Event Reporter 6.3 - SP1

Table of Contents

Part V   Getting Help   87

Part VI   Purchasing EventReporter   89

Part VII   Reference   90

Part VIII   Copyrights   104

Part IX   Glossary of Terms   105

Index

0

1   Introduction

1.1   About EventReporter

EventReporter is an integrated, modular and distributed solution for system management.

Microsoft Windows NT™, Windows 2000™ and Windows XP™ are highly capable operating systems (we will call all of them "NT" in the following documentation). However, their standard event reporting mechanisms are rather limited. Administrators seeking complete control over their server environment need to regularly check the server event logs. Adiscon's EventReporter provides central notification of any events logged to the NT system event log. Messages can be delivered via email and syslog protocol.

The initial product - called EvntSLog - was specifically written with mixed NT and Unix environments in mind. It supported the syslog protocol only. It is currently in use by many large-scale commercial organizations, universities and government bodies (like the military) all around the world. EventReporter empowers data center operators to integrate NT event logs into their central syslog setup. Administrative duties and exception notification can easily be built via Unix-based scripting.

But small sized organizations also demanded relive from checking server logs. As such, EventReporter allows delivery of NT event notifications via standard Internet email. Each server's events are gathered, filtered according to rules set up by the administrator and - if they matter - forwarded to the admin. Especially small sized organizations operating a single server can be rest assured that they won't miss any important log entries.

EventReporter can be teamed with Adiscon's WinSyslog and the MoniLog product. In this scenario, it provides a totally centralized and automated event log collection, monitoring and analysis solution. If you are looking for a solution that not only can forward event information but also monitor additional system settings, you might want to have a look at the MonitorWare Agent.

EventReporter is also a great tool for computer resellers, consultants and other service providers in need to monitor their customer's systems.

The product is easy to install and configure, uses only minimal system resources and is proven to be reliable. Furthermore, it is extremely inexpensive with a per system licensing fee starting at US$ 49.

1.2   Features

Centralized Logging

This is the key feature. EventReporter allows consolidation of multiple NT event logs and forward them automatically to either a system process or an administrator.

Ease of Use

Using the new EventReporter client interface, the product is very easy to setup and customize. We also support full documentation and support for large-scale unattended installations.

Syslog Support

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

SETP Support

SETP was originally developed for MonitorWare but now it's a key feature added in EventReporter 6.2 Professional Edition. NT Event Messages can be forwarded using SETP protocol. [Click here](#) for more information on SETP.

Email Support

NT event log information can also be delivered via standard Internet email. This option is an enabler for smaller organizations or service providers unattended monitoring their client's servers.

Local Filtering

EventReporter can locally filter events based on the NT event log type (e.g. "System" or "Application") as well as severity.

Full Windows 2000 and XP Support

We had full Windows 2000 and XP support since these products were released! All extended Windows 2000 log information can be gathered, fully decoded and submitted to the log targets (email or syslogd).

Robustness

EventReporter is running in a large number of installations. It is written to perform robustly even under unusual circumstances. Its reliability has been proven at customers' side since 1997.

Remote Administration

The client can be used to remotely manage EventReporter instances.

Minimal Resource Usage

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

Full NT Event Log Decoding

EventReporter can fully decode all types of NT event log entries. It has the same capabilities like event viewer.

NT Service

The EventReporter Service is implemented as a native multithreaded Windows NT service. It can be controlled via the control panel services applet or the computer management MMC (Windows 2000).

Full Windows 2000, 2003 and XP Support

We have full Windows 2000 support since Windows 2000 ships! WinSyslog versions 3.6 and above are specifically designed for Windows XP and support advanced features like the new themes and fast user switching.

Runs on large Variety of NT Systems

NT 3.5(1), 4.0, 2000 or XP; Workstation or Server - EventReporter does run on all of them. We also have Compaq (Digital) ALPHA processor versions on platforms supporting this processor (engine only, available on request).

Double Byte Character Set Support (e. g. Japanese)

EventReporter supports characters encoded in double byte character sets (DBCS). This is mostly used with Asian languages like Japanese or Chinese. All DBCS strings are forwarded correctly to the syslog daemon or email recipient. However, the receiving side must also be able to process DBCS correctly. Adiscon's syslog daemon for Windows, WinSyslog, does so. The output character encoding is selectable and support Shift-JIS, JIS and EUC-JP for Japanese users.

