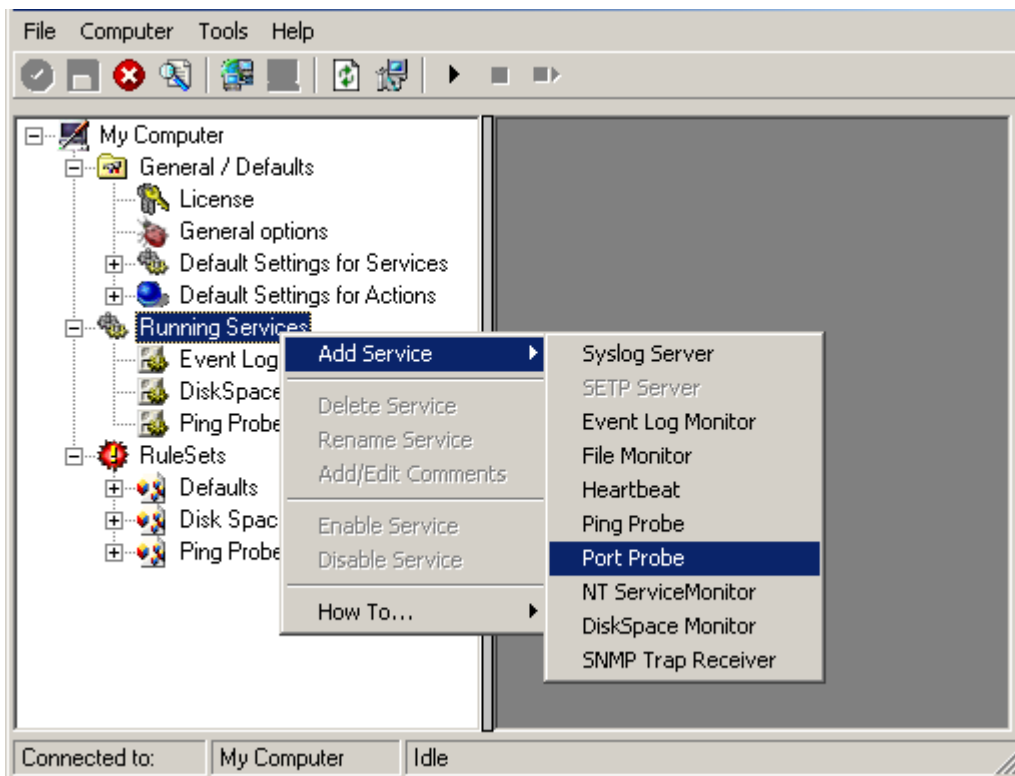
This sample is very similar to the ping probe sample directly above. Thus, we describe it briefly, only.

The main difference between the ping probe and the port probe is that the port probe tries to connect to a specific TCP port. As such, it can only be used with TCP based services like mail server, web servers or ftp servers. For the very same reason, the port probe does not only check the status of the machine it is connecting to but rather if a specific, **service** is available. Let us assume you are interested in monitoring a mail server. If you do a ping probe, the mail server itself might have died while the machine is still running. The ping probe cannot detect this. The port probe, on the other hand, directly connects to the mail server, e.g. on port 25 (the default SMTP port). If the mail server has died, it will probably not answer this connection request and thus the port probe is able to detect the failing state of the service.

In our sample, we probe a web server, which typically listens to port 80 (the default port for http). We will send an alert email if the port probe cannot connect successfully to the web server.

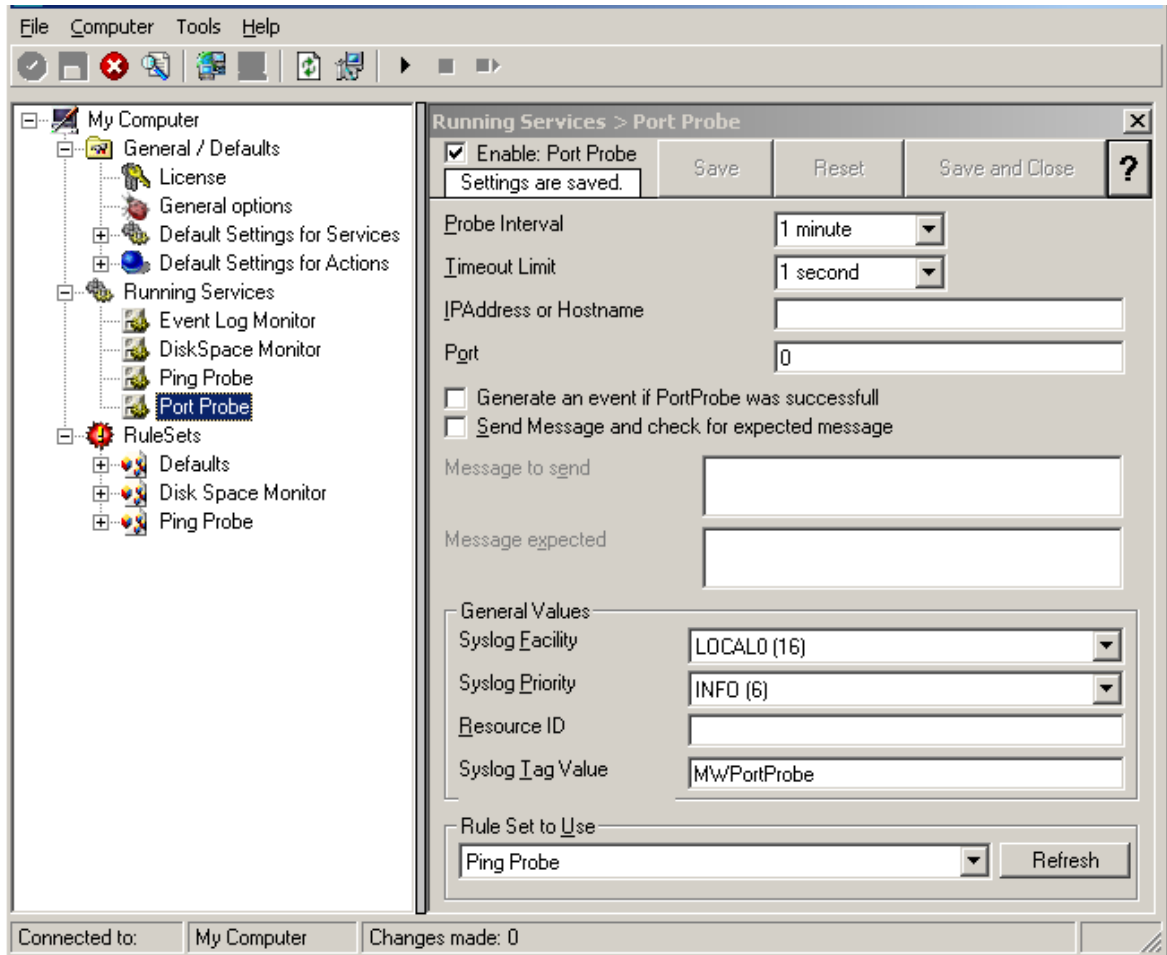
Because this sample is so close to the previous one, we do not create a new rule set specifically for email alerting. It is already covered in the "Ping Probe". This is a good sample of rule set reuse. It might be clever to rename the rule set in such a case. For simplicity reasons, we have not done this here.

Therefore, we begin by creating the new service, done by right-clicking "Running Services":



Use a name of your choosing, leave the defaults as is and click "Next" and then "finish". We have used the name "Port Probe" in our sample.

After doing so, select the newly created service in the tree view to look at its properties. Be sure to bind it to the "Ping Probe" rule set as seen below:



Save the configuration and restart the service. From now on, the following mail alert will be generated when the port cannot be connected to:

Event message:  
 Facility: 16  
 Priority: 6  
 Source: 192.168.1.1

Message:  
 PortProbe status="fail" target="192.168.1.1" port="80" netstate="10065" message="Couldn't connect to host"

## 3 Common Uses

MonitorWare Agent can be used in a multitude of ways to perform well in many different environments serving many different needs. We have set up some web pages to address these questions. This allows us to add timely comments, should need arise. Please follow the links below to access the web pages with detailed descriptions.

In general, there are four main use cases for MonitorWare Agent:

- [Analysis](#)
- [Event Archival](#)
- [Alerting](#)
- [Solving Problems](#)

Besides this main cases, there are also some other scenarios, like [relaying event data](#).

While reading the scenarios, please keep in mind that MonitorWare Agent is extremely flexible. A single instance on a single machine can be configured to perform all actions and functions concurrently. They are grouped here for easier lookup, but this in no way implies that the Agent can do only one thing or the other.

## 4 Step-by-Step Guides

The step-by-step guides are meant to get you started quickly. They provide information on how to configure the product in common scenarios. Each section includes the information necessary to complete a specific task.

The information is presented in an easy to follow "step by step" way (hence the name). Each section begins with the intended result and then explains the steps to achieve it in the correct order. They are documented together with hardcopies, so they should be easy to follow. For best results, please be sure to follow the exact order of the steps.

The step-by-step guides do eventually not include all information that might be relevant to the situation. Please use your own judgment if the scenario described sufficiently matches your need.

In the step-by-step guides, we assume the product is already successfully installed but no configuration has been done. If it is not installed, please do so first.

All step-by-step guides assume that the client is running. This is kind of a step 0 for all the guides.

To keep download times reasonable, the step-by-step guides are not included in this manual. They are kept as separate web pages. This also allows us to modify and add step-by-step guides. Additions are made all the time, so it is probably a good idea to check <http://www.monitorware.com/Common/en/stepbystep/> for new guides.

As of this writing, the following step-by-step guides were available:

- [How to Configure a Syslog Server Service](#)

- [How to setup SETP Server Service](#)
- [How to setup EventLogMonitor Service](#)
- [How to setup a SETP Action](#)
- [Creating a simple Syslog server](#)
- [Forwarding NT event logs to a Syslog server](#)
- [Forwarding NT event logs to an SETP server](#)
- [Creating a rule set for database logging](#)
- [Centralized event reports with Monilog](#)
- [Intrusion detection via the Windows event log](#)
- [Sample Syslog device configurations](#)
- [Firewall setup for MonitorWare Agent](#)
- [Configuring Windows for the Event Log Monitor](#)
- [Creating a hardened log host](#)
- [Optimizing Syslog Server Performance](#)
- [How To Report Log Truncation](#)
- [How To setup PIX centralized Monitoring](#)
- [Centralized Event Reports with MonitorWare Console](#)

You may also want to visit our syslog device configuration pages at <http://www.monitorware.com/en/syslog-enabled-products/>. They contain instructions on setting up several devices for syslog.

## 5 Using Interactive Syslog Server

Interactive Syslog server is an add-on to the MonitorWare Agent. **Please note that it is a utility program, with a primary focus on real-time troubleshooting.**

Interactive Syslog Server is **not** meant to continuously monitor a system. This is what the service is designed for. While Interactive Server allows to view current syslog traffic, the service should be used for all other purposes, like creating log files.

### 5.1 Launching the Interactive Syslog Server

To run the Interactive Syslog Server, click the "Interactive Syslog Server" icon present in the program folder located in the Start menu.

It can also be launched from the command prompt:

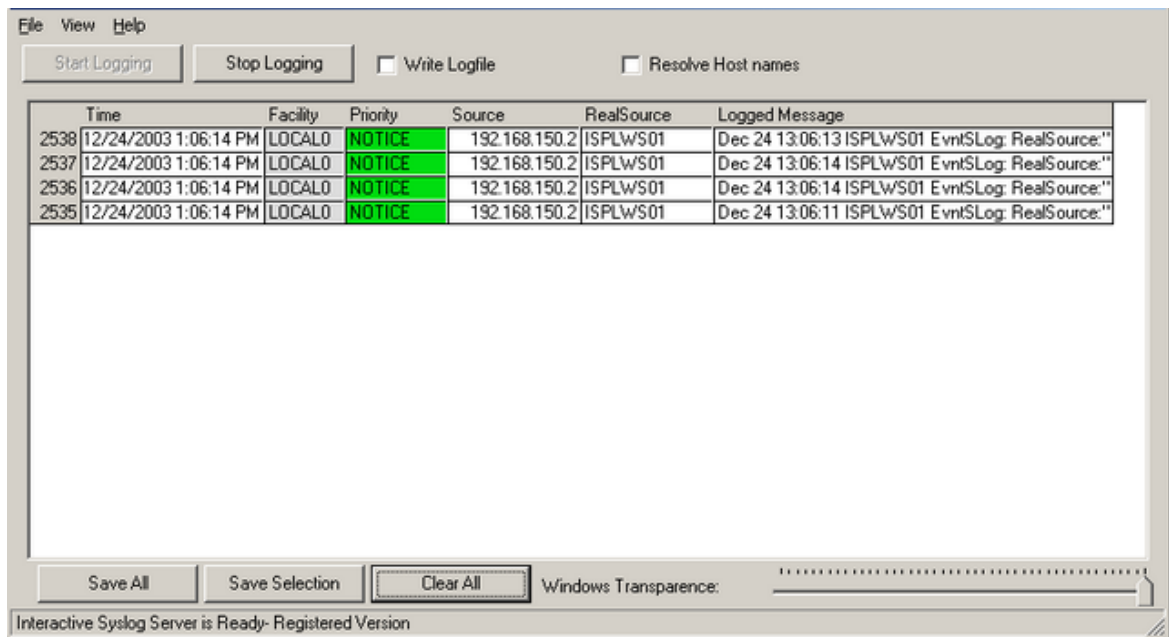
- Open a Command Prompt window
- Change to the drive and directory where the MonitorWare Agent is installed.
- Type "InteractiveSyslogServer.exe" and hit enter.



## 5.2 The Interactive Logging

Interactive Logging enables the client to log Syslog messages itself. Therefore, it can work without the service. However, by default the service is required to run and needs to be configured to forward Syslog messages to port 10514 via UDP . This is done to prevent conflicts between the interactive server and the background service. If you do not have a good reason to do so, we strongly recommend using this default setup.

Interactive Syslog is also supported under Windows 9x and Windows Me systems. The service does not work on these platforms.



### Start / Stop Logging Buttons

These buttons start and stop Interactive logging. Once started, the client will log all incoming messages until logging is stopped by the user. Messages are written to a circular buffer. That means if the maximum buffer size is reached, new messages will be stored, but older messages will be removed from the buffer. This allows the client to run for extended periods without taking up too much system memory. The buffer size is configurable. New messages are always displayed on top of the list. Older ones are towards the bottom.

### Write Logfile

If checked, all messages are written to a log file in addition to the interactive display. Please note that this option influences the client only. If you would like to provide a reliable long term log, we strongly suggest to use the service. Its file logging parameters are customized under the "file tab".

## **Resolve Host Names**

If checked, the sender is displayed as a host name instead of the IP address. This is often useful to quickly see the system that sent the message. Please keep in mind, though, that the host name resolution takes a little bit of time (especially if a host can not be resolved) and as such should not be used on a loaded system.

## **Save All**

Used to save the current buffer contents to a comma-delimited file (so called CSV format). All entries displayed in the grid are written.

## **Save Selection**

Also saves a comma-delimited file. However, only messages selected (highlighted) will be written to the file.

## **Clear All**

Erases all messages from real-time display.

## **Windows Transparency**

When transparency functionality (slider) is moved to either right or left, Interactive Syslog Server becomes transparent.

## 5.3 Interactive Syslog Server Options

This screenshot shows you the available options in the Interactive Server.

The screenshot shows a dialog box titled "You can set all options for the Interactive display". It contains several sections:

- Interactive Display**:
  - Message Buffersize: 60
  - Interactive Syslog Port: 10514
  - Resolve RealSource from Syslog message if available
- File Basename**: A "Browse" button and a text field containing "Syslog".
- File Extension**: A text field containing "txt".
- Create unique filenames
- Choose your language**: A list box with "English" selected, and other options: "Deutsch", "Français", and "Japanese".

At the bottom, there are "Ok" and "Cancel" buttons.

### Message Buffersize

The message buffer size (in number of messages) to be used for real-time display. This is the maximum number of messages to be stored in memory. If this number is reached and a new message arrives, the oldest one is deleted from memory.

### Interactive Syslog Port

The UDP port the real-time display listens to. Zero is default from system services database. Most installations can leave it at 10514.

### Resolve RealSource

If enabled then it would resolve the IP address and display the real source name e.g. any workstation, firewall, router etc. For example, the IP address of the source (named as Test01) was 192.16.1.0 after enabling this option it would display Test01

instead of the IP address.

## File Basename

The File Basename also includes the file path. An example could be "C:\temp\MWAgent".

## File Extension

The File Extension is "txt" by default. This will open the files automatically in the default text viewer. .

## Create unique filenames

If enabled, the Interactive Server will build a unique filename each day containing the year, month and day. An example would be "Syslog-2002-01-01.txt".1

## Language

The Interactive Syslog Server is multilingual by design. Select the user interface language here.

Languages are set on a per user basis. They can be switched instantly without the need to restart.

Additional languages might be available. Please check [www.mwagent.com](http://www.mwagent.com) from time to time. If you are interested in other languages and volunteer to provide translation services, please email [info@adiscon.com](mailto:info@adiscon.com) . We will gladly help.

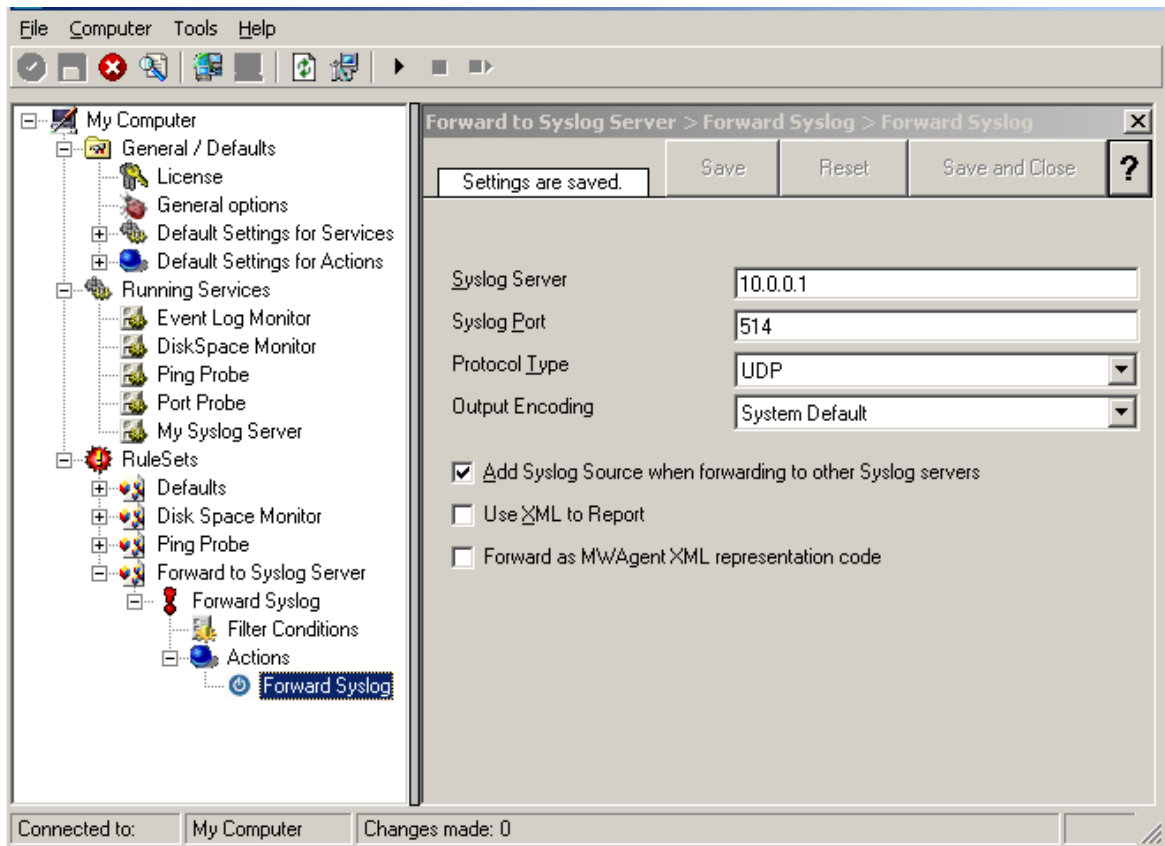
# 6 Configuring MonitorWare Agent

*MonitorWare Agent is easy to use and is powerful.*

In this chapter, you will learn how to configure the MonitorWare Agent Service.

The MonitorWare Agent service runs in the background once it is configured. There is no manual intervention needed to operate it. As such, this chapter focuses on the MonitorWare Agent configuration client application. It is used to configure the service settings.

To run the MonitorWare Agent Configuration client, simply click its icon present in the MonitorWare Agent program folder located in the Start menu. Once started, a Window similar to the following one appears:



### *MonitorWare Agent Configuration Client*

The configuration client ("the client") has two elements. On the left hand side is a tree view that allows you to select the various elements of the MonitorWare Agent system. On the right hand side are parameters specific to the element selected in the tree view. In the sample above, the right hand side displays the specific parameters for a rule action.