Multi-Language Client

The EventReporter client comes with multiple languages ready to go. Out of the box English, French, German, Spanish and Japanese are supported. Languages can be switched instantly. Language settings are specific to a user.

Additional languages can be easily integrated using Adiscon's brand new XML based localization technology. We ask customers interested in an additional language for a little help with the translation work (roughly 1 hour of work). Adiscon will than happily create a new version. This service is free!

Friendly and Customizeable User Interface

New Skinning feature added into the EventReporter Client. By default 5 new fresh skins are installed and can be selected. These skins can be colorized with Hue, Saturation and RGB colors. Click to see.

New Cloning feature added to the EventReporter Client. In short you can now clone a Ruleset, a Rule, an Action or a Service with one mouse click.

Move up and Move down function has been added for Actions in the EventReporter Client.

The EventReporter Client Wizards has been enhanced for creating Actions, Services and RuleSets. And other minute changes!

1.3   Components

EventReporter Client

The EventReporter Client is used to configure all components and features of EventReporter. 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.

EventReporter Service

The EventReporter Service - called "the service" runs as an NT Service and coordinates all log processing and forwarding activity at the monitored system (server or workstation).

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

The EventReporter 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 service operates as follows:

After starting, it periodically reads the NT event log. Each message is formatted and then sent to the given syslog daemon or email recipient. After all entries have been read, EventReporter goes to sleep and waits a given amount of time without any processing. This so-called "sleep period" is user configurable. As soon as the service returns from the sleep period, it once again iterates through the NT event logs. This processing continues until the process is stopped.

Due to its optimized structure, EventReporter uses only very minimal processing power. How much it uses mainly depends on how long the sleep period is. We recommend a sleep period between 1 and 5 minutes for syslog delivery and some hours up to 1 day for email delivery. However, feel free to customize this value according to your needs. We strongly recommend not to use sleep periods of 500

milliseconds or less (although possible).

1.4 System Requirements

EventReporter has minimal 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 **EventReporter client** needs roughly 10 MB of disk space.
- Internet Explorer 5.5 (or higher) is necessary for the Client.
- The EventReporter client is optional and needs not to be present on a production system.
- 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.

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.
- **Engine-only installations** require roughly 200 KB of disk space and 2MB of virtual memory. Please note that this is not actual used RAM - RAM usage is roughly 1 MB during iterations (can be higher for very large entries). During the idle period, the engine does not need any actual RAM - just swap space. Idle periods are implemented via operation system sleep() calls which do not use any processor cycles at all.
- **Please note that EventReporter is developed under Windows 2000 and XP.** It is tested under Windows 2000, XP and NT 4.0. Although not tested under NT 3.5(1), we do not see any reason why it should not perform well in this environment.
- EventReporter runs on top of Windows NT server and Windows NT Workstation. Under Windows 2000, the 3 additional event logs ("DNS Server", "File Replication Service" and "Directory Service" are automatically supported).
- The default install set (most probably the one you found in this documentation) contains the executable for the Intel platform. However, there is an ALPHA version available on request. As ALPHA is not supported for Windows 2000 or XP, there is no ALPHA executable for those operating systems.

2 Getting Started

EventReporter can be used for simple as well as complex scenarios. This chapter provides a quick overview of EventReporter and what can be done with it. Most importantly, it contains a tutorial touching many of the basic tasks that can be done with EventReporter 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 Installation

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

A number of different Download Versions of the product is 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 small 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 or to find a setup.exe or such.

Depending on the download directory, the setup program may also be supplied in a ZIP file. The EventReporter setup is based on Microsoft Windows Installer technology. So it can easily be integrated into MSI aware tools.

All users are highly encouraged to use the full install. It is the default install set download able from the EventReporter web site.

Note: EventReporter must be installed by a user with administrative permissions.

2.2 Obtaining a Printable Manual

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

The manuals offered on this web page are in printable (in PDF format) or HTML Versions for easy browsing and printing. The manual is also included as a standard Windows help file with all installations. So if you have the product already installed, there is no need to download these documents.