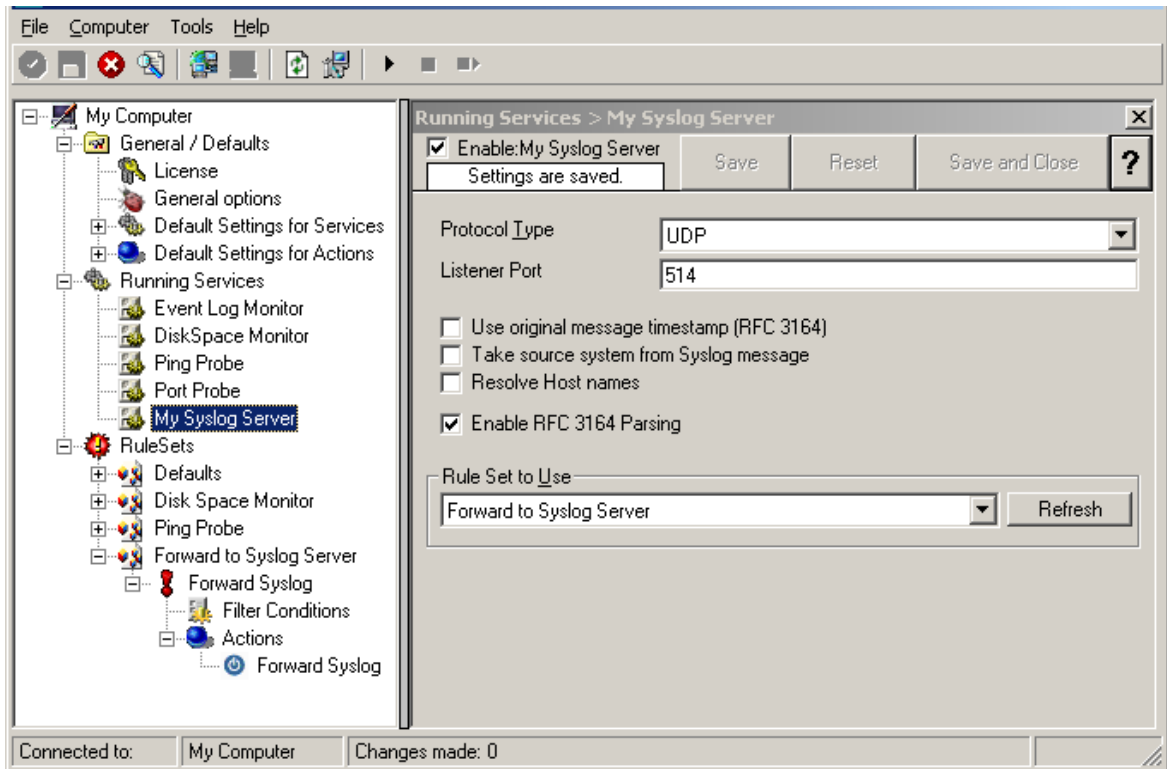
The tree view has three top-level elements: **General**, **Running Services** and **Rules**.

Under **General**, basic operational parameters as well as defaults for actions and services are defined. The default themselves do not activate anything. However, the parameters in here are used each time an actual service or action needs a configuration parameter and none is defined in that specific instance. We highly recommend putting the most common parameters into the defaults. That will reduce the amount of data entry in the specific elements dramatically. Please note that each default can be overwritten in a specific service or action.

The tree view's **Running Services** area lists all configured services as well as their parameters. There is exactly one service entry for each service created. Please note that there can be as many instances of a specific service type as your application

requires. In the above example, there are two instances of the Syslog Server, each one listening to a separate port. Theoretically, you can run a few hundred services in a single service instance. However, both from a usage scenario point of view as well as concerning operating system resources, we recommend limiting the services to a maximum of 20 to 30. Of course, there are some applications where more than this limit is useful. MonitorWare Agent does not restrict this number. If there is a need for a large number of services and the hardware is capable of managing all these tasks, there is nothing in the MonitorWare Agent that limits from doing so.

The service definition looks like this:



#### MonitorWare Agent Configuration Client - Service Definition View

The actual parameters depend on the service type. Common to all services is the capability to enable or disable a service. A service is started only if it is enabled. Otherwise, it will be not run, but the configuration data can still be present. That way, it is easy to temporarily disable a service without deleting it.

Also common to all service types is the association to a rule set seen at the bottom of the right hand configuration dialog. This specifies which of the rule sets will be applied to information units generated by this service.

To create a new service, right click on "Running Services". Then select "Add Service" and the respective service type from the pop up menu. Then follow the wizard. To delete an existing service, right click it and select "Delete Service". This will remove the service and its configuration irrecoverable. To temporarily "remove" a service, simply disable it in the property sheet.

The tree view's last main element is **Rules**. Here, all rule sets are configured. Directly beneath "Rules" are the individual rule sets. Each set is completely independent from each other. They are just centrally stored so they can be associated with services (see above for an explanation).

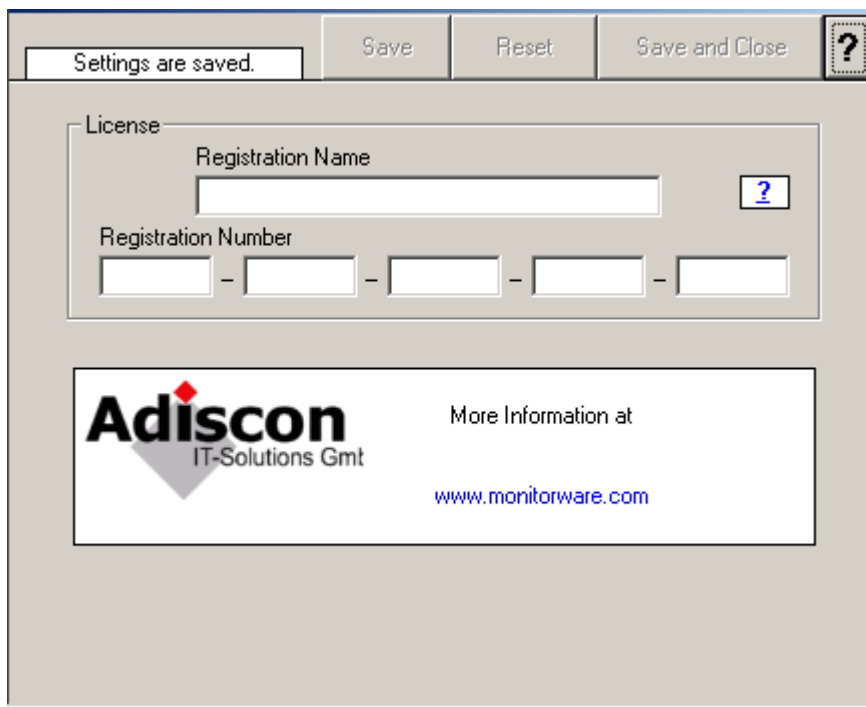
Beneath each rule set are the individual rules. As described in [Rules](#), a rule's position in the list is vitally important. Rules at the top of the rule set are executed before those further down. To move a rule up or down, simply right click it and select "move up" or "move down" from the pop up menu.

In the tree view, filter conditions and actions are beneath the rule they are associated with. Finally, beneath actions are all actions to carry out.

The following sections describe each element's properties.

## 6.1 License Options

This tab can be used to enter the MonitorWare Agent license after purchase.



The screenshot shows a dialog box titled "License Options". At the top, there is a status bar with the text "Settings are saved." and three buttons: "Save", "Reset", and "Save and Close". A help icon (?) is also present. The main area is divided into two sections. The first section, labeled "License", contains a "Registration Name" text box with a help icon (?) to its right, and a "Registration Number" field consisting of five separate boxes separated by hyphens. The second section features the Adiscon logo (IT-Solutions GmbH) on the left and the text "More Information at" followed by the URL [www.monitorware.com](http://www.monitorware.com) on the right.

*License Option Parameters*

### Registration Name

The user chooses the registration name. It should correspond to your organization name, e.g. a company called "AA Carpenters, Inc." should not choose "AA" as

registration name. This can easily be mistaken and most probably will be rejected by Adiscon for that reason. With the above scenario, we recommend using the full company name "AA Carpenters, Inc."

**Please note:** the registration name is case sensitive. It must be entered exactly as given. Leading and trailing spaces are also part of the registration name, so be sure to enter none.

## Registration Number

Adiscon provides this number. It is valid for a specific registration name. Be sure to enter the correct registration number. The client will detect invalid registration numbers and report and corresponding error.

## 6.2 Debug Options

This tab can be used to debug rule bases. Especially with complex bases, it might be necessary to learn what MonitorWare Agent is internally doing while it is processing them. With the debug log, the service will tell you some of these internal workings.

Other than rule basis testing, the debug log is also helpful when contacting Adiscon support. An Adiscon support engineer might ask you to set the debug log to a specific level while doing troubleshooting.

**Important:** Debug logging requires considerable system resources. The higher the log level, the more resources are needed. However, even the lowest level considerable slows down the service. As such, **we highly recommend turning debug logging off for normal operations.**



The screenshot shows a configuration window for the MonitorWare Agent. At the top, there are buttons for 'Save', 'Reset', and 'Save and Close', along with a status indicator 'Settings are saved.' and a help icon. The window is divided into two sections: 'General Options' and 'Debug Options'. In the 'General Options' section, there are three fields: 'ProcessPriority' set to 'Normal', 'CustomerID' set to '0', and 'SystemID' set to '0'. In the 'Debug Options' section, there is a checkbox for 'Enable Debug output into file' which is unchecked. Below it is a text field for 'File and path name' containing 'C:\MWDebug3.txt' and a 'Browse' button. The 'Debug Level' is set to 'Full Debugoutput' in a dropdown menu.

*Debug Options Parameters*

## Process Priority

Configurable Process Priority to fine-tune MonitorWare Agent behavior.

## CustomerID

CustomerID is of type integer provided for customer ease. For example if someone monitors his customer's server, he can put in different CustomerIDs into each of the agents. Let us say someone monitors servers A and B. A has 5 servers all of them with CustomerID = 1 and B has 2 servers all of them with CustomerID = 2. Both A and B happen to have a server named "SERVER". Together with the customerID, these machines are now uniquely identifiable. This is user configurable.

## SystemID

SystemID is of type integer to be used by our customer. In addition, it is user configurable.

## Enable Debug output into file

If checked, the debug log is enabled and written as the service operates. If unchecked, no debug log is written. For performance reasons, it is highly recommended that this box is unchecked during normal operations.

## File and path name

The full name of the log files to be written. Please be sure to specify a full path name **including** the driver letter.

If just the file and/or path name is specified, that information is local to the service default directory. As this depends on a number of parameters, it might be hard to find the actual log file. So for consistency purposes, be sure to specify a fully qualified file name including the drive

## Debug Level

This controls the amount of debug information being written. We highly recommend only selecting "Minimum Debug Output" unless otherwise instructed by Adiscon support.

## 6.3 Services

### 6.3.1 Understanding Services

Services gather event data. For example, the syslog server service accepts incoming syslog messages and the Windows event log monitor extracts Windows event log data. There can be unlimited multiple services. Depending on the service type, there can also be multiple instances running, each one with different settings.

You must define at least one service, otherwise the product does not gather event data and hence does not perform any useful work at all. Sometimes, services are mistaken with service defaults, that are pre-existing in the tree view. Service defaults are just the templates that carry the default properties assigned to a service, when one of the respective type is to be created. Service defaults are NOT executed and thus can not gather any data.

## 6.3.2 Syslog Server

Configures a Syslog server service.

Enable: My Syslog Server  
Settings are saved. Save Reset Save and Close ?

Protocol Type: UDP

Listener Port: 514

Use original message timestamp (RFC 3164)  
 Take source system from Syslog message  
 Resolve Host names  
 Enable RFC 3164 Parsing

Rule Set to Use: Forward to Syslog Server Refresh

### Protocol Type

Syslog messages can be received via UDP , TCP or RFC 3195 RAW. One listener can only listen to one of the protocols. Typically, Syslog messages are received via UDP protocol, which is the default. The syslog server also can receive Syslog messages via TCP and reliable Syslog messages via TCP using the new [RFC 3195 RAW](#) standard.

### Listener Port

The port the Syslog server listens on. The typical (standard) value is 514. This should be changed only if there is a definite need for it. Such a need typically arises from security concerns. If the port is changed, all reporting devices (routers, printers ...) must also be configured to use the non-standard port.

### Use Original Message Timestamp

If this box is checked, the timestamp is retrieved from the Syslog message itself (according to RFC 3164). If left unchecked, the timestamp is generated based on the local system time. The Syslog message timestamp does not contain time zone information. Thus, we strongly recommend unchecking this box if messages from devices in multiple time zones are to be received.

## Take source system from Syslog message

If this box is checked, the name or IP address of the source system is retrieved from the Syslog message itself (according to RFC 3164). If left unchecked, it is generated based on the address, the message was received from.

**Please note that there are many devices, which do NOT generate RFC 3164 compliant messages. If you check this option here, you might see a very strange value as the event source!**

## Resolve Hostnames

If this box is checked, the name of the source system is retrieved via DNS reverse name resolution. If unchecked, the IP address itself is used as the name.

Please note that this setting does have **no** effect if the "Take source system from Syslog message" setting is checked. In this case, the message is always taken from the Syslog message itself.

## Enable RFC 3164 Parsing

If this box is checked, [RFC 3164](#) compliant message parsing is enabled. If unchecked, "traditional" Adiscon message parsing is selected. If you experience trouble with the sender host name or the timestamp, we suggest that you turn off RFC 3164 compliant message parsing. Many existing devices do not fully comply with RFC 3164 and this can cause those issues.

## Default Ruleset Name

Name of the rule set to be used for this service. The rule set name must be valid.

### 6.3.3 SETP Server

Configures a SETP server service. A SETP server is used inside the MonitorWare line of products to reliably receive events from other systems. There are only few configuration options, as SETP takes the original message from the sender and uses the exact settings that the sender was configured for. No alteration occurs at the SETP server side; as such, no values need to be configured for the message format.

Settings are saved.

Save Reset Save and Close ?

Listener Port 5432

Enable SSL / TLS Encryption. Note if this Option is enabled, NON-SSL Clients will not be able to connect to this Service.

Rule Set to Use

Forward to SETP Server Refresh

*SETP Server Properties*

## Listener Port

The port the SETP server listens on. The default value is 5432. This should be changed only if there is a definite need for it. Such a need typically arises from security concerns. If the port is changed, all reporting agents must also be configured to use the non-standard port. SETP operates over TCP .

## Enable SSL/TLS

If this option is enabled then this action will be able to connect to SSL/TLS SETP servers. Please make sure that you want this option to be enabled.

**Please note:** If this option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

## Default Ruleset Name

Name of the rule set to be used for this service. The rule set name must be valid.

### 6.3.4 Event Log Monitor

This dialog configures the Windows Event Log Monitor service.

This service was initially introduced by Adiscon's EventReporter product. To allow previous EventReporter customers seamless upgrades, there are a number of compatibility settings to support older message formats.

Enable: Event Log Monitor

Settings are saved. Save Reset Save and Close ?

**General Options**

Use Legacy Format

Add Facilitystring

Add Username

Add Logtype

Syslog Message Numbers

Sleep Time (ms): 1 minute

Overrun Prevention Delay (ms): 5

Configure for MoniLog

**EventLogTypes**

Enable Application Event Log Advanced

Enable Security Event Log Advanced

Enable System Event Log Advanced

Enable Directory Event Log Advanced

Enable DNS Event Log Advanced

Enable File Replication Event Log Advanced

**Rule Set to Use**

Defaults Refresh

## Use Legacy Format

This option enhances compatibility to scripts and products working with previous versions of EventReporter. The legacy format contains all Windows event log properties within the message itself.

The new format includes the plain text message only. The additional information fields (like event ID or event source) are part of the XML formatted event data. If the new format is used, we highly recommend sending or storing information in XML format. This is an option in each of the action properties (of those actions that support it – the write database option for example always stores the fields separated, so there is no specific option to do so).

**Legacy format is meant to support existing parser scripts. We encourage you to use the new, XML-bound format for new implementations.** Legacy format will be maintained in future releases to support backward compatibility, but it is no longer actively being developed. There are some shortcomings in legacy format, which can lead to issues when building or operating a log parser. These shortcomings are by design. We will not change this in legacy format - the solution is to use the new format. After all, the new format was created in order to address the issues with legacy format.

## Add Facility String

If checked, facility identification is prepended to the message text generated. This parameter enhances compatibility with existing Syslog programs and greatly facilitates parsing the generated entries on the Syslog server. We strongly encourage users to use this enhancement.

**This setting only applies if the "Use Legacy Format " option is checked.**

Otherwise, it does not have any meaning and consequently cannot be configured in that case.

## Syslog Message Numbers

If checked, a continuously advancing message number is prepended to the generated message. This is useful for syslog delivery to make sure that all messages have been received at the remote server.

**This setting only applies if the "Use Legacy Format " option is checked.**

Otherwise, it does not have any meaning and consequently cannot be configured in that case.

## Add Username

If checked, the NT user that generated the event log entry is transmitted. If unchecked, this information is not forwarded. This is a configurable option for customers who have written parsing scripts for a previous format which did not contain Usernames. This option must also be unchecked if MoniLog is being used.

**This setting only applies if the "Use Legacy Format " option is checked.**

Otherwise, it does not have any meaning and consequently cannot be configured in that case.

## Sleep Time

The event log monitor periodically checks for new event log entries. The "Sleep Time" parameter specifies how often this happens. This value is in milliseconds.

We recommend a value of 60000 milliseconds for the "Sleep Time". With that setting, the event log monitor will check for new events every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended.

Very security-aware environments might use a shorter interval. The event log monitor service is specifically designed to limit the burden on the monitored system. As such, resource usage is typically low, even with frequently run event log checks. However, we recommend not running the event log monitor more often than once a second.

## Overrun Prevention Delay

This property allows configuring a delay after generating an event. The time is the delay in milliseconds.

If run at a value of zero, the event log monitor service generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because the service runs at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the service can still generate 1000 events per second.

The default setting is an overrun protection of five millisecond, which allows roughly 200 events per second. This should be sufficient for even very busy servers.

## Configure for MoniLog

To work together with [MoniLog](#), the event log monitor needs to be especially configured. When clicking the "Configure for MoniLog" button, all the necessary configurations are made automatically. The checkboxes "Use Legacy Format" and "Add Facilitystring" will be checked, while "Add Username", "Add Logtype" and "Syslog Message Numbers" will be unchecked.

Please note that these settings are **not** compliant with [MonitorWare Console](#), which needs the new, non-legacy message format. If you need to support both products, you need to configure two Event Log Monitor services.

## Event Log Types

The "Event Log Types" configure per-event-log settings. The corresponding log will only be processed if the respective "Enable" checkbox is checked. The parameters are common to all logs and will be explained only once. Each dialog looks similar:

Report Truncated Log

Syslog Facility: LOCAL\_0

Last Record: 17342

Event types to log

<input checked="" type="checkbox"/> Success	<input checked="" type="checkbox"/> Information
<input checked="" type="checkbox"/> Error	<input checked="" type="checkbox"/> Audit Success
<input checked="" type="checkbox"/> Warning	<input checked="" type="checkbox"/> Audit Failure

OK Cancel



## Report Truncated Log

Windows event logs can be truncated programmatically or via the Windows Event Viewer program. When a log is truncated, all information is erased from it. Any entries not already processed by the service will be lost.

The service detects event log truncation. If "Report Truncated Log" is checked, it will generate a separate message stating the truncation. This option is most useful in environments where truncation is not expected and as such might be an indication of system compromise.

If you regularly truncate the NT event logs as part of your day-to-day operation, we suggest you turn this option off. In this case, we also recommend using a short sleep period (for example 10,000 which is 10 seconds) to avoid losing log entries.

## Syslog Facility

The Syslog facility to map information units stemming from this log to. Most useful if the message shall be forwarded to a Syslog daemon.

## Last Record

Windows event log records are numbered serially, starting at one. The service records the last record processed. This textbox allows you to override this value. Use it with caution!

If you would like a complete dump of a specific Windows event log, reset the "Last Record" to zero. If you missed some events, simply reset it to some lower value than currently set. It is possible to set "Last Record" to a higher value. This will suspend event reporting until that record has been created. We strongly discourage to use this feature unless definitely needed.

## Event Types to Log

These checkboxes allow local filtering of the event log. Filtering is based on the Windows event type. There is a checkbox corresponding to each Windows event type. Only checked event types will be processed. Unchecked ones will be ignored.

Filtering out unnecessary log types at this level enhances system performance because no information units will be generated and passed to the rule engine. Thus, Adiscon strongly recommends dropping unnecessary log types.

## Default Ruleset Name

Name of the rule set to be used for this service. The rule set name must be valid.

### 6.3.5 File Monitor

The file monitor monitors the content of a text file just as the event monitor monitors the NT event log. Its purpose is to gather vital information that is stored in system text files. Many applications do not write events to the event log but to a text file. This is also the case with many Microsoft applications (for example the WINS log).

The file monitor can also gather Internet Information Server (Windows' web server) log files. This is very useful for monitoring web activity and detecting attacks.

The screenshot shows the File Monitor configuration dialog box. At the top, there is a checked checkbox labeled "Enable: File Monitor" and a status bar that says "Settings are saved." To the right of the status bar are buttons for "Save", "Reset", "Save and Close", and a help icon. The main configuration area includes:

- "File and path name": A text input field with a "Browse" button.
- "Timemode used for Filename": A dropdown menu set to "Localtime".
- "Check Interval (ms)": A dropdown menu set to "1 minute".
- "Overrun Prevention Delay (ms)": A dropdown menu set to "5".
- "Logfile Type": A dropdown menu set to "Standard".
- "General Values" section:
  - "Syslog Facility": A dropdown menu set to "LOCAL0 (16)".
  - "Syslog Priority": A dropdown menu set to "INFO (6)".
  - "Resource ID": An empty text input field.
  - "Syslog Tag Value": A text input field containing "MwFileMonitor".
- "Rule Set to Use": A dropdown menu set to "Defaults" with a "Refresh" button.

#### File and path name

Here, type the name of the file to be monitored. To select a file from a browser, press the browse button. If a complete file name is specified, exactly that file is monitored.

The file name will never change automatically. However, many systems generate changing log files. For example, Internet Information Server can be configured to change the log file every day. Therefore, each day's log file has a different name.

To support changing log file names, there are replacement characters available within

the file name. These are:

Character	Meaning
%y	Year with two digits (e.g. 2002 will become "02")
%Y	Year with 4 digits
%m	Month with two digits (e.g. March will become "03")
%M	Minute with two digits
%d	Day of month with two digits (e.g. March, 1 <sup>st</sup> will become "01")
%h	Hour as two digits
%S	Seconds as two digits. It is hardly believed that this will ever be used in reality.
%w	Weekday as one digit. 0 means Sunday, 1 Monday and so on.
%W	Weekday as three-character string. Possible values are "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat". This replacement character is most useful for DHCP log files.

**Please note: the replacement characters are case sensitive!**

For example, daily Internet Information Server log files are named "exyymmdd.log", with yy being the 2 digit year, mm the month and dd the day of month. To generate the same name with file monitor, use the following name "ex%y%m%d.log". Please note that there is no replacement character for the monthly week number (1<sup>st</sup> week, 2<sup>nd</sup> week). As such, the weekly log file setting of IIS is not supported.

### TimeMode Used for Filename

Select the time mode used for the log file to be monitored with this drop-down list. Available options are:

Local Time: log file is monitored based on local time.

UTC: log file is monitored based on universal coordinated time. UTC was formerly referred to as "GMT" (Greenwich Mean Time) and is the basis of the international time zone system.

### Check Interval

This is the interval, in milliseconds that the file monitor checks the file for new records.

We recommend a value of 60000 milliseconds for the "Check Interval". With that setting, the file monitor will check for new records every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended.

Very security-aware environments might use a shorter interval. The MonitorWare Agent 2.0 is specifically designed to limit the burden on the monitored system. As such, resource usage is typically low, even with frequently run file monitor checks. However, we recommend not running the file monitor more often than once a second.

## **Overrun Prevention Delay**

This property allows configuring a delay after generating an event. The time for the delay is in milliseconds.

If run at a value of zero, the MonitorWare Agent 2.0 generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because MonitorWare Agent 2.0 runs at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, MonitorWare Agent 2.0 can still generate 1000 events per second.

The default setting is an overrun protection of five millisecond, which allows roughly 200 events per second. This should be sufficient for even very busy servers.

## **Logfile Type**

Select the type of the log file to be monitored with this drop-down list. Available options are:

Standard - a standard text log file

W3C Web Server logfile - log files in the W3C web server compliant format.

## **Syslog Facility**

The Syslog facility to be assigned to events created by the File Monitor service. Most useful if the message shall be forwarded to a Syslog daemon.

## **Syslog Priority**

The Syslog priority to be assigned to events created by the File Monitor process. Most useful if the message shall be forwarded to a Syslog daemon.

## Resource ID

The resource id to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a Syslog daemon.

## Syslog Tag Value

The Syslog tag value to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a Syslog daemon.

## Default Ruleset Name

Name of the rule set to be used for this service. The rule set name must be valid.

### *Further Reading*

Please visit our white paper [monitoring IIS logs](#).

## 6.3.6 Heartbeat

The heartbeat process can be used to continuously check if everything is running well. It generates an information unit every specified time interval. That information unit can be forward to a different system. If it does not receive additional packets within the configured interval, it can doubt that the sender is either in trouble or already stopped running.

The screenshot shows the configuration window for the heartbeat service. At the top, there is a checked checkbox labeled "Enable: Heartbeat". Below it, a status bar says "Settings are saved." and there are buttons for "Save", "Reset", and "Save and Close", along with a help icon. The main configuration area includes:

- "Message that is send during each heartbeat": A text area containing "I am still running".
- "Heartbeat clock (SleepTime)": A dropdown menu set to "1 minute".
- "General Values" section:
  - "Syslog Facility": A dropdown menu set to "LOCAL0 (16)".
  - "Syslog Priority": A dropdown menu set to "INFO (6)".
  - "Resource ID": An empty text field.
  - "Syslog Tag Value": A text field containing "MWHeartbeat".
- "Rule Set to Use": A dropdown menu set to "Defaults" with a "Refresh" button next to it.

## Message to Send

This is the message that is used as text inside the information unit. Use whatever value is appropriate. The message text does not have any special meaning, so use whatever value you seem fit.

## Sleep Time

This is the interval, in milliseconds, that the heartbeat service generates information units in. Please note that the receiving site should be tolerant. The interval specified here is the minimum time between packets. Under heavy load, the interval might be slightly longer. It is good practice to allow twice this interval before the service is considered suspect by the system monitoring the services health.

## Syslog Facility

The Syslog facility to be assigned to events created by the heartbeat service. Most useful if the message shall be forwarded to a syslog server.

## Syslog Priority

The Syslog priority to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a syslog server.

## Syslog Tag Value

The Syslog tag value to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a syslog server.

## Resource ID

The resource id to be assigned to events created by the heartbeat process. Most useful if the message shall be forwarded to a syslog server.

## Default Ruleset Name

Name of the rule set to be used for this service. The rule set name must be valid.

### 6.3.7 Ping Probe

The ping probe can be used to check the health of a remote system. The ping probe process sends ping messages (more precisely: ICMP Echo Requests) to a configured system. If configured properly, the remote system will send a response. If this response is received, the machine and its IP stack are operating. This does not indicate, however, that all services on this machine are alive.

If no response is received, the remote system or its IP stack is most probably not operating properly. However, the ping message might have been lost in transit or the round-trip time might have been too long so that a timeout occurred. Therefore, a single failing ping makes a system suspect, but it alone cannot be used to confirm problems at the remote system. If multiple successive pings fail, it is relatively safe to assume that the remote system has failed

Please note that most firewall setups do not allow ping messages. As such, a system behind a firewall typically cannot be pinged and the ping probe cannot be used in this configuration. If in doubt, please check with your firewall administrator.

The ping probe is typically used to check the availability of a remote system. The ping probe periodically sends the ping messages. As long as responses are received, nothing happens. If no response is received, it generates an event and passes it to the rule engine. As ping messages can get lost, the ping probe will retry failed probes before it reports an error. Both the number of retries and the retry interval can be specified.

<input checked="" type="checkbox"/> Enable: Ping Probe	Save	Reset	Save and Close	?
Settings are saved.				
Probe Interval	1 minute			
Timeout limit	1 second			
IPAddress or Hostname	172.19.1.18			
Number Of Retries	3			
Retry Interval (ms)	5 seconds			
<input type="checkbox"/> Generate an event if PingProbe was successfull				
General Values				
Syslog Facility	LOCAL0 (16)			
Syslog Priority	INFO (6)			
Resource ID				
Syslog Tag Value	MwPingProbe			
Rule Set to Use				
	Ping Probe			Refresh

## Probe Interval

This is the interval of the ping probes. After each probe, the MonitorWare Agent 2.0 ping probe process goes "to sleep". This period is specified in

## Timeout Limit

The amount of time (in milliseconds) the remote system is expected to answer in. If no response is received within this period, the ping fails and an event is generated. The default value of 1000 milliseconds is a proper value for most well connected networks. If the ping probe runs against a heavily loaded system and/or slow network link, the amount must be adjusted accordingly.

## IP Address / Hostname

Either the IP address or resolvable host name of the system the ping probe is to be run against. This system has been called "remote host" in the description above. Please note that specifying a host name can cause the ping probe to fail if DNS name resolution fails (for example due to a failing DNS server). To avoid this, specify an IP address.



## **Syslog Facility**

The Syslog facility to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

## **Number of Retries**

If a ping fails, it is first retried to see if it is a persistent problem. The "Number of Retries" controls how many retries will be made. If this is set to zero, no retries will be made and a ping probe fail event is immediately generated.

For typical systems, we recommend a setting of three retries. This is also the default value.

## **Retry Interval**

If there is a temporary network issue like network congestion, it most probably takes some seconds to resolve it. As such, an immediate retry might not be appropriate. To delay it, configure a retry interval. This value is in milliseconds. If a ping fails, the next retry will be after a pause specified in this property.

The default and recommended value is 5 seconds (5000 milliseconds).

## **Generate an event if Ping Probe was successful**

When checked, an event is generated every time. If unchecked, it is generated only when the ping fails. The most common option is to leave it unchecked to catch events upon a failed ping.

## **Syslog Priority**

The Syslog priority to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

## **Syslog Tag Value**

The Syslog tag value to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

## **Resource ID**

The resource id to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Default Ruleset Name**

Name of the rule set to be used for this service. The rule set name must be valid.

#### **6.3.8 Port Probe**

The port probe is very similar to the ping probe described above. The main difference is that it does not check the IP stack availability but rather a specific TCP port.

The difference here is that using this method a specific service on the remote machine is monitored, for example a mail (SMTP) server. The port probe tries to connect to the service port (25 in our example). If that fails, the service is definitely not running. In this case, an event will be generated. A single event is a definite indication of problems, as such there is no need for repetitive failures before initiating action on this (although this can be configured in the rule engine).

Being able to connect to the remote machine and service TCP port most probably means that the remote service is running. However, more certainty can be gained by actually initiating some communication with the service. The exact application protocol needs to be known to try this test. Thus, this step is optional. If turned on, a single command can be send to the remote service and a single response will be expected back and be compared to a pre-defined response. This does not take care of all possible application protocols, but provides an additional layer of confidence for important services like SMTP. It is up to the user to know the command sequences that a given service will understand and reply with.

As a rule of thumb, the port probe provides superior protection against service failure even without checking the message exchange. So if in doubt, use it without this advanced feature.

Please note that the port probe can probe TCP based services only. Most application services are TCP based, but there are some – mostly system – services out there, that are not. One of the most notable exceptions is DNS, which is operated primarily over UDP. In UDP, there is no notion of a session and as such, it is not possible to probe the session setup, which essentially is what the port probe does. As such, a port probe can unfortunately not be used to check the status of those services. However, the majority of services like application server, databases, mail, web and a large number of others can be used with the port probe.

The screenshot shows a configuration window for the Port Probe. At the top, there is a checked checkbox labeled "Enable: Port Probe". To its right are buttons for "Save", "Reset", and "Save and Close", along with a help icon. Below this is a status bar that says "Settings are saved." The main configuration area includes:

- Probe Interval:** A dropdown menu set to "1 minute".
- Timeout Limit:** A dropdown menu set to "1 second".
- IP Address or Hostname:** A text field containing "192.168.1.1".
- Port:** A text field containing "0".
- Two unchecked checkboxes: "Generate an event if PortProbe was successfull" and "Send Message and check for expected message".
- Two empty text fields labeled "Message to send" and "Message expected".
- A section titled "General Values" containing:
  - Syslog Facility:** A dropdown menu set to "LOCAL0 (16)".
  - Syslog Priority:** A dropdown menu set to "INFO (6)".
  - Resource ID:** An empty text field.
  - Syslog Tag Value:** A text field containing "MWPortProbe".
- A section titled "Rule Set to Use" with a dropdown menu set to "Ping Probe" and a "Refresh" button.

## Probe Interval

This is the interval of the port probes. After each probe, the MonitorWare Agent 2.0 port probe process goes "to sleep". This period is specified in milliseconds.

## Timeout Limit

The amount of time (in milliseconds) the remote system is expected to answer in. If no response is received within this period, the ping fails and an event is generated. The default value of 1000 milliseconds is a proper value for most well connected networks. If the ping probe runs against a heavily loaded system and/or slow network link, the amount must be adjusted accordingly.

## IP Address / Hostname

Either the IP address or resolvable host name of the system the ping probe is to be run against. This system has been called "remote host" in the description above.

Please note that specifying a host name can cause the ping probe to fail if DNS name resolution fails (for example due to a failing DNS server). To avoid this, specify an IP address. Please note that you typically can use 127.0.0.1 (the so-called loop back address) to check a service that is running on a local machine. This ability might be limited by service configuration, because the service must listen to that IP address).

## Port

This port is to be probed. Please see your server's reference for the actual value to use. For example, mail servers typically listen to port 25 and web servers to 80.

## Generate an event if Port Probe was successful

When checked, an event is generated every time. If unchecked, it is generated only when the port probe fails. The most common option is to leave it unchecked to catch events upon a failed port probe.

## Send Message and wait for expected Message

If left unchecked, the port probe checks the TCP session setup to the remote service only. As stated above, a successfully completed session setup most probably means the service is healthy. As an extra measure, some actual message exchange can be enabled. This is done by checking this box.

## Message to Send

This message text will be sent to the service after the TCP session has been established.

## Message Expected

This is the message expected to be received from the service. Reception starts after sending the "Message to Send". Please note that the "Message Expected" is compared against the **first** message sent from the service on the TCP session. With some protocols, this means the message compared will be an initial greeting message and **not** a response to the "Message to Send".

## Syslog Facility

---

The Syslog facility to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Syslog Priority**

The Syslog priority to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Resource ID**

The resource id to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Syslog Tag Value**

The Syslog tag value to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Default Ruleset Name**

Name of the rule set to be used for this service. The rule set name must be valid.

## **6.3.9 NT Services Monitor**

The NT Services Monitor is used to monitor if vital operating services are running. The monitor continuously checks all services set to "automatic" startup. If such a service does not run, an event is generated and passed to the rule engine.

The screenshot shows the configuration window for the NT Service Monitor. At the top, there is a checked checkbox labeled "Enable: NT Service Monitor". Below it, a status bar indicates "Settings are saved." and there are buttons for "Save", "Reset", "Save and Close", and a help icon "?".

The main configuration area includes:

- "Check Interval in Milliseconds (1000 ms --> 1 second)" set to "1 minute".
- "Delay on Startup in Milliseconds (1000 ms --> 1 second)" set to "custom" with a value of "0".
- A "General Values" section containing:
  - "Syslog Facility" set to "LOCAL0 (16)".
  - "Syslog Priority" set to "INFO (6)".
  - "Resource ID" is an empty text field.
  - "Syslog Tag Value" set to "MWNTServiceMonitor".
- A "Rule Set to Use" section with a dropdown menu set to "Defaults" and a "Refresh" button.

## Check Interval

This is the interval in which the service status is checked. This period is specified in milliseconds. The default is 60,000 ms, which is one minute. We recommend to lower this interval only if the server is performing very critical operations and service stops need to be detected in close real-time.

For performance reasons, we do not recommend using an interval of less than 2000 ms.

## Delay on Startup

During system boot, the monitoring service eventually starts before all other services have been started. As such, the service monitor will most probably find some services not running – simply because they are to be started very soon. Nevertheless, the service monitor will still generate a "service not running" event.

To avoid this situation, use the startup delay setting. It specifies an amount of time (in milliseconds) that the service monitor will be hold right after startup. So during system boot, the operating system has a chance to start all other services before the service monitor comes into action.

The actual delay is very much depending on the number of services and hardware sizing of a particular server. Typically, a value 60,000 (one minute) should be a good value. But a busy server with many services might require a much higher value.

### **Syslog Facility**

The Syslog facility to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Syslog Priority**

The Syslog priority to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Resource ID**

The resource id to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Syslog Tag Value**

The Syslog tag value to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### **Rule Set to Use**

Name of the rule set to be used for this service. The rule set name must be valid.

## **6.3.10 Disk Space Monitor**

This monitor checks the available and used space on all hard disks in the system. All hard disks present in the system are automatically checked. New disks are automatically detected. One event specifying the maximum size and the used size is generated per disk. The Disk Space Monitor runs continuously based on an interval set in the configuration.

Enable: DiskSpace Monitor  
Settings are saved. Save Reset Save and Close ?

Check Interval: 1 minute

General Values

Syslog Facility: LOCAL0 (16)

Syslog Priority: INFO (6)

Resource ID:

Syslog Tag Value: MWDiskSpaceMonitor

Rule Set to Use: Disk Space Monitor Refresh

## Check Interval

This is the interval in which the service status is checked. This period is specified in milliseconds. The default is 60,000 ms, which is one minute. This should be sufficient for a typical server. If you would like to have the disk space check run less often, you might for example use the value of 3,600,000 for one hour (or a multiple for multiple hours).

For performance reasons, we do not recommend using an interval of less than 30,000 ms.

## Syslog Facility

The Syslog facility to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

## Syslog Priority

The Syslog priority to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

## Resource ID



The resource id to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### Syslog Tag Value

The Syslog tag value to be assigned to events created by this service. Most useful if the message shall be forwarded to a Syslog server.

### Rule Set to Use

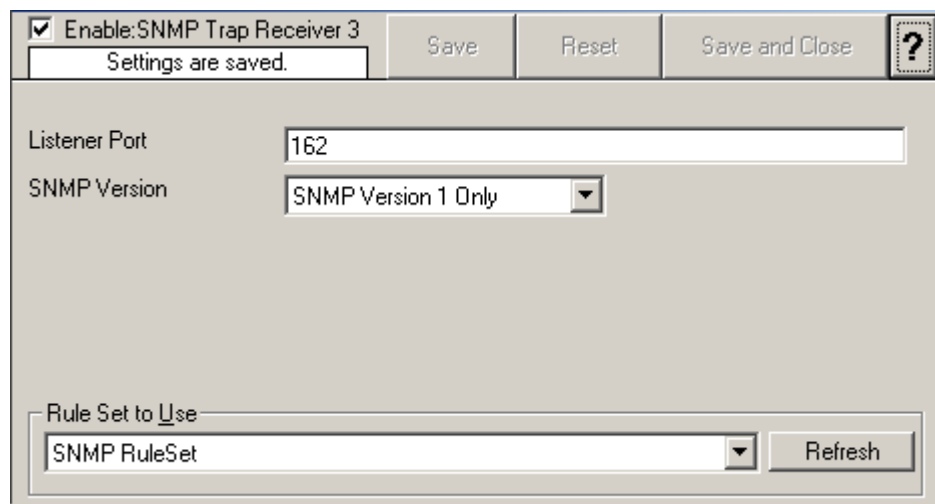
Name of the rule set to be used for this service. The rule set name must be valid.

## 6.3.11 SNMP Trap Receiver Service

SNMP Trap Receiver allows you to receive SNMP messages. A rough description of a Trap is that it is somewhat like a Syslog message, just over another protocol (SNMP). A trap is generated when the device feels it should do so and it contains the information that the device feels should be transmitted. It also contains some (few) standard items, as the version, community etc. To know more about the SNMP Trap Receiver Service please look into

<http://www.mwagent.com/Common/en/tutorial/mwagent-snmp-service.asp>

The SNMP Trap Receiver Service runs continuously based on the configuration mentioned below.



The screenshot shows a configuration window for the SNMP Trap Receiver Service. At the top, there is a checked checkbox labeled "Enable:SNMP Trap Receiver 3" and a "Settings are saved." message. To the right are buttons for "Save", "Reset", and "Save and Close", along with a help icon (question mark). The main configuration area includes a "Listener Port" text box containing "162" and an "SNMP Version" dropdown menu set to "SNMP Version 1 Only". At the bottom, there is a "Rule Set to Use" dropdown menu set to "SNMP RuleSet" and a "Refresh" button.

### Listener Port

The port the SNMP listener is listening to. If in doubt, leave it at the default of 162, which is the standard port for this.

## **SNMP Version**

Can be used to restrict the SNMP versions.

## **Rule Set to Use**

Name of the rule set to be used for this service. The rule set name must be valid.