The version on the web might also include some new additions, as we post manual changes frequently – including new samples and as soon as they become available. Past manual versions are also available for those customers in need of it.

2.3 EventReporter Tutorial

The goal of this tutorial is to provide a rough overview over the EventReporter as well as some typical uses. It is in no way complete, but should help in understanding EventReporter 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 specified scenarios that follow below.

EventReporter 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 condition evaluates to true ("matches"), actions associated with this rule are carried out (e.g. storing the information unit to disk or emailing an administrative alert).

Note: The screenshots in this tutorial are made with EventReporter 6.2 and MonitorWare Agent. MonitorWare Agent, is used as the user interface is similar to the one EventReporter 6.2 uses.

2.3.1 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 filter conditions.

If you are note 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.

XOR – it yields true if exactly one (but not both) of two operands is true. Example: XOR (A, B): The end result is false if A and B both are True or 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.

TRUE – returns true.

FALSE – returns false.

2.3.2 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 ID 105, 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:
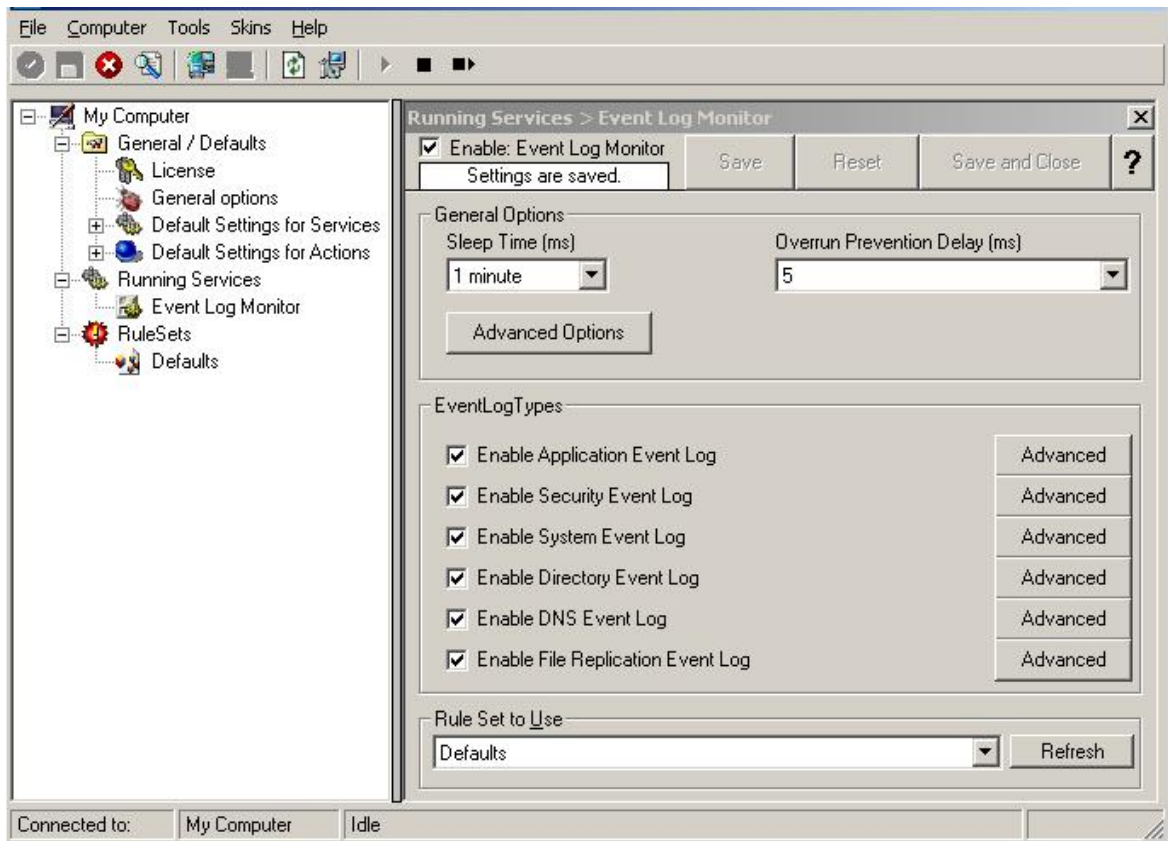