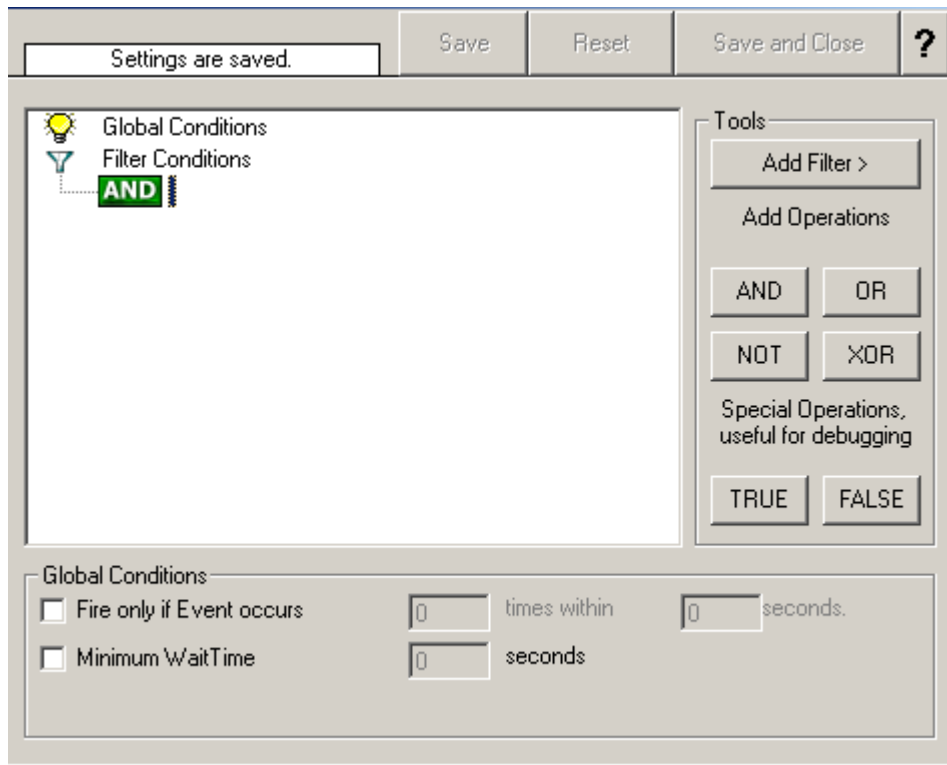
## **6.4 Filter Conditions**

### **6.4.1 Filter Conditions**

Filter conditions specify **when** to apply a rule. If the filter condition evaluates to true, the rule containing those conditions is treated as matching and the actions specified in that rule will be carried out.

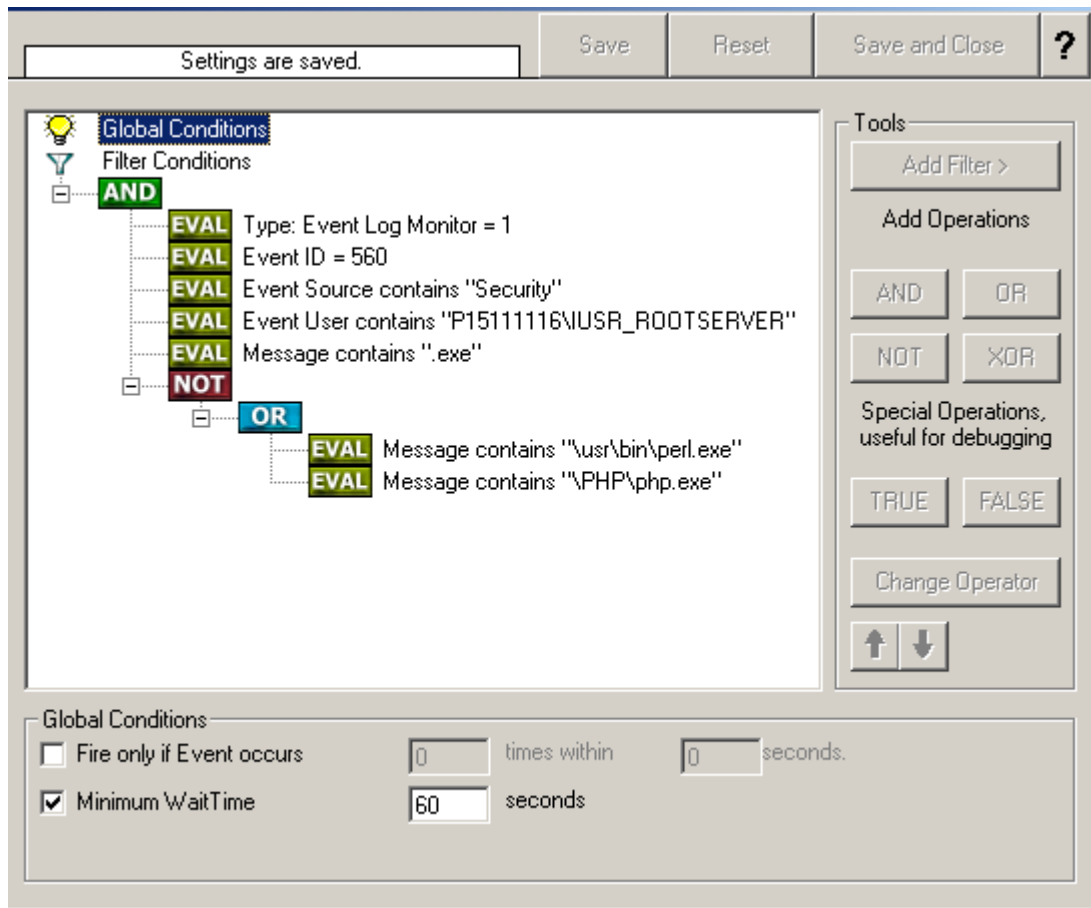
Filter conditions can be as complex as needed. Full support for Boolean operations and nesting of conditions is supported.

By default, the filter condition is empty, respective contains only a single "AND" at the top level. This is to facilitate adding filters (the top level-node is typically "AND" and thus provided by default. A filter condition containing only the "AND" always evaluates as true. A sample screenshot can be found below:



The default filter condition means that the actions associated with the rule are to be carried out for every information unit received. It is often used for actions that should be broadly taken, for example to write all incoming information units to a database or text file.

On the other hand, there are actions that should only be executed under very special conditions. They may even require a complex filter condition including multiple levels of Boolean operations. Below is a sample of such a condition:



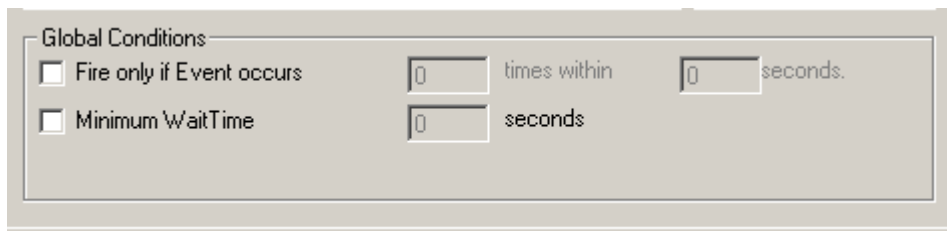
This filter condition is part of an intrusion detection rule set. Here Windows file system auditing is used to detect a potentially successful intrusion via Internet information server. This is done by enabling auditing on all executable files. Internet Information Server will access them under the IUSR\_<machinename> account, which in our sample is "P15111116\IUSR\_ROOTSERVER". If that user runs any unexpected executables, chances are good that someone was able to intrude the machine via IIS. Please note that Perl and PHP scripts need to run the Perl and PHP engine. This is reflected by specifically checking if perl.exe and php.exe is executed – and if so, no alarm shall be triggered.

Here is how the above sample works: first, the message contents are checked if it contains either the full path name to perl.exe or php.exe. This is done in the "OR" branch at the bottom. We now need to keep in mind that when a filter condition evaluates to "true", the actions are executed. In case of perl.exe and php.exe, this is just the opposite of what we want. We need it to be executed, when other files are executed. Consequently, we negate (Boolean "NOT") the result of the OR. The end result of the "NOT" operation is then combined via a "AND" with some other properties describing the event we need. First, we check if the specific event really occurred. For this, we need to make sure we deal with an Event Log Monitor information unit. Then, these information units are identified by the event source as well as the event id. We also check for the event user to identify only IIS generated requests. Lastly, we check if the message contains the string ".exe".

In order to avoid too frequent alerts, we also have specified a minimum wait time of 60 seconds. So the filter condition will evaluate as "true" at most every 60 seconds, even if all other conditions are true.

## 6.4.2 Global Conditions

Global Conditions apply to the rule as whole. They are automatically combined with a logical "AND" with the conditions in the filter tree.



The screenshot shows a dialog box titled "Global Conditions" with two rows of configuration options:

- Row 1:  Fire only if Event occurs [0] times within [0] seconds.
- Row 2:  Minimum WaitTime [0] seconds

### Fire only if Event occurs

This is kind of the opposite of the "Minimum Wait Time". Here, multiple events must come in before a rule fires. Take another example. This time, we use a ping probe. Ping is not a very reliable protocol, so a single ping might be lost. Thus, it may not be the best idea to restart some processes just because a single ping failed. It would be much better to wait for repetitive pings to fail before doing so.

Exactly this is why the "Fire only if Event Occurs" filter condition is made for. It waits until a configured amount of the same events occurs within a period. Only if the count is reached, the filter condition matches and the rule can fire.

If you used previous versions of the product, you might remember a filter called "Occurrences". This has just been renamed.

### Minimum Wait Time

This filter condition can be used to prevent rules from firing too often. For example, a rule might be created to check the status of a port probe event. The port probe probes an SMTP server. If the event is fired and the rule detects it, it will spawn a process that tries to restart the service. This process will take some time. Maybe the SMTP gateway needs some more time to fully start up so that the port probe might fail again while the problem is already taken care of. The port probe as such will generate an additional event. Setting a minimum wait time will prevent this second port probe event to fire again if it is – let's say – within 5 minutes from the original one. In this case, the minimum wait time is not yet reached and as such, the rule will not match.

If, however, the same event is generated 5 hours later (with the mail gateway failing again), the rule will once again fire and corrective action taken.

### 6.4.3 Operators

In general, Operators describes how Filter conditions are linked together. The following Operators can be used.

#### **AND**

All filters placed below must be true. Only then AND will return true.

#### **OR**

Even if one of the filter placed below OR is true, OR will return true.

#### **NOT**

Only one Filter can be placed below NOT operator, and if the filter evaluation is true, NOT will return false.

#### **XOR**

Only one to two Filters are possible in the XOR Operator.

#### **TRUE**

Useful for debugging, will just return TRUE.

#### **FALSE**

Useful for debugging as well, will return FALSE.

### 6.4.4 Filters

Filters can be added under each Operation node. There are a few common filters which can be used for all Services, and there are special filters which only apply if a special kind of Information Unit is evaluated. Note, if a filter is used that does not apply to the evaluated Info Unit, it will be just ignored. This gives you the possibility to build one Filter set for several types of Information Units.

For details on how filter conditions are evaluated, please see "Filter conditions" on page 9.

There are different types of Filter, and so there a different ways in which you can

---

compare them to a value. The following Types exist:

### **String**

Can be compared to another String with "=", "Not =" and "Range Match".

### **Number**

Can be compared with another number with "=", "Not =", "<" and ">"

### **Boolean**

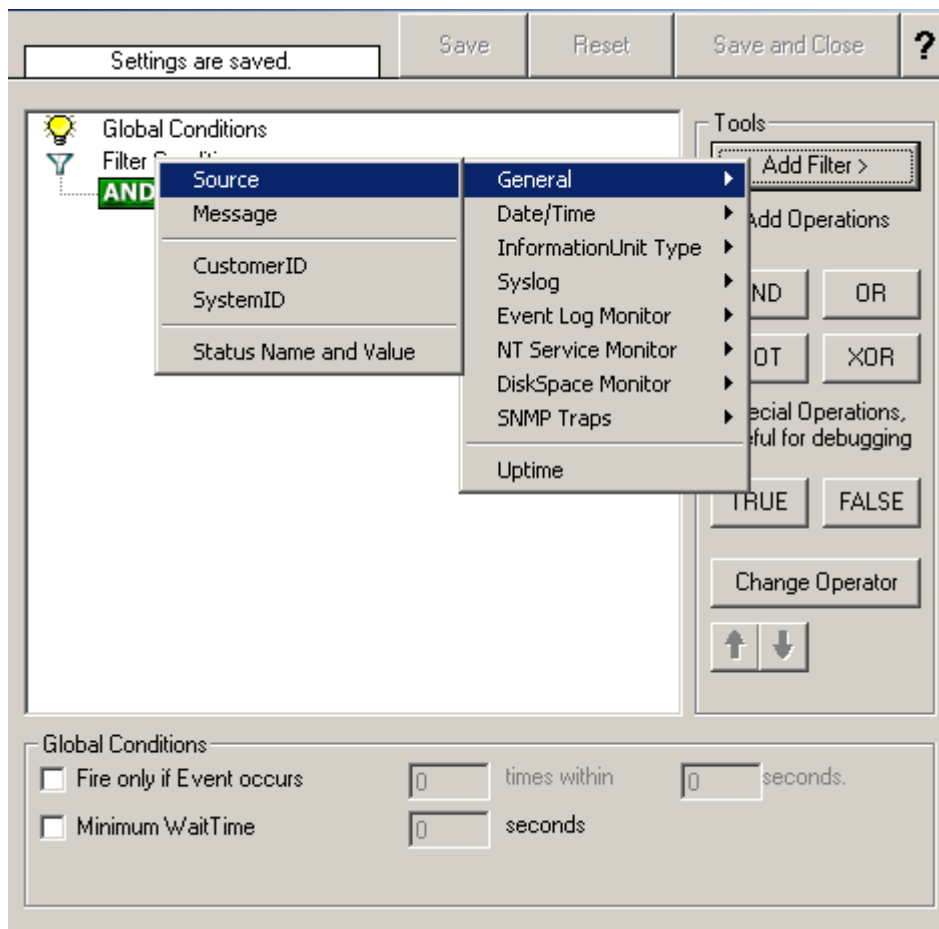
Can be compared to either TRUE or FALSE with "=" and "Not ="

### **Time**

Can be compared with another time but only with "=".  
Below is a List of possible filters, which can be evaluated.

### 6.4.5 General

These are non-event log specific settings.



#### Source System

This filter condition checks the system that generated the information unit. For example, in case of the Syslog server, this is the Syslog device sending a Syslog message.

This filter is of type string and should contain the source system name or IP address.

#### Message Content

The message content filter condition is very powerful. It evaluates to true if the specified content is found anywhere within the message. As there is implicit wildcarding, there is no need for extra wildcards to be specified.



The content search can be limited to a region within the message. To do so, select a starting and ending position within the string. This can be done via the start and end list boxes. Please note that you can enter the character position you desire in these fields. The default "Start" and "End" or only there as shortcuts. If you would like to search for a string just between positions 10 and 50, specify these values as start and end values, respectively.

This filter is of type string.

### **CustomerID**

CustomerID (Type=Number).

### **SystemID**

SystemID (Type=Number).

### **Status Name and Value**

This filter type corresponds to "Set Status Action" on page 85. Status Name and Value (Type=String)

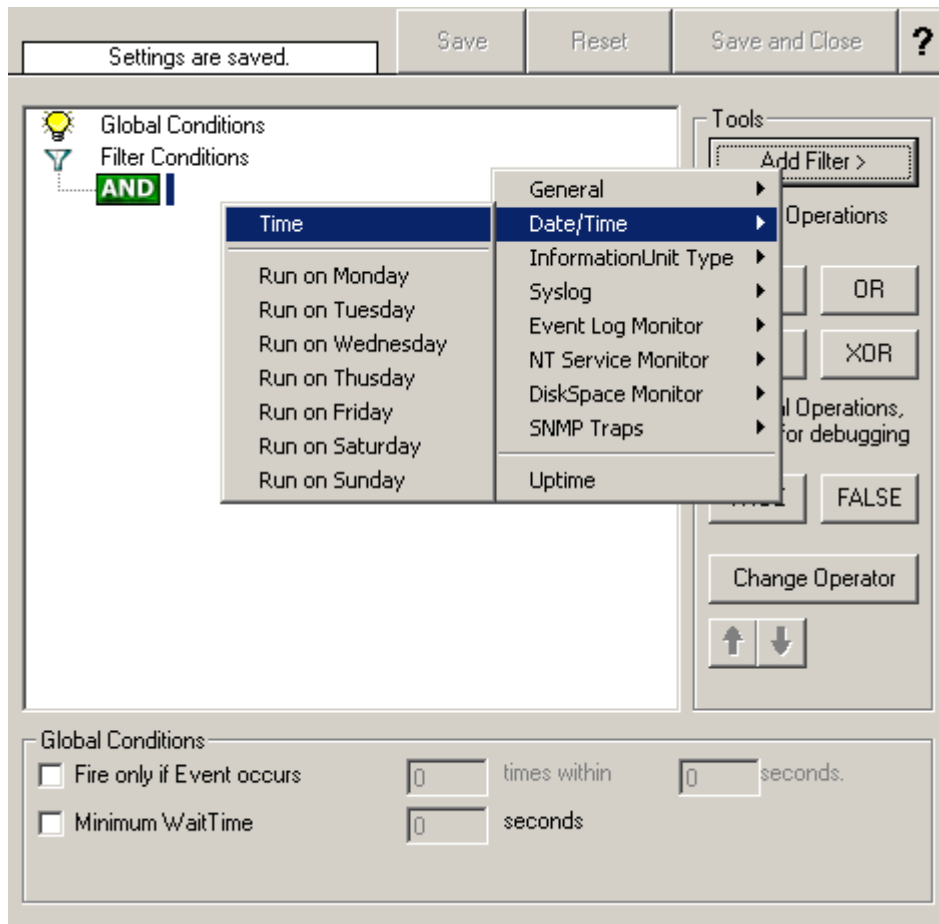
#### **6.4.6 Date/Time**

This filter condition is used to check the time frame (and/or day of week in which an event occurred).

For example, a Syslog message from a Cisco router saying that it dialed up is normal if it occurs during office hours.

If it occurs at night, so, it is an alerting signal and an administrator might receive notification of this event (while he might otherwise decide to discard it).

This can be done with the time setting.



The following filters are available in detail:

Start time (Type=Time)

End Time (Type=Time)

Run on Monday (Type=Boolean)

Run on Tuesday (Type=Boolean)

Run on Wednesday (Type=Boolean)

Run on Thursday (Type=Boolean)

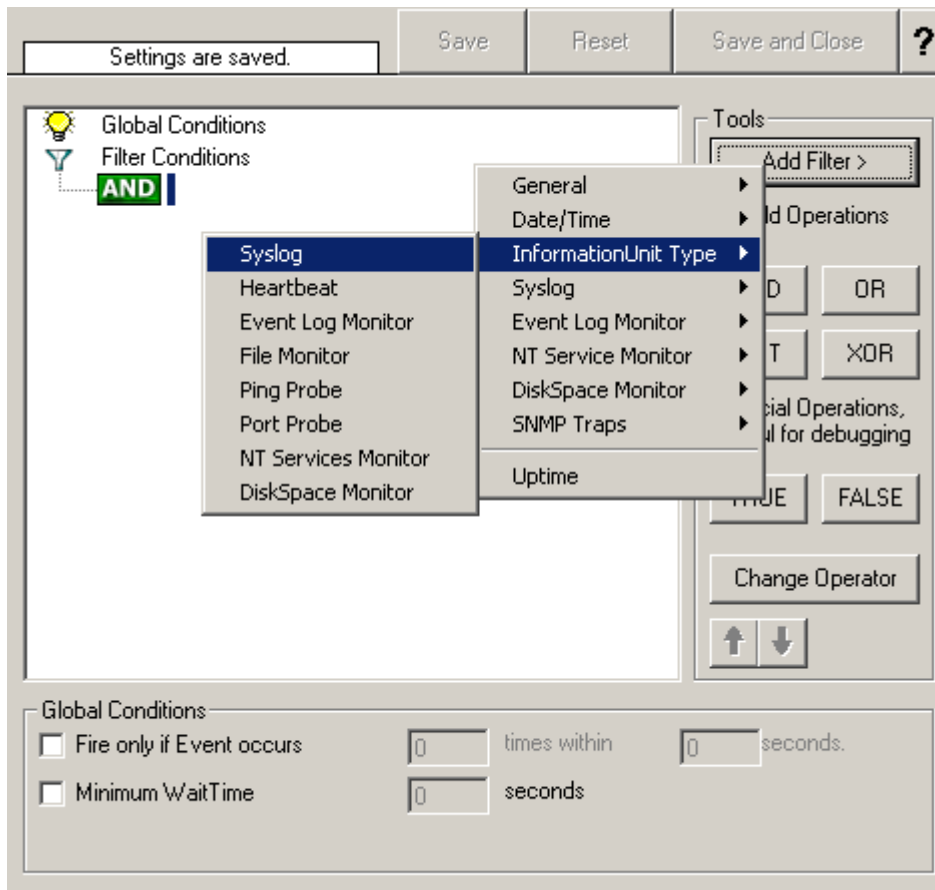
Run on Friday (Type=Boolean)

Run on Saturday (Type=Boolean)

Run on Sunday (Type=Boolean)

### 6.4.7 InformationUnit Type

Select the specific information if a rule should just be processed for some information unit types. This is especially useful if a specific type needs non-standard processing. There is one pre-defined filter for each possible InformationUnit Type available (shown below).