Ignoring Events - Figure 1

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 it 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.
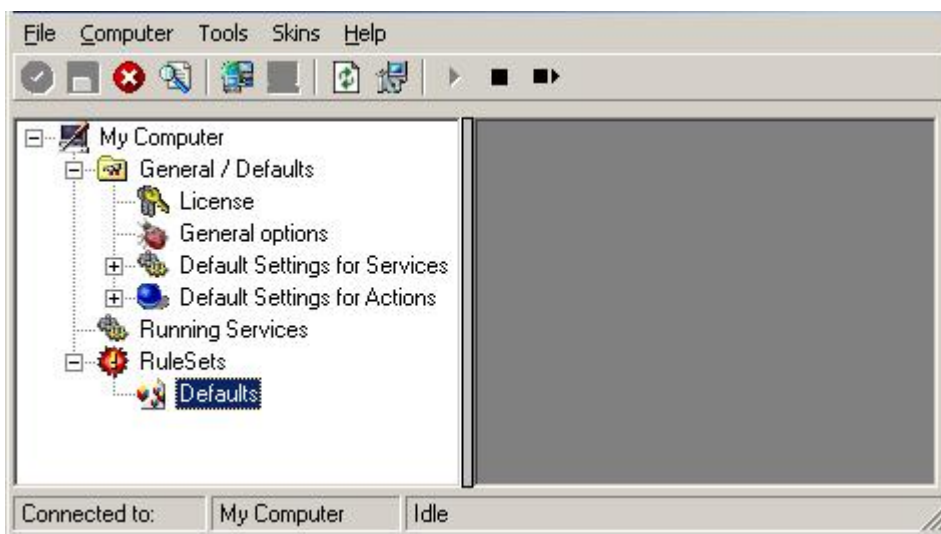
Ignoring Events - Figure 2

Of course we can only use a rule if we configure a corresponding service. To do so, right-click on "Running Services" and then select "Add Services". Choose the desired "Service" from the context menu i.e. "Event Log Monitor" in this sample. 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:
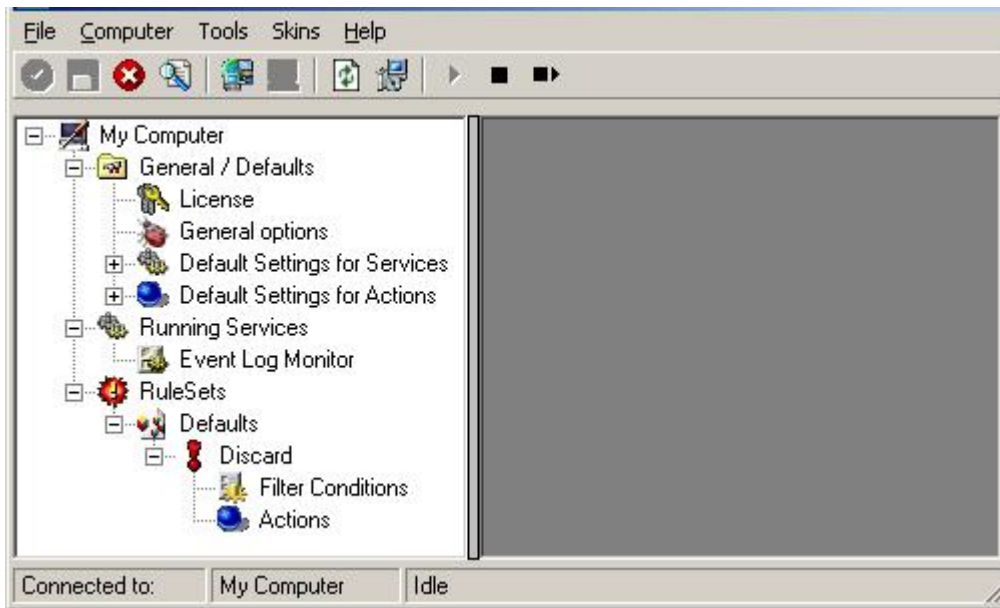
Ignoring Events - Figure 3

As we had 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 "Rules", select "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:

Ignoring Events - Figure 4

Click on "Filter Conditions" to see this dialog:
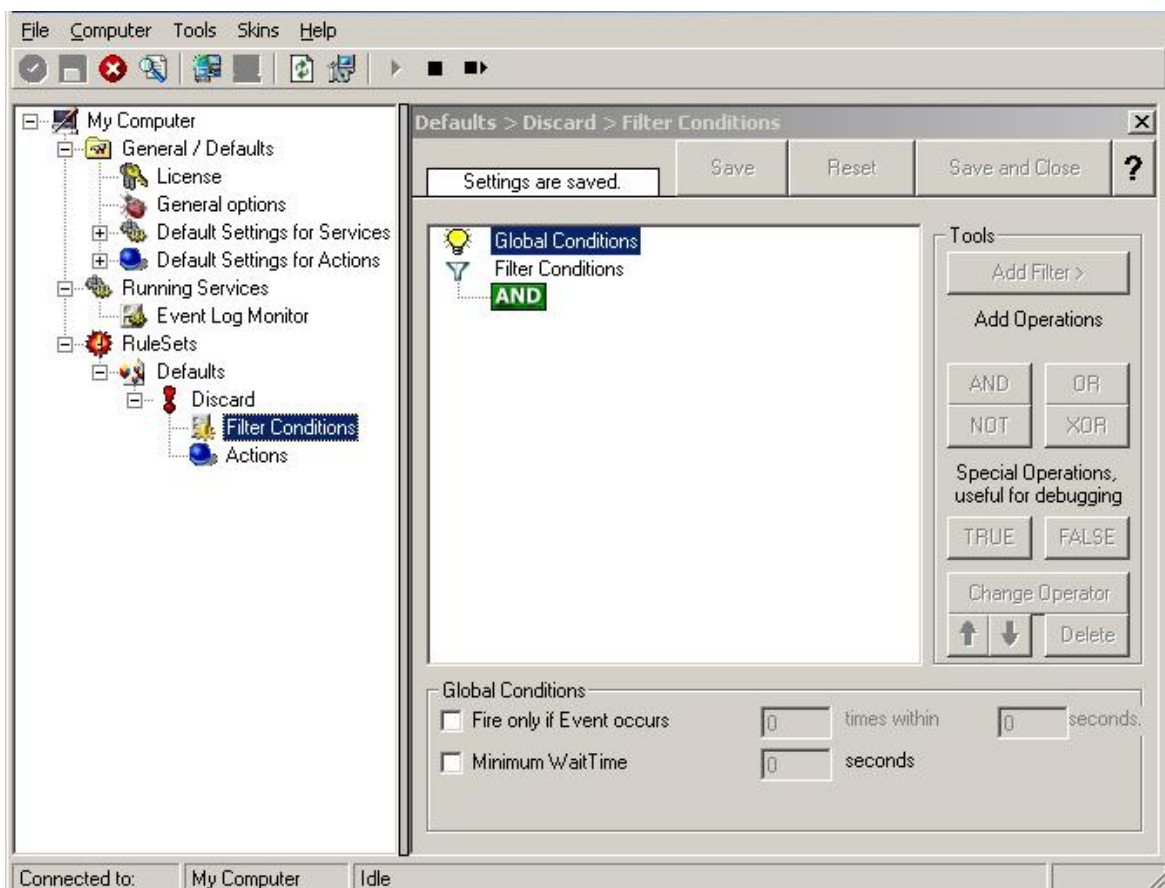
Ignoring Events - Figure 5