Syslog (Type=Boolean)

Heartbeat (Type=Boolean)

Event Log Monitor (Type=Boolean)

File Monitor (Type=Boolean)

Ping Probe (Type=Boolean)

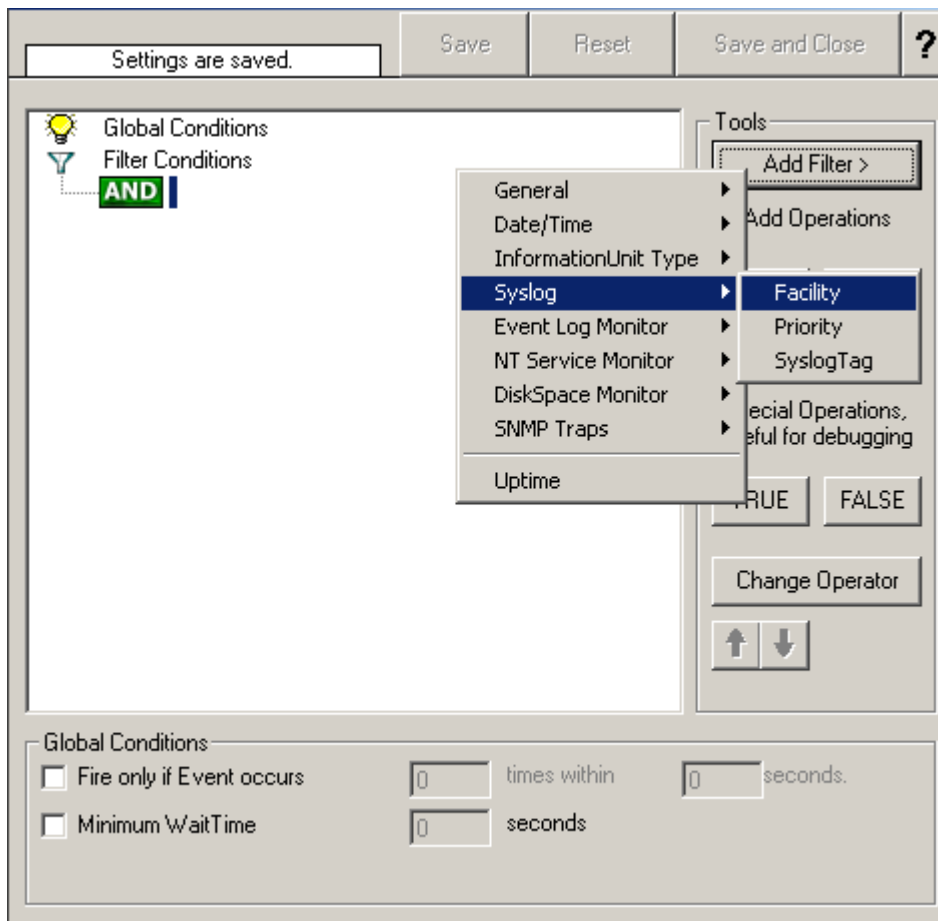
Port Probe (Type=Boolean)

NT Services Monitor (Type=Boolean)

Disk Space Monitor (Type=Boolean)

### 6.4.8 Syslog

Syslog related filters are grouped here. Please keep in mind that every Information Unit has assigned a Syslog priority and facility and thus these filters can be used with all Information Units.



### Syslog Facility

The information unit must have the specified Syslog facility value. For Syslog type information units, it is the actual Syslog priority code, for all others it is a value mapped on a best effort basis.

This filter is of type number.

## Syslog Priority

The information unit must have the specified Syslog priority value. For Syslog type information units, it is the actual Syslog priority code, for all others it is a value mapped on a best effort basis.

The first list box allows to set a matching mode. The operations "less than" (<), "greater than" (>) and "equal" (=) can be selected. The match is made depending on these operations, so a "less than" operation means that all priorities below the specified priority math. Please note that the specified priority is **not** a match. If you would like to include it, be sure to specify the next higher one.

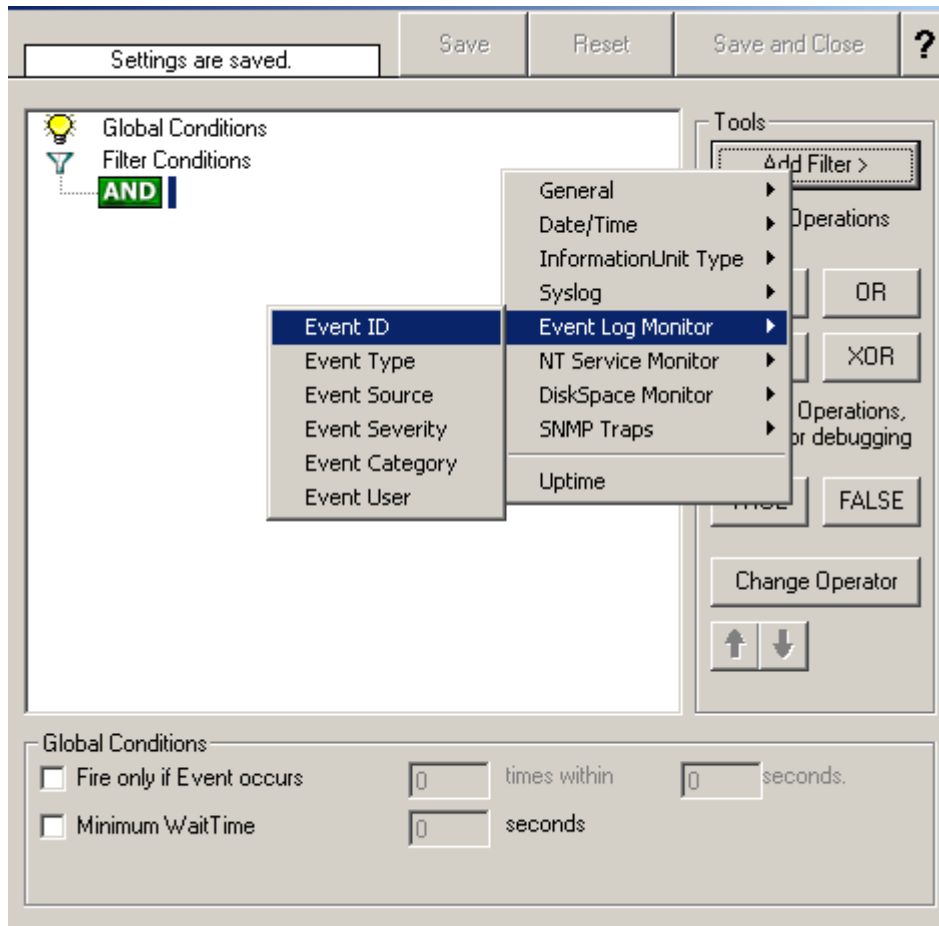
This filter is of type number.

## Syslog Tag

This filter is of type string.

## 6.4.9 Event Log Monitor

Event log monitor specific filters are grouped here.



### Event ID

This is the event log id as specified in the NT event log. If enabled, the event must have the configured event id or the rule will not match. This is an integer value.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type number.

### Event Type

This is the event log type as specified in the NT event log. If enabled, the event must have the configured event type or the rule will not match. The supported values can

be selected from the list box.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type string.

### **Event Source**

This is the event log source as specified in the NT event log. If enabled, the event must have the configured event source or the rule will not match. This is a string value. There must be an exact match. Please note that this value is case-sensitive.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type string.

### **Event Severity**

This is the event log severity as specified in the NT event log. If enabled, the event must have the configured severity or the rule will not match. The supported values can be selected from the list box.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type number.

### **Event Category**

This is the event log category as specified in the NT event log. If enabled, the event must have the configured event category or the rule will not match.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type number.

### **Event User**

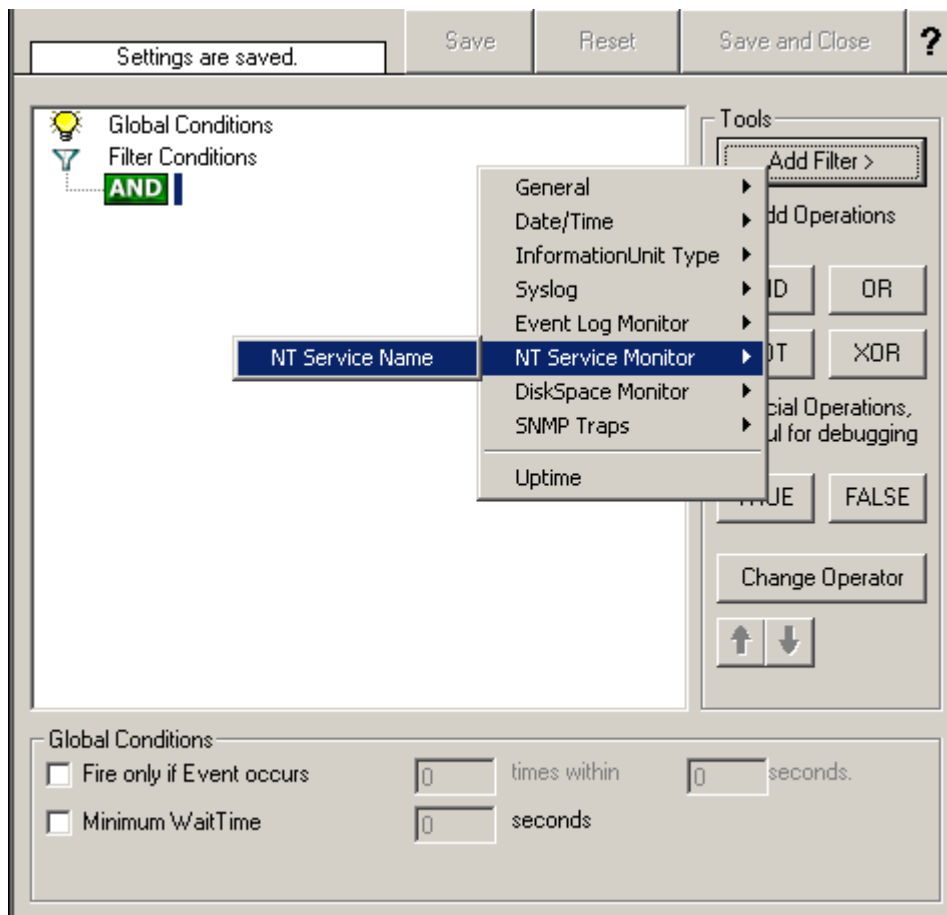
This is the event log user as specified in the NT event log. If enabled, the event must have the configured event user or the rule will not match. Since it's a string value there must be an exact match. Please note that this value is case-sensitive.

This filter condition should only be used with event log information units. If used with others, a mapped value will be used which might not properly reflect the actual value.

This filter is of type string.

### 6.4.10 NT Service Monitor

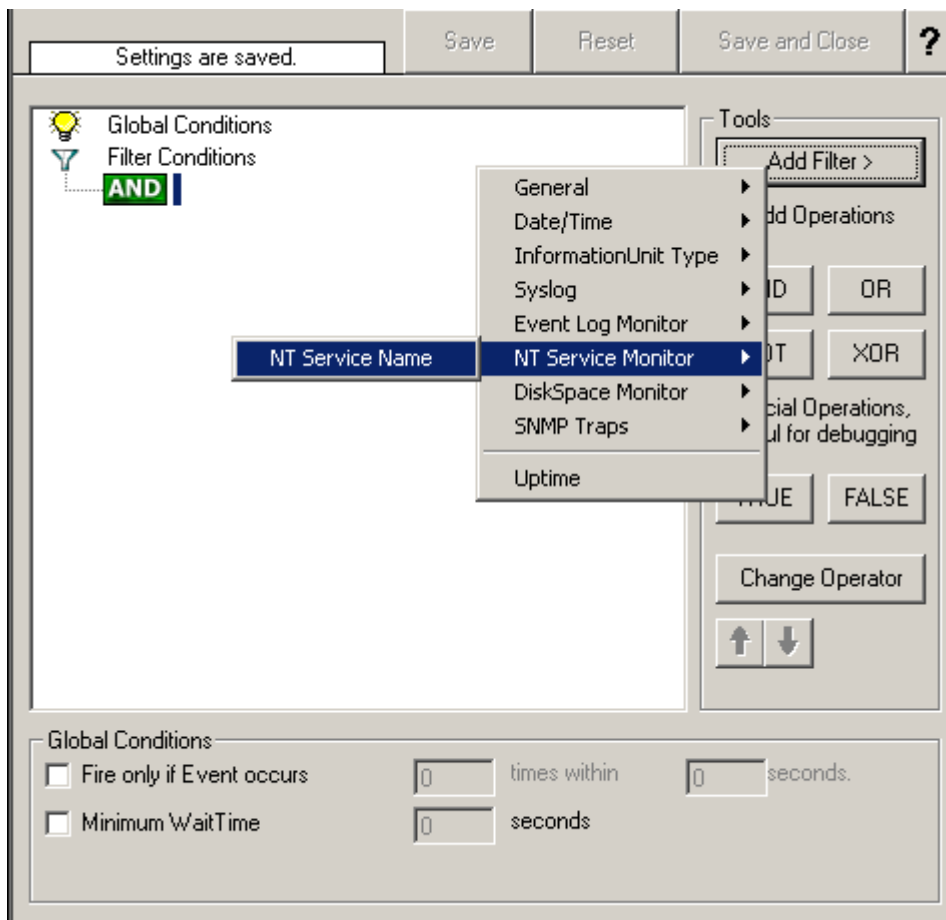
NT Service Name (Type=String)





### 6.4.11 DiskSpace Monitor

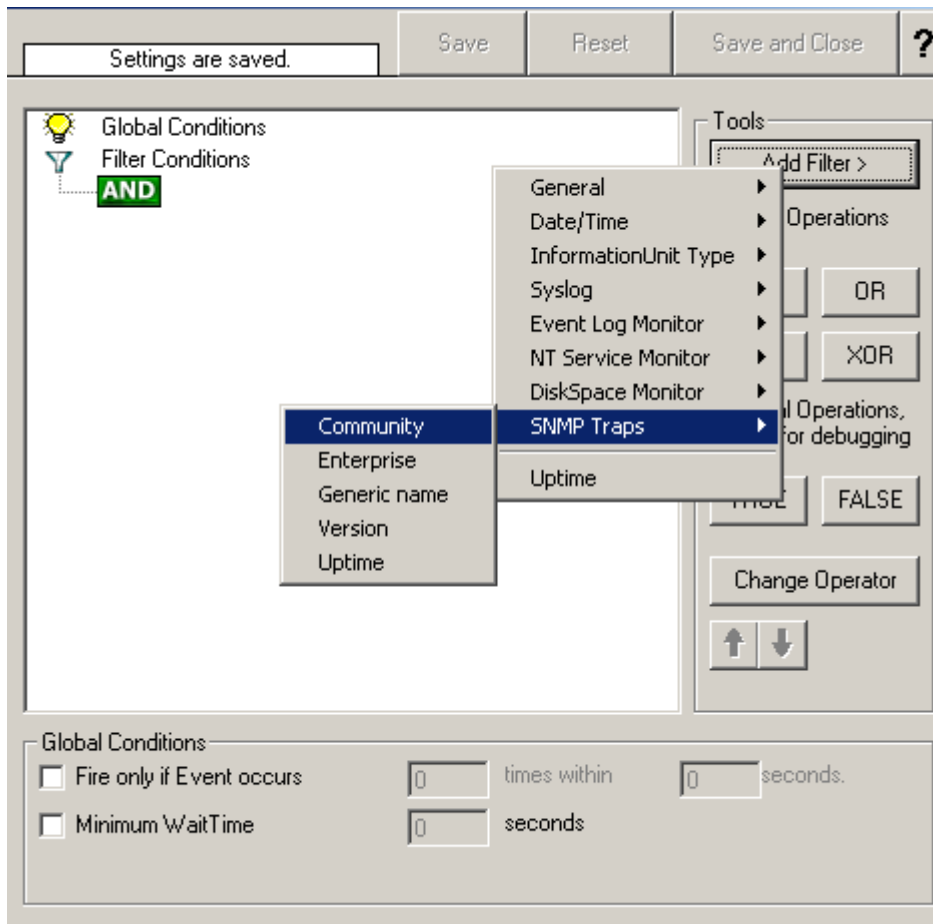
NT Service Name (Type=String)



### 6.4.12 SNMP Traps

Using SNMP Traps MonitorWare Agent 2.0 now can be used to manage and monitor all sorts of equipment including computers, routers, wiring hubs, toasters and jukeboxes.

A trap is generated when the device feels it should do so and it contains the information that the device feels should be transmitted.



## Community

It corresponds to the respective SNMP entity.  
This filter is of type string.

## Enterprise

It corresponds to the respective SNMP entity.  
This filter is of type string.

## Generic name

It corresponds to the respective SNMP entity.  
This filter is of type string.

## Version

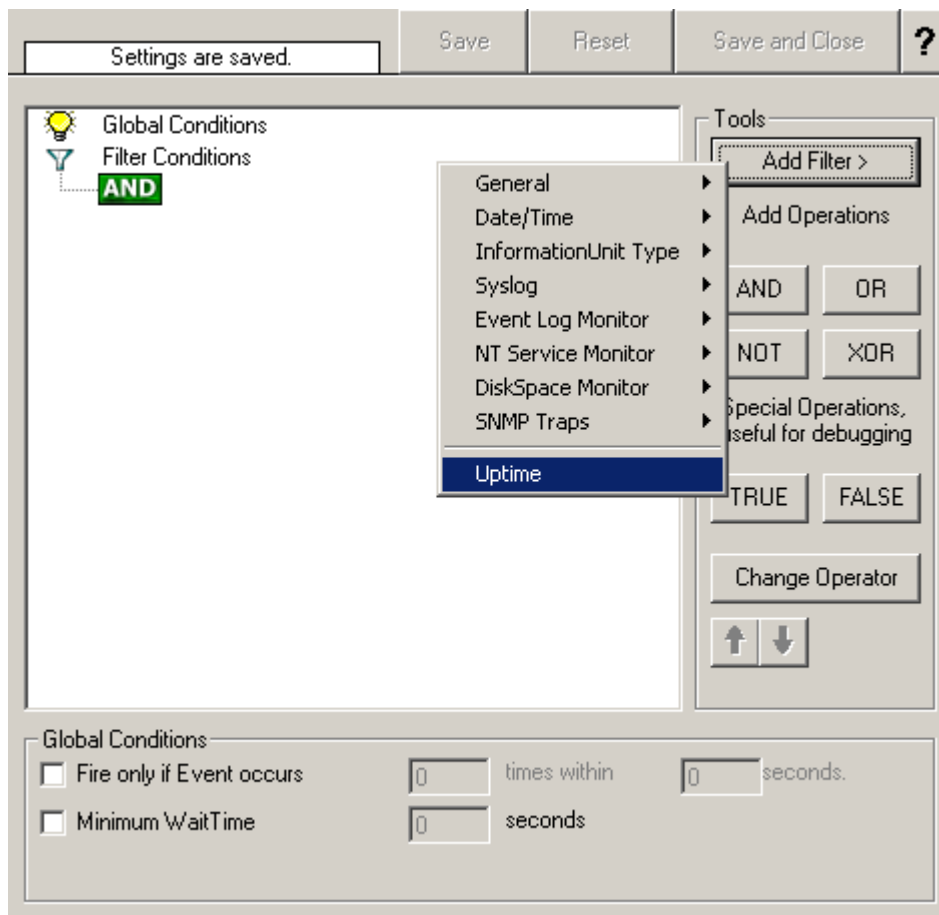
It corresponds to the respective SNMP entity.  
This filter is of type number.

## Uptime

It corresponds to the respective SNMP entity.  
This filter is of type string.

### 6.4.13 Uptime

Uptime (Type= String)



## 6.5 Actions

### 6.5.1 Understanding Actions

Actions tell the product what to do with a given event. With actions, you can forward events to a mail recipient or syslog server, store it in a file or database or do many other things with it.

There can be multiple actions for each rule. Actions are processed in the order they are listed.

### 6.5.2 File Options

This configuration dialog is available both in the defaults section as well as with file logging actions.

File logging is used to write text files of received messages. One file per day is written. New entries are appended to the end of the file.

File locks are released when currently no data is written. Therefore, other applications can access the files while the service is running. However, please be sure that the other applications do not place a file-lock onto it. Popular WordPad does so. In this case, the service will not be able to log any further messages (an error event is written to the NT event log in this case). We recommend copying the file when accessing it at runtime - or use notepad.exe, which does not place file-locks on the files it opens.

#### **The filename is build as follows:**

<FilePathName><FileBaseName>-year-month-day.<FileExtension>

With the parameters in brackets being configured via the dialog.

The screenshot shows a configuration window for MonitorWare Agent. At the top, there is a status bar that says "Settings are saved." and three buttons: "Save", "Reset", and "Save and Close". A help icon (?) is also present. The main area is divided into two sections: "Filename related options" and "General file options".

**Filename related options:**

- File Path Name: C:\temp (with a "Browse" button)
- File Base Name: logdata
- File Extension: log
- File format: Adiscon (dropdown menu)
- Custom Line Format: %msg%%\$CRLF% (with an "Insert" button)
- Create unique filenames
- Use Circular Logging
- Include Source in Filename
- Number of Logfiles: 10
- Use UTC in Filename
- Maximum Filesize (KB): 4096

**General file options:**

- Use XML to Report
- Use UTC for Timestamps
- Include Date and Time
- Include Date and Time reported by Device
- Include Syslog Facility
- Include Syslog Priority
- Include Source
- Include Message
- Include RAW Message

*File Logging Options*

## Create unique Filenames

If checked, MonitorWare Agent 2.0 will create a unique file name for each day. This is done by adding the current date to the base name (as can be seen above).

If left unchecked, the date is not added and as such, there will be a single file, consistent file name. Some customers that have custom scripts to look at the file name use this.

## Use UTC in Filename

This works together with the "Create unique Filenames" setting. If unique names are to be created, the "Use UTC in Filename" selects if the file name is generated based on universal coordinated time (UTC) or on local time. UTC was formerly referred to as

"GMT" and is the basis of the time zone system. For example, New York, USA is 5 hours behind UTC. Therefore, if it is 12 noon in New York, the UTC time is 5pm.

When it comes to log file creation, it means that the date is computed on UTC. Taking the same example, if the "Use UTC in Filename" is checked, the log file name would roll over to the next date at 7pm New York time. If it were unchecked, the rollover would occur exactly at midnight New York time (5am UTC).

Using UTC for file name creation can be helpful if log files are written among different time zones and later consolidated. Using UTC ensures a consistent time notation across all log files.

Please note that this setting does affect the file name creation only. A different setting controls the dates recorded inside the file.

### **File Path Name**

The base path (directory) of the file. Please see above for exact placement. Default is "c:\temp".

### **File Base Name**

The base name of the file. This is the part before the date specific information. Please see above for exact placement.

### **File Extension**

The extension to be used when writing the file. Please see above for exact placement. Default is ".log".

### **File Format**

This controls the format that the log file is written in. The default is "Adiscon", which offers most options. Other formats are available to increase log file compatibility to third party applications.

The "Raw Syslog message" formats writes raw Syslog format to the log file. That is, each line contains the Syslog message as of RFC3164. No specific field processing or information adding is done. Some third party applications require that format.

The "WebTrends Syslog compatible" mimics the format that WebTrends applications expect. Please note that we only mimic the log file format. It is still the job of the reporting device (most notable firewall) to generate the correct WebTrends WELF format. The "WebTrends" format is supported because many customers would like to

use MonitorWare Agent 2.0 enhanced features while still having the ability to work with WebTrends.

Please note that any other format besides "Adiscon Default" is a fixed format. As such, if it is selected, all other formatting options do not apply and consequently are turned off.

### **Include Source in Filename**

If checked, the file name generation explained above is modified. The source of the Syslog message will be automatically added to the file name.

This feature has been introduced because many customers would like to have separate log files for each device. While this can be achieved with multiple rules, it is much more straightforward with this single checkbox. If it is checked, the messages are automatically written to separate files and the file name includes the originating device information.

### **Use XML to Report**

If checked, the message part includes a complete XML-formatted information record. It includes additional information like timestamps, Syslog facility and priority and others in an easy to parse format. If XML output format is selected, you might consider turning all other information fields off, as they are already included in the XML stream. However, this is not a requirement.

### **Use UTC for Timestamps**

Please see the definition of UTC above at "Use UTC in Filename". This setting is very similar. If checked, all time stamps will be written in UTC. If unchecked, local time will be used instead. Again, UTC is useful if logs written in multiple time zones are to be consolidated.

### **Include <Fieldname>**

The various "include" settings control which fields are written to the log file. All fields except the message part itself are optional. If a field is checked, it will be written to the log file. If unchecked, it will not be written. All fields are comma-delimited.

Please note the difference between the "Date and Time" and "Date and Time reported by Device". Both are timestamps. Either both are written in local time or UTC based on the "Use UTC for Timestamps" check box. However, "Date and Time" is MonitorWare Agent 2.0 received the time the message. Therefore, it always is a consistent value.

In contrast, the "Date and Time Reported by Device" is a timestamp taken from the actual message. As such, it is dependent on the reporting device clock, which might be off. In addition, in the case of Syslog messages, there is no time zone information within the device reported timestamp. As such, if devices from multiple time zones are reporting, the timestamp information is not consistent. This is due to Syslog design as of RFC 3164. The Syslog server can be configured to ignore the RFC in this case and provide a consistent time stamp. However, from the view of the log file writer, the "Date and Time Reported by Device" might not be as trustworthy as the "Date and Time" field. Nevertheless, it might also be more useful than the former one. This is the reason both timestamps are present and can individually be selected.

The "Include Message" and "Include RAW Message" fields allow customizing the message part that is being written. The raw message is the message as MonitorWare Agent 2.0 – totally unmodified, received it. This might be useful if a third party application is expecting raw Syslog entries. The message itself is just that part of the Syslog message that is being parsed as message that is without e.g. host information or a tag value. Please note that we recommend selecting only one of these options, as otherwise two message fields will be written. Similarly, if none is selected no message is written at all. Please note that we support these configurations, too – there might be a legitimate need for them.

### 6.5.3 Database Options

Database logging allows persisting all incoming messages to a database. Once they are stored inside the database, different message viewers as well as custom applications can easily browse them.



Settings are saved. Save Reset Save and Close ?

DSN  Data Sources (ODBC) Create Database

User-ID  Password

Table Name   Enable Encryption

Output Encoding

Table Field Names

General Fields	EventReport Specific Fields
Device Reported Time <input type="text" value="DeviceReportedTime"/> <input type="text" value="Localtime"/>	NTSeverity <input type="text" value="NTSeverity"/>
ReceivedAt <input type="text" value="ReceivedAt"/> <input type="text" value="UTC"/>	EventSource <input type="text" value="EventSource"/>
FromHost <input type="text" value="FromHost"/>	EventUser <input type="text" value="EventUser"/>
Message <input type="text" value="Message"/>	EventCategory <input type="text" value="EventCategory"/>
Importance <input type="text" value="Importance"/>	EventID <input type="text" value="EventID"/>
CustomerID <input type="text" value="CustomerID"/>	EventBinaryData <input type="text" value="EventBinaryData"/>
SystemID <input type="text" value="SystemID"/>	NTEventLogType <input type="text" value="EventLogType"/>
InfoUnitID <input type="text" value="InfoUnitID"/>	
Syslog Specific Fields	DispSpace Monitor Fields
Facility <input type="text" value="Facility"/>	MaxAvailable <input type="text" value="MaxAvailable"/>
Priority <input type="text" value="Priority"/>	CurrUsage <input type="text" value="CurrUsage"/>
SysLogTag <input type="text" value="SysLogTag"/>	File Monitor Fields
	GenericFileName <input type="text" value="GenericFileName"/>

### Database Logging Options

Database logging allows writing incoming events directly to any ODBC-compliant database (virtually any database system currently available for the Windows operating system supports ODBC). Adiscon directly supports Microsoft JET databases (as used by Microsoft Access) and Microsoft SQL Server. We also know of many customers who run it successfully with Oracle and Sybase as well as a variety of other systems.

## DSN

This is the name of the system data source (DSN - data source name) to be used when connecting to the database. Create this in ODBC manager (can be found in control panel under Windows NT). Press the "Data Sources (ODBC)" button to start the operating system ODBC Administrator where data sources can be added, edited and removed.

**Important:** The DSN must be a system DSN, not a user or file DSN. The DSN must be

configured to have the correct connection parameters (for example database type and name, server name, authentication mode, etc.).

## User-ID

The user id used to connect to the database. It is dependant on the database system used if it must be specified (e.g., Microsoft Access does not need one, while Microsoft SQL Server can force you to use one). If in doubt, please see your database administrator.

## Password

The password used to connect to the database. It must match the "User ID". Like the user id, it is dependant on the database system if a password is needed. Passwords can be stored either encrypted or unencrypted. We highly recommend storing them encrypted.

## Enable Encryption

Check to store the ODBC password encrypted. If left unchecked, the password is stored unencrypted. We strongly recommend checking this box.

If you store the password unencrypted for some reason, please be aware of the security implications. In this case, we recommend using an account with limited access privileges, only. Even when stored encrypted, we recommend using limited privileges accounts. We are not applying very strong cryptography here.

## Table Name

The name of the table to log to. This name is used to create the SQL insert statement and must match the database definition. Default is "SystemEvents".

**Please note that the default table name must be used when other members of the MonitorWare family (like the web interface or the MonitorWare Console) should work with the database. This customization option is meant for those customers that use third-party or custom software.**

## Table Field Names

These settings allow overriding the default field names to be used when storing data into the system events table. The field names can be changed to any name as long as that name is a valid database field (column) name. However, all fields need to be

present. Otherwise, the ODBC writer will fail.

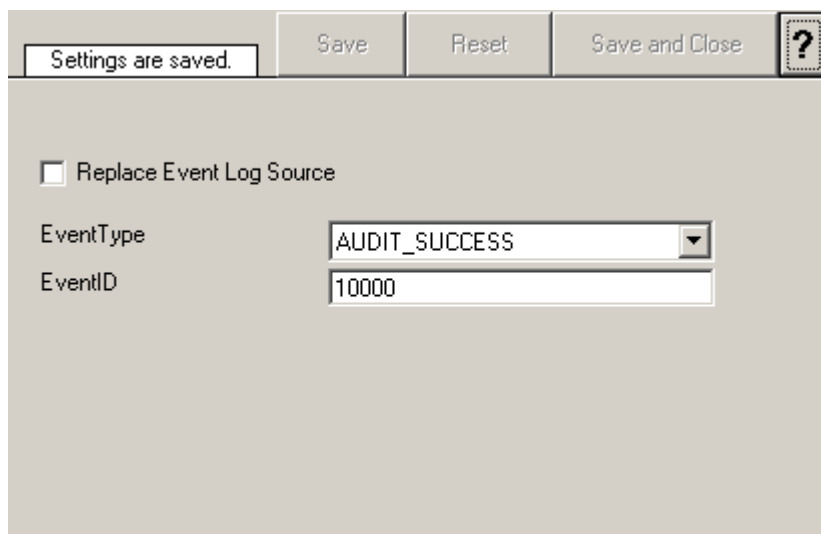
**Please note that the default field names must be used when other members of the MonitorWare family (like the web interface or the MonitorWare Console) should work with the database. This customization option is meant for those customers that use third-party or custom software.**  
**Important**

The default name for the message field - "Message" is a reserved name on Sybase database systems. If you would like to log to a Sybase database, you must change that field name. Otherwise, you will receive an ODBC error (visible in NT Event Viewer). We are unfortunately not able to change the default, as this would break many existing logging environments that migrate from WinSyslog to MonitorWare Agent 2.0.

The database conforms to the Common MonitorWare Database Format.

#### 6.5.4 Event Log options

This tab is used to configure the logging to the Windows NT / 2000 or XP event log. It is primarily included for legacy purposes.



*Event Logging Options*

#### Replace Event Log Source

If checked, a special mapping mechanism is activated. In this mode, the Windows

event source is set to the IP address of the system sending the Syslog message. In addition, the ID is set to Syslog facility. This mode helps to quickly gather information about the system state in Windows event viewer.

**However, this mode has its drawbacks.** Effectively, we are writing invalid event source information to the event log. This does not harm any application, but Windows event viewer will try to locate the matching message libraries. Of course, this is impossible. As such, event viewer will warn the user that the message library could not be found. Nevertheless, it will display the complete logged message. This happens only in detail view.

Users should fully understand the implications of this mapping mechanism for their environment before turning this option on.

## EventType

The type – or severity – this log entry is written with. Select from the available Windows system values.

## EventID

The ID to be used when writing to the event log. Different IDs can be used to provide other processes with a consistent interface to specific messages. WinSyslog does not restrict the IDs that can be used. However, if an ID is written that is not registered with the operating system, Windows Event Viewer places an error message pointing this out before the actual message text. To avoid this text, event IDs 10,000 to 10,100 have been registered with the OS. We highly recommend that these IDs be used for all custom messages. IDs below 10,000 should not be used as they might potentially interfere with events generated by MonitorWare Agent 2.0 itself.

### 6.5.5 Mail Options

This tab is used to configure mail (SMTP) parameters. These here are the basic parameters for email forwarding. They need to be configured correctly if mail message should be sent by the service

Settings are saved. Save Reset Save and Close ?

Mailservers 127.0.0.1

Port 25

Sender Your@Sender

Recipient Your@Recipient

Subject Email for you

Use legacy subject line processing [Insert](#)

Mail Message Format

```
Event message:
Facility: %syslogfacility%
Priority: %syslogpriority%
Source: %source%

Message:
%msg%
```

Session Timeout (0 - 4000 ms) 0

Output Encoding System Default

Use SMTP Authentication

SMTP Username

SMTP Password

Include message / event in email body

Use XML to Report

*Forward Email Properties*

## Mailserver

This is the Name or IP address of the mail server to be used for forwarding messages. Please note that this server must be able to relay messages if the recipient is not hosted at this server. Be sure to contact your mail server's administrator if in doubt on this issue.

The service expects to talk to a standard SMTP mail server. Message relaying to the final destination must be permitted.

## Port

Port the mail server is to be contacted at. Usually, this is 25. It might, however, be changed by in your system. Then, specify the port your mail server uses. If in doubt, try the default of 25 - or contact your mail server administrator.

## **Sender**

Email address used as the sender address for outgoing messages. In order for your SMTP server to accept it, it probably must be a valid address.

## **Recipient**

The recipient emails are addressed to. If multiple recipients are to receive an email via a single "Send Email" action, a server distribution list must be supported. Alternatively, multiple "Send Email" actions can be defined, each one with another recipient.

## **Subject**

Subject line to be used for outgoing emails. The subject line is used for each message sent. It can contain replacement characters or event properties to customize it with event details. This is especially useful when sending email to cellular phones or pagers, which often display only the subject line and not the actual message body. The subject line – after expansion of the any replacement sequences – can hold a maximum of 255 characters. Characters beyond this will be truncated. Please note that many email systems impose a stricter limit and truncation as such may occur before the 255-character limit. It is best to try to limit the subject line length to 80 characters or less.

The mail body will also include full event information, including the source system, facility, priority and actual message text as well as any other information that came with this event. As there is no size limitation for message bodies, the body always contains the full message received (except otherwise configured – see below).

There will be one email for each received message. Email delivery is meant for urgent notifications and actions (e. g. calling pagers and such). It is not meant to provide an email report.

## **Use legacy subject line processing**

This checkbox specifies which type of subject line processing will be done. If it is checked, the old-style processing using single character replacement sequences is applied. If it is left unchecked, the far more powerfull event property based method is used.

**In legacy mode**, the following replacement characters are recognized inside the subject line:

<b>%s</b>	IP address or name (depending on the "resolve hostnames" setting) of the source system that sent the message.
<b>%f</b>	numeric facility code of the received message
<b>%p</b>	numeric priority code of the received message
<b>%m</b>	the message itself. Please note: this is the complete message text and can be rather lengthy. As such, it is most probably subject to truncation. If that occurs, all other information after the %m replacement character is also truncated. As such, we strongly recommend using the %m replacement at the end of the subject line only.
<b>%%</b>	represents a single % sign.

As an example, you may have the subject line set to "Event from %s: "m" and enabled legacy processing. If a message "This is a test" were received from "172.16.0.1", the resulting email subject would read: "Event from 172.16.0.1: This is a test"

**In non-legacy mode**, the Property Replacer can be used. With it, you can include any property from the event message and also modify it. Please visit the Property Replacer documentation for details.

As an example, in non-legacy mode, you can set the subject line to "Msg: '%msg:1:15%' From: %fromhost%". If the message "This is a lengthy test message" were received from "172.16.0.1", the resulting email subject would read: "Msg: 'This is a lengt' From: 172.16.0.1". Please note that the message is truncated because you only extracted the first 15 characters from the message text (position 1 to 15).

## Mail Message Format

This is the format of the message body. Properties from the event can be included by using the Property Replacer. Please note that the message body is only sent if "[Include Message/Event in Email Body](#)" is checked.

## Session Timeout

This option controls if multiple rapidly incoming messages should be combined to a single email message. The SMTP session with the server is held open for the specified timeout period. Please note that the period is specified in milliseconds, not seconds.

If a new event arrives within the specified timeout period, that event will be included in the same email message as the previous one. Then, the timeout is re-started. As such, any events coming in within successive timeout periods will be combined in a single mail.

This is most appropriate when large burst of messages are expected and these should

be combined in few mail messages. Otherwise, multiple mail messages can easily overflow the administrator's mailbox.

The session timeout is user configurable between 0 and 4000 milliseconds. Larger values are not supported as they probably affect the SMTP server performance and can lead to unpredictable results.

The session timeout of zero milliseconds has a special meaning: if it is selected, every event will be sent in a separate message, no matter how fast two messages occur after each other.

## Use SMTP Authentication

Check this box if your server requires SMTP authentication. To fight SPAM, more and more server operators allow relaying only for authenticated users. It might even happen that an existing account does no longer work because the server has been reconfigured to disallow anonymous posting.

If your server requires (or supports) SMTP authentication, check this box and enter your User ID and password in the boxes below. The exact values will be provided by your server operator – if in doubt, please ask the mail server support.

If the mail server does not support authentication, leave this box unchecked.

We recommend using authentication if it is available. Even when the current server configuration allows unauthenticated relay, this potentially will change in the future (as the SPAM problem grows). If you already use authentication, such a server configuration change will not affect you. Otherwise, it will disrupt mail service.

## Include message / event in email body

This checkbox controls whether the Syslog message will be included in the message body or not. If left unchecked, it will **not** be included in the body. If checked, it will be sent.

This option is useful for pagers and mobile phones, especially those with WML support. These devices are often capable of displaying only limited amounts of data. Some do not display the message body at all. As such, it makes limited sense to send a message body. As such, it can be turned off with this option. With these devices, use a subject line with the proper replacement characters .

Even if your WML enabled phone supports receiving message bodies, it might be a good idea to turn them off. WML and WAP are relatively expensive. Generated messages can become lengthy (depending on the message source). As such, it might be appropriate to disable the message body in such a scenario.

This option is must useful together with a well-formatted subject line in [non-legacy mode](#).



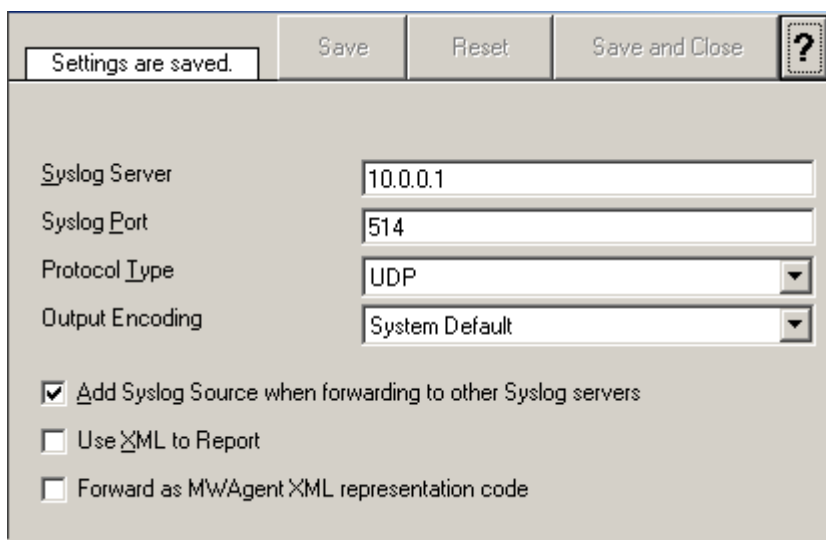
## Use XML to Report

If checked, the received event will be included in XML format in the mail. If so, the event will include **all** information, like the original timestamp, the facility, priority etc. XML format is especially useful if the mail is sent to an automated system, which will then parse the message.

If unchecked, just the plain text message will be included in the mail. This format is more readable for a human reader.

### 6.5.6 Forward Syslog Options

This dialog controls Syslog forwarding options.



Settings are saved. Save Reset Save and Close ?

Syslog Server: 10.0.0.1

Syslog Port: 514

Protocol Type: UDP

Output Encoding: System Default

Add Syslog Source when forwarding to other Syslog servers

Use XML to Report

Forward as MWAagent XML representation code

*Forward Syslog Properties*

### Syslog Server

This is the name or IP address of the systems Syslog messages should be sent to.

### Syslog Port

The remote port on the Syslog server to report to. If in doubt, please leave it at the default of 514, which is typically the Syslog port. Different values are only required for special setups, for example in security sensitive areas.

## Protocol Type

Syslog messages can be received via UDP, TCP or RFC3195RAW. One listener can only listen to one of the protocols. Typically, Syslog messages are received via UDP protocol, which is the default. MonitorWare Agent 2.0 also can receive Syslog messages via TCP and reliable Syslog messages via TCP using the new RFC 3195 standard.

## Output Encoding

This setting is most important for Asian languages. A good rule is to leave it at "System Default" unless you definitely know you need a separate encoding. "System Default" works perfect in the far majority of cases, even on Asian (e.g. Japanese) Windows versions.

## Add Syslog Source

If this box is checked, information on the original originating system is prepended to the actual message text. This allows the recipient to track where the message originally came from.

**Please note:** This option is not compatible with RFC 3164. We recommend selecting it primarily when message forwarding to a WinSyslog Interactive Server is intended.

## Use XML to Report

If checked, the forwarded Syslog message is a complete XML-formatted information record. It includes additional information like timestamps or originating system in an easy to parse format.