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. Or you can use the Add Filter Button. This can be seen in the screen shot below:
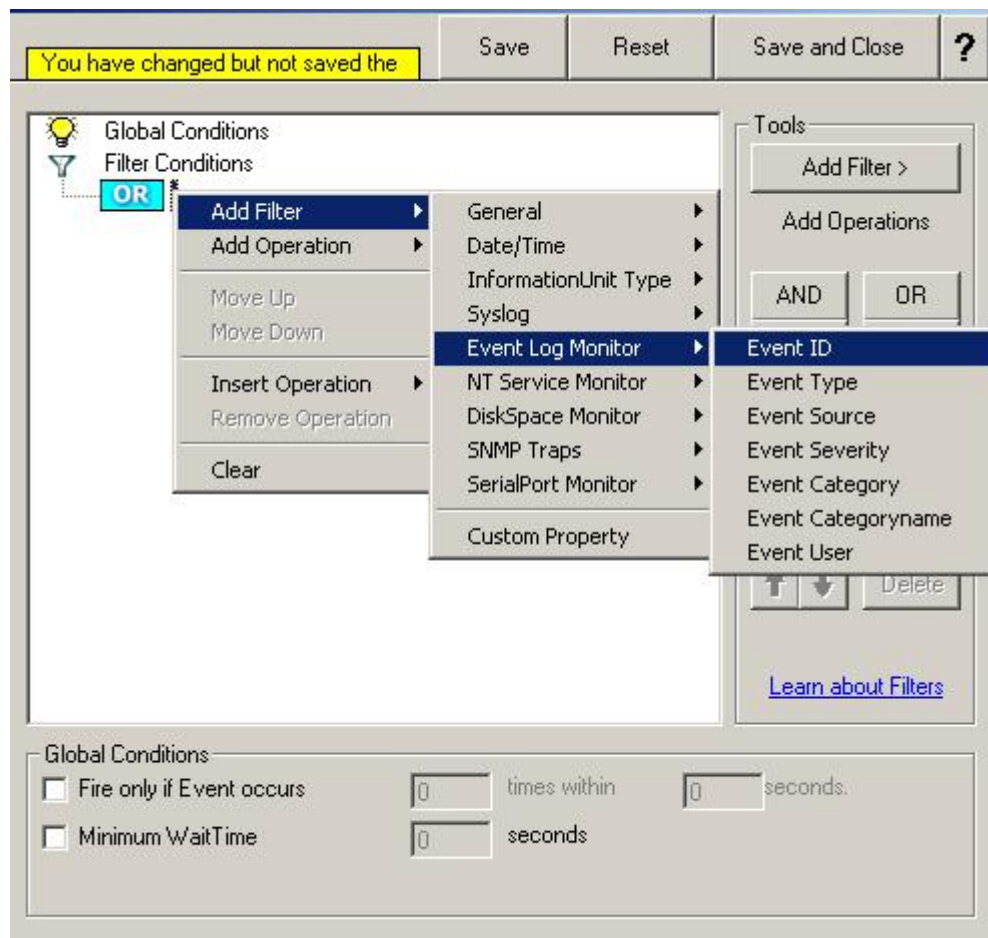
Ignoring Events - Figure 6

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:
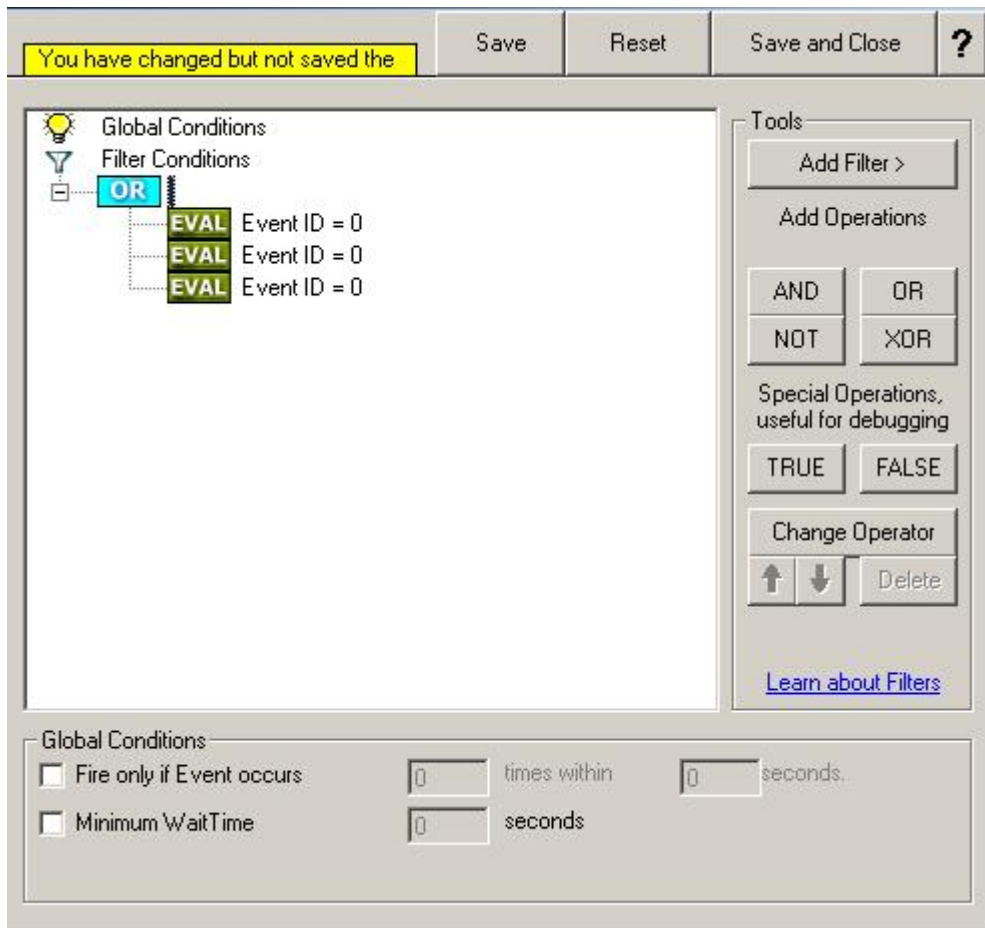
Ignoring Events - Figure 7