The XML formatted message is especially useful if the receiving system is capable of parsing XML data. However, it might also be useful to a human reader as it includes additional information that cannot be transferred otherwise.

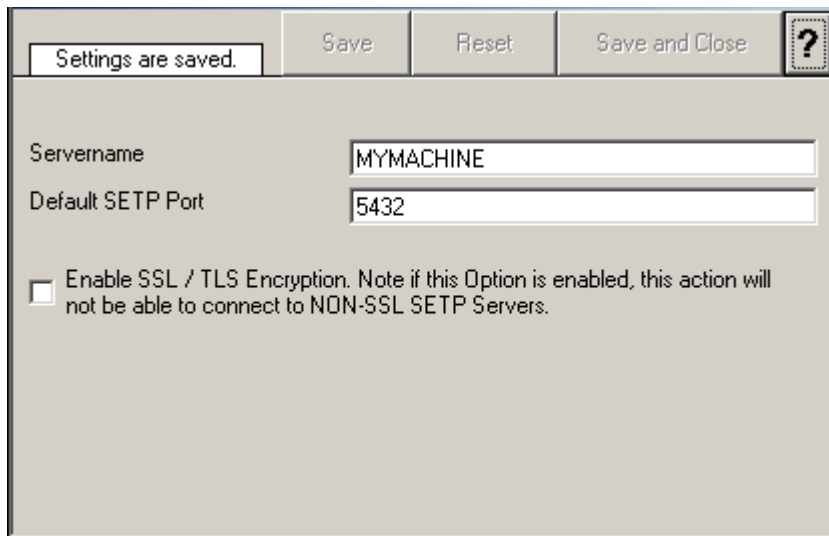
## Forward as MW Agent XML Representation Code

MonitorWare supports a specific XML-Representation of the event. If it is checked, that XML representation is used. It provides additional information (like InformationUnit Type, original source system, reception time & many more) but is harder to read by a human. At the same time, it is obviously easier to parse.

## 6.5.7 Forward SETP Options

This dialog controls the Send options.

With the "Send SETP" action, messages can be sent to a SETP server.



Settings are saved. Save Reset Save and Close ?

Servername MYMACHINE

Default SETP Port 5432

Enable SSL / TLS Encryption. Note if this Option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

*Send SETP Dialog*

### Servername

The MonitorWare Agent 2.0 sends SETP to the server/listener under this name.

### Default SETP Port

The Send SETP sends outgoing requests on this port. The default value is 5432.

**Please note:** The SETP port configured here **must** match the port configured at the listener side (i.e. MonitorWare Agent 2.0 or WinSyslog Enterprise edition). If they do not match, a Send SETP session cannot be initiated. The rule engine will log this to the NT Event Log.

### Enable SSL/TLS

If this option is enabled then this action will be able to connect to SSL/TLS SETP servers. Please make sure that you want this option to be enabled.

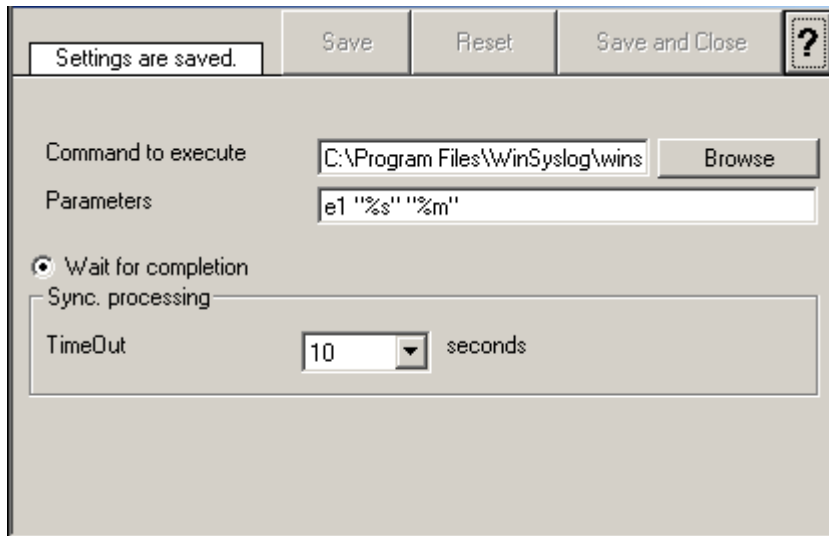
**Please note:** If this option is enabled, this action will not be able to connect to NON-SSL SETP Servers.

### 6.5.8 Start Program

This dialog controls the start program options.

With the "Start Program" action, an external program can be run. Any valid Windows executable can be run. This includes actual programs (EXE files) as well as scripts like batch files (.BAT) or VB scripts (.vbs).

Start Program can, for example, be combined with the service monitor to restart failed services. Another example application is a script that deletes temporary files if the disk space monitor detects a low space condition.



*Start Program Dialog*

#### Program to execute

This is the actual program file to be executed. This can be any valid executable file. A relative file name can be specified if it can be found via the operating system default search path.

#### Parameters

These parameters are passed to the program executed. They are passed as command line parameters. There is no specific format – it is up to the script to interpret them.

Parameters can contain replacement characters to customize it with event details. This allows passing event data to the script. The following replacement characters can be used:

<b>%d</b>	date and time in local time
<b>%s</b>	IP address or name (depending on the "resolve hostnames" setting) of the source system that sent the message.
<b>%f</b>	numeric facility code of the received message
<b>%p</b>	numeric priority code of the received message
<b>%m</b>	the message itself
<b>%%</b>	represents a single % sign.

In the example above, replacement characters are being used. If a message "This is a test" were received from "172.16.0.1", the script would be started with 3 parameters:

Parameter 1 would be the string "e1" – it is assumed that this has some meaning to the script. Parameter 2 would be the IP address, 172.16.0.1. Parameter 3 would be "This is a test". Please note that due to the two quotes ("), the message is interpreted as a single parameters. If they were missing, it would typically be split into several ones, with parameter 3 being "This", 4 being "is" and so on. So these quotes are very important!

## Time Out

When a program is executed, the service waits for it to finish before it carries on further actions. This is needed in order to ensure that all actions are carried out in the correct sequence.

The external program should only run for a limited amount of time. If it would block for some reason, the agent would be prevented from carrying out any further processing. As such, a timeout value must be specified. If the program still runs after the configured timeout, the rule engine cancels it, flags the action as unsuccessful and then carries on with processing.

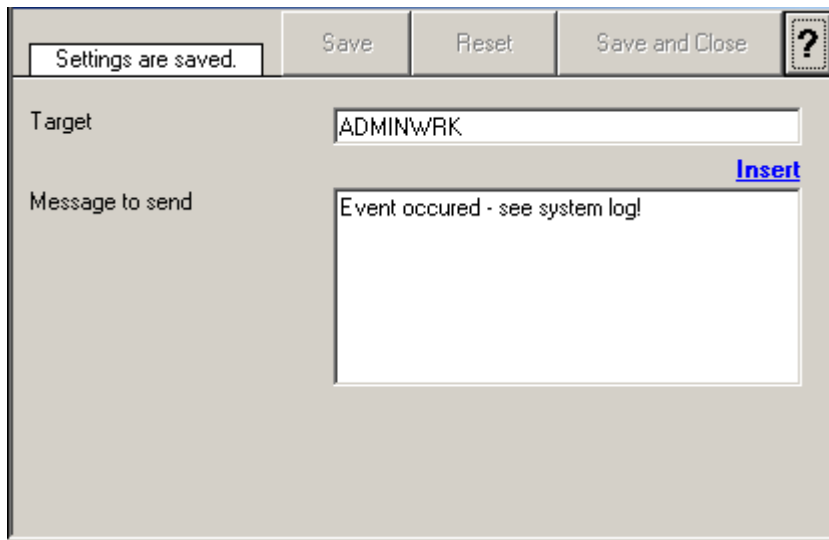
**Important:** Even though the timeout value can be as high as 30 seconds, we strongly recommend limiting the run time of external program to below 5 seconds. Otherwise, they could affect the overall performance too much. If the average run time is 5 seconds, the default timeout of 10 seconds ensures that the program can finish even when there is high system activity.

For performance reasons, we also strongly recommend to use the "Start Program" action only for rules that apply relatively seldom.

### 6.5.9 Net Send

This dialog controls the net send options.

With the "Net Send" action, short alert messages can be sent via the Windows "net send" facility. These messages are delivered on a best-effort basis. If the recipient can be reached, they will pop up in a message box on the recipient's machine. If the recipient cannot be reached, they will simply be discarded. No buffering takes place. Consequently, the rule engine does not check if the message can be delivered. It will never flag an action to be in error due to a reported delivery problem with "net send".



The screenshot shows a dialog box titled "Net Send Dialog". At the top, there is a status bar with a message "Settings are saved." and three buttons: "Save", "Reset", and "Save and Close". A help icon (?) is also present. The main area of the dialog is divided into two sections. The first section is labeled "Target" and contains a text box with the value "ADMINWRK" and an "Insert" button. The second section is labeled "Message to send" and contains a text box with the value "Event occurred - see system log!".

*Net Send Dialog*

#### Target

This is the Windows user name of the intended recipient, a NETBIOS machine name or even an IP address (in the form of 10.1.1.1)

#### Message to Send

This is the message that is sent to the intended target.

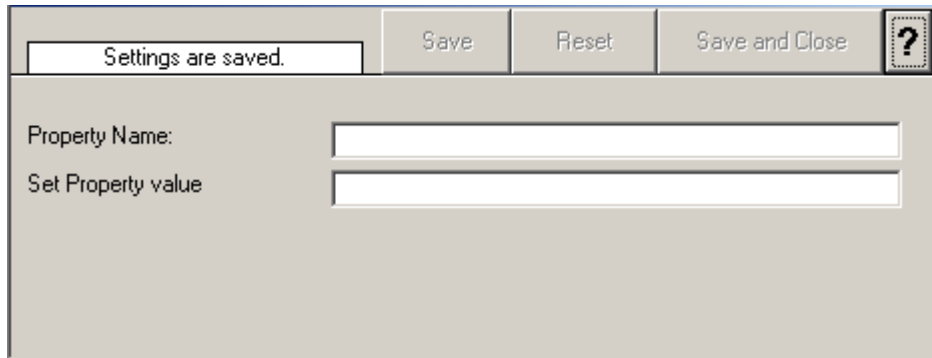
### 6.5.10 Set Status

This dialog controls the set status options.

There is an internal Status List within MonitorWare Agent 2.0 which you can use for more complex filtering.

You can set property over the Set Status action and you can define filter for them.

Please note: when you change a property, the value will be changed as soon as the set status action is carried out. It will not change before that happens and the old value is no longer available thereafter. That means all actions and filter conditions will use the new value after it is set. So if you would like e.g. rename a system, make sure the set status actions are at the top of the rule base!



The screenshot shows a dialog box titled "Set Status Dialog". At the top, there is a status bar with the text "Settings are saved." and three buttons: "Save", "Reset", and "Save and Close". To the right of these buttons is a help icon (a question mark in a square). Below the status bar, there are two input fields. The first is labeled "Property Name:" and the second is labeled "Set Property value".

*Set Status Dialog*

## Property Name

Select the property name to be changed.

## Set Property Type

The new type to be assigned to the property. Any valid property type can be entered.

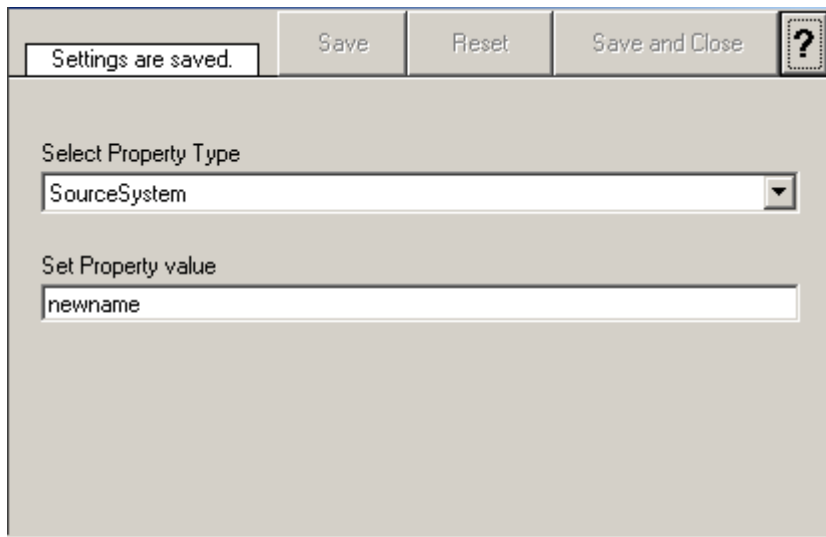
That name will from now on be used inside the rule base. More precisely, it will be used in the filter conditions and actions.

### 6.5.11 Set Property

This dialog controls the set property options.

With the "Set Property" action, some properties of the incoming message can be modified. This is especially useful if an administrator would like to e.g. rename two equally named devices.

Please note: when you change a property, the value will be changed as soon as the set property action is carried out. It will not change before that happens and the old value is no longer available thereafter. That means all actions and filter conditions will use the new value after it is set. So if you would like e.g. rename a system, make sure the set property actions are at the top of the rule base!



The screenshot shows a dialog box titled "Set Property Dialog". At the top, there is a status bar with the text "Settings are saved." and three buttons: "Save", "Reset", and "Save and Close". To the right of these buttons is a help icon (a question mark in a square). Below the status bar, the dialog is divided into two sections. The first section is labeled "Select Property Type" and contains a dropdown menu with "SourceSystem" selected. The second section is labeled "Set Property value" and contains a text input field with the text "newname" entered.

*Set Property Dialog*

### Select Property Type

Select the property type to be changed. The list box contains all properties that can be changed.

### Set Property Value

The new value to be assigned to the property. Any valid property value can be entered.

In the example above, the SourceSystem is overridden with the value "newname". That name will from now on be used inside the rule base. More precisely, it will be used in the filter conditions and actions.

## 6.5.12 Call RuleSet

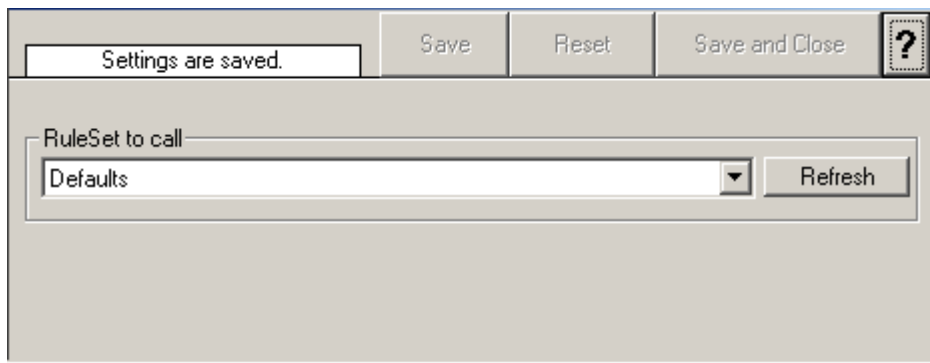
This dialog controls the Call RuleSet options.

A Call RuleSet Action simply calls another Rule Set in some existing Rule Set. When this Action is encountered, the Rule Engine leaves the normal flow and goes to the called Rule Set (which may contain many rules as well). It executes all the rules that have been defined in that called Rule Set. After the execution of all of them, it will return to its point from where it left the original flow. Let's take an example to clarify it a little further.



Let's say that the Action 1 or Rule 1 is an include action. If the Filter Condition result of Rule 1 evaluates to true, it will execute the Action 1. Since Action 1 is the include action in this example, it will go to the included rule set and will execute its Filter Condition. If that filter condition evaluates to true, it will execute all of its actions and will return to Action 2 of Rule 1 (of normal flow) and if on the other hand, the filter condition of the included rule set evaluates to false, it will skip all of its actions and will come back to the Action 2 of Rule 1 (of normal flow).

Note that there is no limit on including the rules which means that a rule that has been included in another rule may contain another rule in it which might contain another rule in it and so on.



*Call Ruleset Dialog*

## Ruleset to Call

Select the Ruleset to be called.

### 6.5.13 Discard

A Discard Action immediately destroys the current Information Unit and any action of any Rule that has been defined after the Discard Action will not be executed at all. When this action is been selected then no dialog appears as nothing needs to be configured for this.

## 7 Getting Help

*The MonitorWare Agent is very reliable. In the event you experience problems, find here how to solve them.*

Please note that all options (except priority support) are also open to evaluating

customers. So do not hesitate to try them. Help is available in English and German language. Our local resellers may provide local language support. Please check with them.

## Frequently asked Questions

For a current list of Frequently Asked Questions (FAQ), please visit <http://www.mwagent.com/en/FAQ/>. The FAQ area is continuously being updated

## MonitorWare Web Site

Visit the support area at [www.mwagent.com/en/support/](http://www.mwagent.com/en/support/) for further information. If for any reason that URL will ever become invalid, please visit [www.adiscon.com](http://www.adiscon.com) for general information.

## Support Forum

Share questions and answers with your peers! The forum is also monitored by Adiscon support staff. To access the forum, point your browser at <http://forum.adiscon.com/viewforum.php?f=1>

## Email

Please address all support requests to [support@adiscon.com](mailto:support@adiscon.com). An appropriate subject line is highly appreciated.

## Online Seminars

Adiscon offers a selection of online seminars. This selection is continuously being expanded. All available seminars can be found at <http://www.adiscon.com/Common/SeminarsOnline/>

**Please note:** Windows Media Player is required to view the seminars.

## Phone

**Phone support is limited to those who purchased support incidents. If you are**

---

**interested in doing so, please email** [info@adiscon.com](mailto:info@adiscon.com) **for further details.**

## **Fax**

Please direct your faxes to

**+49-9349-928820**

**Toll free in the US: 1-888-900-3772**

with "+" being the international dialing prefix, e.g. 011 in the US and 00 in most other countries.

## **Software Maintenance**

Adiscon's software maintenance plan is called UpgradeInsurance. It offers unlimited free upgrades and priority support during its duration. It can be purchased for a period between 1 and 5 years.

To learn more about UpgradeInsurance, please visit

<http://www.adiscon.com/Common/en/products/upgrade-insurance-details.asp>

## **Non-Technical Questions**

Please address all non-technical questions to [info@adiscon.com](mailto:info@adiscon.com). This email alias will answer all non-technical questions like pricing, licensing or volume orders.

## **Product Updates**

The [MonitorWare line of products](#) is being developed since 1996. New versions and enhancements are made available continuously.

Please visit [www.mwagent.com](http://www.mwagent.com) for information about new and updated products.

# **8 MonitorWare Concepts**

MonitorWare Agent offers advanced monitoring capabilities. It can not only monitor the system it is installed on; it can also include information received from Syslog-enabled devices. To fully unleash MonitorWare's power, you need to learn a bit about its concepts. These web resources (provided links) describe each element in detail.

MonitorWare operates on a set of elements. These are

- [Services](#)
- [Information Units](#)
- [Filter Conditions](#)
- [Actions](#)
- [Rules](#)
- [Rule Engine](#)
- [The SETP Protocol](#)

It is vital to understand each element and the way they interact. MonitorWare Agent has multiple and very powerful capabilities. This enables very quick configuration of highly efficient and comprehensive systems. On the other hand, the concepts must be fully understood to make such complex systems really work.

## 9 Purchasing MonitorWare Agent

All MonitorWare Agent features can be used for 30 days after installation without a license. However, after this period a valid license must be purchased. The process is easy and straightforward.

### The License

The end user license agreement is displayed during setup. If you obtained a ZIP file with the product, there is also a file license.txt inside that ZIP file. If you need to receive a copy of the license agreement, please email [info@adiscon.com](mailto:info@adiscon.com).

### Pricing & Ordering

Please visit <http://www.mwagent.com/en/intermediate-order.asp> to obtain pricing information. This form can also be used for placing an order online. If you would like to place a purchase order, please visit <http://www.adiscon.com/Common/en/OrderByPO.asp> to obtain details.

If you would like to receive assistance with your order or need a quote, please contact [info@adiscon.com](mailto:info@adiscon.com).

## 10 Reference

The following references provide in-depth information to some very specific things. You may want to review them if you are looking for one of these. Some references are placed on the web and some other are directly contained in this manual. We decided to provide web-links wherever we considered them useful.

- [The MonitorWare Agent Service](#)
- [Support for Mass Rollouts](#)
- [Formats](#)

- [Version History](#)
- [ICMP Codes](#)
- Property Replacer

## 10.1 Property Replacer

The property replacer is a generic component that allows you to merge properties from the event processed to e.g. the email subject line or a log file line. It is a central component that is used as often in the product as possible. The idea behind the property replacer is that there is often need to specify a value from the event processed.

Events have certain properties. Each of this properties has an assigned name. The properties available depend on the type of event.

Properties are accessed by their name. The property replacer is used within regular text. If a property value should be replaced, the property is specified using this special sequence:

`%property:fromPos:toPos%`

The percent-signs ("%") indicates the start of a special sequence. The other parameters have the following meanings

### Property

This is the name of the property to be replaced. It can be any property that a given event posses. If a property is selected that is empty for the event processed, an empty string is returned.

A property is either an always-present property, an event specific property, a dynamic property or a [system property](#).

### FromPos

If you do not want to use the full string from the property, you can specify a start position here. The first character is at position 1. If not specified, the property string is copied starting at position 1.

### ToPos

If you do not want to sue the full string from the property, you can specify the highest character position to be copied here. If not specified, the ending position will be the last character.

FromPos and ToPos can be used to copy a substring from a lengthy property.

### Examples

A good example for this is the email subject line, which has severe length constraints. If you would like to have only the first 40 characters of the actual message text in the subject, you could use the replacer: "%msg:1:40%".

If you know the first 10 characters of the message are meaningless for you but you would like to see the full rest of the message (no matter how long it may be), you can use a sequence like "%msg:11%".

If you would just like to see the plain message from beginning to end, you can simply omit FromPos and ToPos: "%msg".

Of course, all of these sample not only work with the "msg" property, but also with all others like "facility" or "priority", or W3C-log header extracted property names.

### 10.1.1 System Properties

System properties are special sequences that can be helpful. They are available with all event types. They are:

<b>\$CRLF</b>	A Windows newline sequence consisting in the characters CR and LF. If you just need CR, you can use %\$CRLF:1:1% and if you need use LF you can use "\$CRLF:2:2%
<b>\$TAB</b>	An US-ASCII horizontal tab (HT, 0x09) character
<b>\$HT</b>	same as \$TAB
<b>\$CR</b>	A single US-ASCII CR character (shortcut for %\$CRLF:1:1%)
<b>\$LF</b>	A single US-ASCII LF character (shortcut for %\$CRLF:2:2%)

## 10.2 Complex Filter Conditions

The rule engine uses complex filter conditions.

Powerful boolean operations can be used to build filters as complex as needed. An boolean expression tree is graphically created. The configuration program is modelled after Microsoft Network Monitor. So thankfully, many Administrators are already used to this type of Interface. If you are not familiar with it, however, it looks a bit confusing at first. In this chapter, we are providing some samples of how boolean expressions can be brought into the tree.

### Example 1

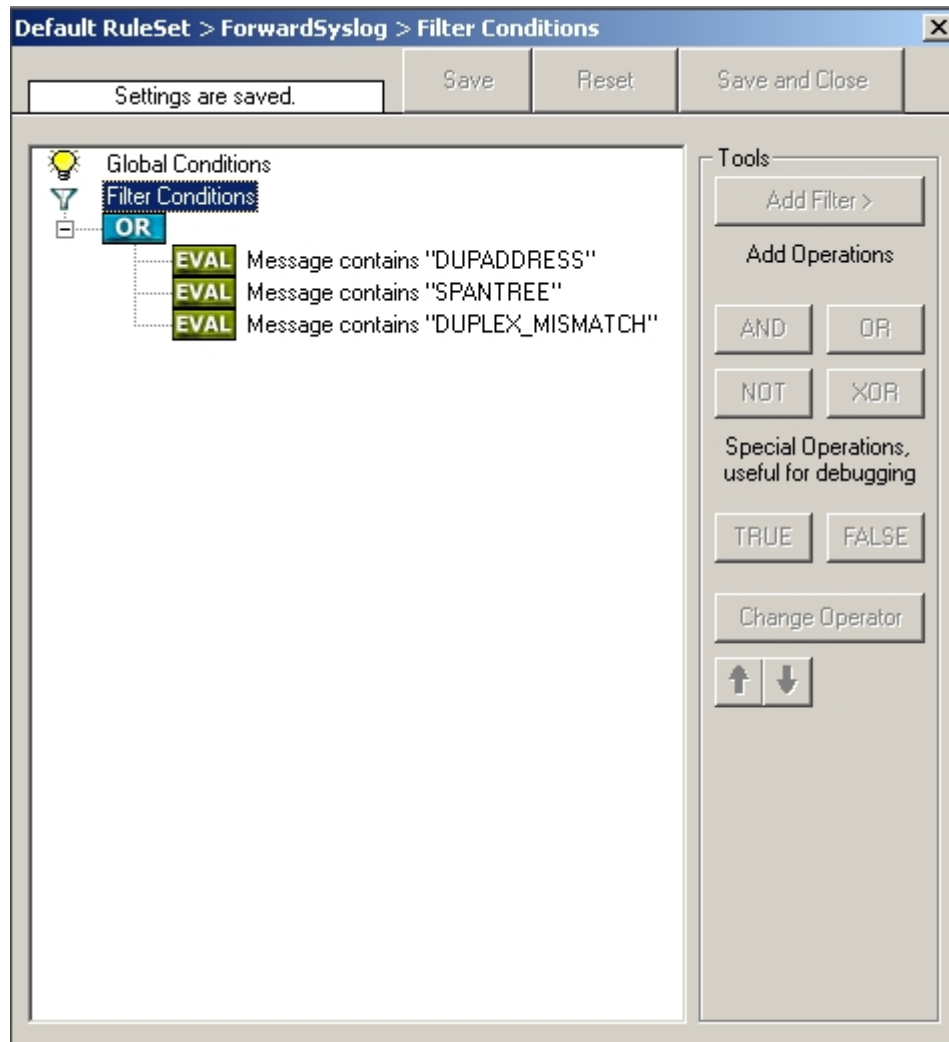
In this example, the message text itself shall be checked. If it contains at least one of three given strings, the filter should become true. If none of the string is found, the boolean expression tree evaluates to false, which means the associated action(s) will not be executed.

In pseudo-code, the filter could be written like this:

```
If (msg = "DUPADDRES") Or (msg = "SPANTREE") Or (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

Please note: in the example, we have abbreviated "message" to just "msg". Also note that for brevity reasons we use the equals ("=") comparison operator, nicht the contains. The difference between the equals and the contains operator is that with "contains", the string must just be part of the message.

In the filter dialog, this pseudo code looks as follows:



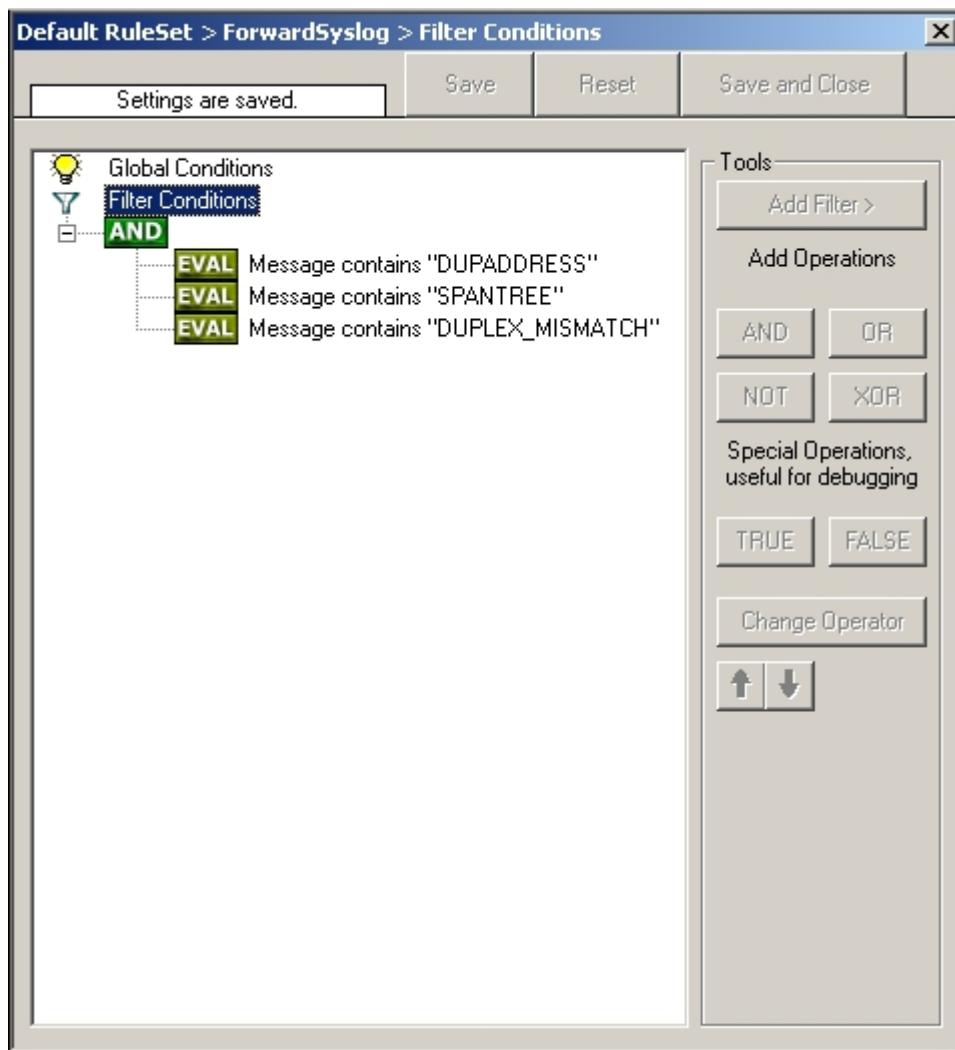
## Example 2

Example 2 is very similar to example 1. Again, the message content is to be checked for three string. This time, **all** of these strings must be present in order for the boolean tree to evaluate to false.

The pseudo code would be as follows (under the same conditions outlined in example 1 above):

```
If (msg = "DUPADDRES") And (msg = "SPANTREE") And (msg = "DUPLEX_MISMATCH) then
    execute action(s)
end if
```

In the filter dialog, this pseudo code looks as follows:



### Example 3

This example is a bit more complex version of example 1. Again, the same message text filtering is done, that is if any one of the provided substrings is present, the filter eventually evaluates to true. To do so, the source system must also contain the string "192.0.2", which can be used to filter on a device from a specific subnet.

An example like this can be used for a rule where the administrator of a specific subnet should be emailed when one of the strings indicate a specific event.

The pseudo code would be as follows (under the same conditions outlined in example 1 above):

```

If ((sourceSys = "192.0.2")
And
((msg = "DUPADDRESS") Or (msg = "SPANTREE") Or (msg = "DUPLICATE_MISMATCH"))
) then

```

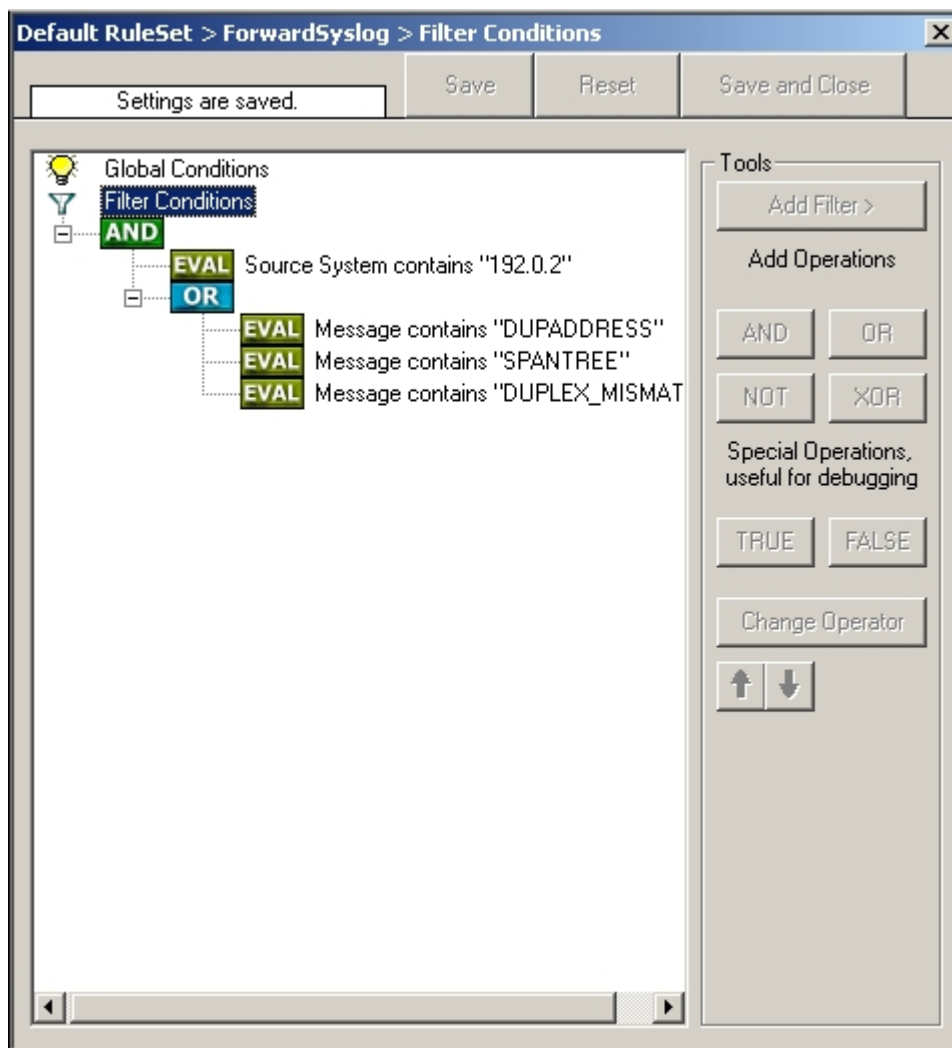


```

execute action(s)
end if

```

In the filter dialog, this pseudo code looks as follows:



As a side note, you may want to use a range check instead of a simple include for the source system. With a range string check, you can specify that the string must be within a specified column range, in this case obviously at the beginning of the source system IP address.

## Real-World Examples

To see some real-world examples of where boolean conditions inside filtering are used, please visit these web links:

- [Detecting Password Attacks under Windows](#)

## 11 Copyrights

This documentation as well as the actual MonitorWare Agent product is copyrighted by Adiscon GmbH, Germany. To learn more about other Adiscon products, please visit <http://www.adiscon.com/en/products>. To obtain information on the complete [MonitorWare product line](#), please visit [www.monitorware.com](http://www.monitorware.com).

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Other mentioned trademarks are for reference only. They belong to their respective owners.

## 12 Glossary of Terms

**The Glossary of Terms is also available on the Web:**

<http://www.adiscon.com/Common/en/glossary/>

The web version most probably has more and more up-to-date content. We highly encourage you to visit the web if in doubt.

### 12.1 EventReporter

EventReporter is Adiscon's solution to forward Windows NT/2000/XP event log entries to central system.

These central systems can be either WinSyslog's, other syslog daemons (e.g. on UNIX) or MonitorWare Agents. EventReporter is part of Adiscon's MonitorWare line of products

More Information about EventReporter:

<http://www.adiscon.com/Common/en/glossary/eventreporter.asp>

### 12.2 Millisecond

A millisecond is a thousand of a second. It is abbreviated as "ms". As such, 500ms mean half a second.

Inside the MonitorWare line of products, many timers are expressed in milliseconds as a fine control over the services and actions is provided to the administrator.

More Information about Milliseconds:

<http://www.adiscon.com/Common/en/glossary/Millisecond.asp>

## 12.3 Monitor Ware Line of Products

Adiscon's MonitorWare line of products includes monitoring and operations management tools. It consists of several components, each of which can be used either individually or as a complete solution. As of this writing, the following products are available:

- Adiscon Logger ([www.monitorware.com/logger/](http://www.monitorware.com/logger/))
- ActiveLogger ([www.activelogger.com](http://www.activelogger.com))
- EventReporter ([www.eventreporter.com](http://www.eventreporter.com))
- IISLogger ([www.iislogger.com](http://www.iislogger.com))
- MoniLog ([www.monilog.com](http://www.monilog.com))
- MonitorWare Agent ([www.monitorware.com](http://www.monitorware.com))
- MonitorWare Console ([www.mwconsole.com](http://www.mwconsole.com))
- WinSyslog ([www.winsyslog.com](http://www.winsyslog.com))

There is also an open source syslog library available for programmers wishing to integrate syslog into their C/C++ programs:

- liblogging ([www.liblogging.org](http://www.liblogging.org))

New products are continuously being added - please be sure to check [www.monitorware.com](http://www.monitorware.com) from time to time for updates.

More Information about the MonitorWare Line of Products:

<http://www.adiscon.com/Common/en/glossary/MonitorWare-Line-of-Products.asp>

## 12.4 Resource ID

The resource ID is an identifier used by the MonitorWare line of products. It is a simple, administrator assigned string value. It can be used to correlate different events - even from different source - to a specific resource.

For example, on a Windows server running Microsoft Exchange, all Exchange events could be assigned to a resource id of "Exchange Server".

In MonitorWare Agent 1.0 and WinSyslog 4.0 support for resource ids is limited. The field is present and can be persisted to the database or stored in XML files, but besides this there is no value in it.

Later releases of the MonitorWare line of products will much broader support the resource id.

More Information about the Resource ID:

<http://www.adiscon.com/Common/en/glossary/Resource-ID.asp>

## 12.5 SETP

SETP is the "Simple Event Transfer Protocol". SETP allows reliable delivery of events between SETP supporting systems. WinSyslog and MonitorWare Agent support SETP. WinSyslog works as SETP client, only. As such, it can forward events generated and gathered by them to central or intermediary SETP servers. The MonitorWare Agent can operate both as a SETP server and client and as such also as a relay. It plays a vital role in a complex, distributed environment.

SETP was developed for MonitorWare. It allows synchronous communication between SETP clients and servers. With SETP, an event can be forwarded exactly as it was on the original event generating system. For example, if a syslog message is received on a remote system, that exact syslog message can be forwarded via as many SETP relays as is configured. During that relaying, no information from the original message is altered or lost. As such, each of the relays as well as the final SETP server will see the original source address, time stamps and message.

Furthermore, SETP guarantees reliable delivery. It is based on TCP, so each of the SETP peers know exactly that the communication partner can successfully receive and process the message. SETP guarantees that new events are only forwarded after the previous ones were successfully received and processed. SETP also checks for on the wire errors. Due to its characteristics, SETP can successfully be used in barely or occasionally connected environments like radio connected systems.

The SETP design is influenced by many industry standard movements, most notably the BEEP protocol and XML. However, SETP is optimized to have a very lightweight footprint. As such, it can be implemented even in low powered devices with little overhead.

More Information about SETP:

<http://www.adiscon.com/Common/en/glossary/SETP.asp>

## 12.6 SMTP

The "Simple Mail Transfer Protocol". This is an Internet standard for sending email messages. Virtually all major email systems are either based on SMTP or at least offer gateways to SMTP capable systems.

SMTP is used for sending email. It can not be used to pick up email messages. For this purpose, protocols like POP3 or IMAP4 are required.

SMTP is highly standardized. As such, a standard email client can work with all SMTP compliant servers. In the public Internet, almost all providers offer SMTP compliant mail servers for their customer's use.

More Information about SMTP:

<http://www.adiscon.com/Common/en/glossary/SMTP.asp>

## 12.7 Syslog Facility

Syslog Facility is one information field associated with a syslog message. It is defined by the syslog protocol. It is meant to provide a very rough clue from what part of a system the message originated from. Traditionally, under UNIX, there are facilities like KERN (the OS kernel itself), LPD (the line printer daemon) and so on. There are also the LOCAL\_0 to LOCAL\_7 facilities, which were traditionally reserved for administrator and application use.

However, with the wide adaption of the syslog protocol, the facility field contents has become a little less clear. Most syslog enabled devices nowadays allow configuring any value as the facility. So it is basically left to distinguish different classes of syslog messages.

The facility can be very helpful to define rules that split messages for example to different log files based on the facility level.

<http://www.adiscon.com/Common/en/glossary/>

## 12.8 TCP

A reliable IP transport protocol. TCP communication ensures that no packets are lost in transit. As such, it is most useful in low-bandwidth or unreliable environments. Examples are slow WANs or packet radio networks.

More Information about TCP:

<http://www.adiscon.com/Common/en/glossary/TCP.asp>

## 12.9 UDP

A non-reliable IP transport protocol. It provides best effort delivery. Typically, in LAN environments UDP packets are never lost. However, in WAN scenarios or with heavily loaded LANs, UDP packets might be lost.

More Information about UDP:

<http://www.adiscon.com/Common/en/glossary/UDP.asp>

## 12.10 Upgrade Insurance

UpgradeInsurance is Adiscon's software maintenance plan. It offers free major upgrades as well as priority support. UpgradeInsurance is available for all Adiscon products and can be purchased for a period between 1 and 5 years.

More Information about Upgrade Insurance:

<http://www.adiscon.com/Common/en/products/upgrade-insurance-details.asp>

## 12.11 UTC

UTC is the so-called "universal coordinated time". UTC was formerly referred to as "GMT" (Greenwich Mean Time) and is the basis of the international time zone system. For example, New York, USA is 5 hours behind UTC. So if it is 12 noon in New York, the UTC time is 5pm.

The MonitorWare line of products often uses UTC. UTC has the fast advantage of providing one consistent time notation, even if devices are across multiple time zones. This is extremely valuable if a central location is to consolidate events from senders in multiple time zones.

Using UTC might not be appropriate if a whole system is contained within a single time zone. As such, most time parameters inside the MonitorWare line of products can be configured to work with local time instead of UTC.

More Information about UTC:

<http://www.adiscon.com/Common/en/glossary/UTC.asp>