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, you screen should look as follows:
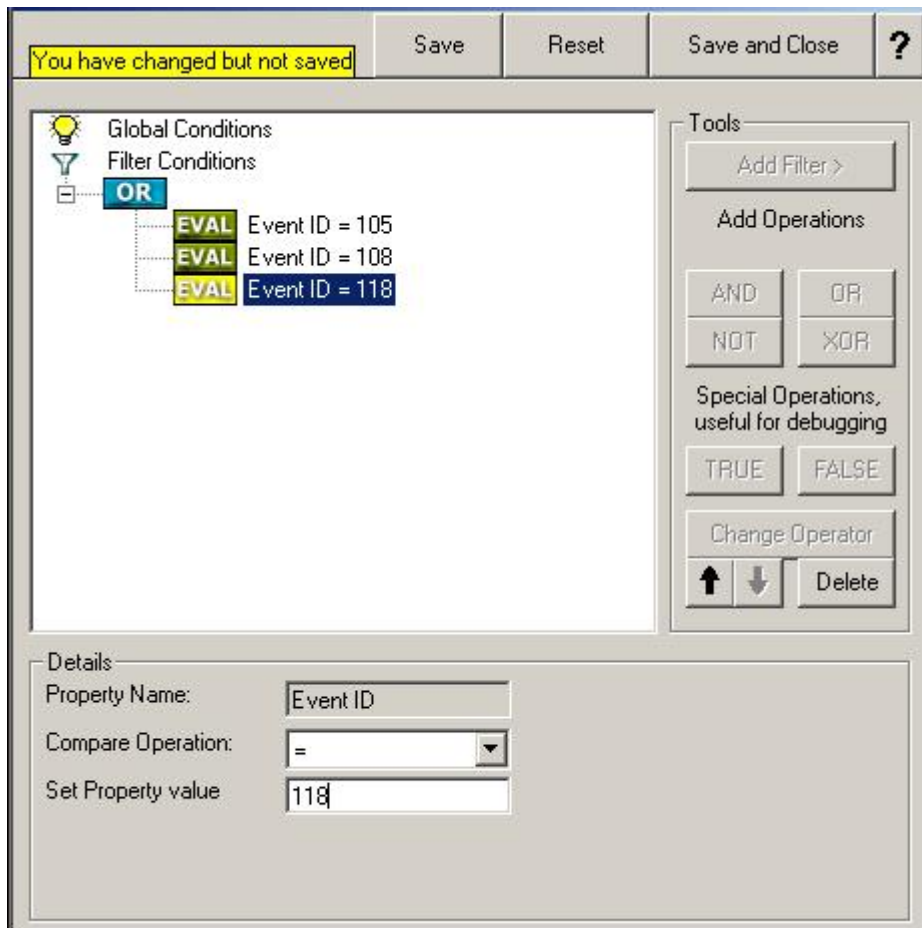
Ignoring Events - Figure 8

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."
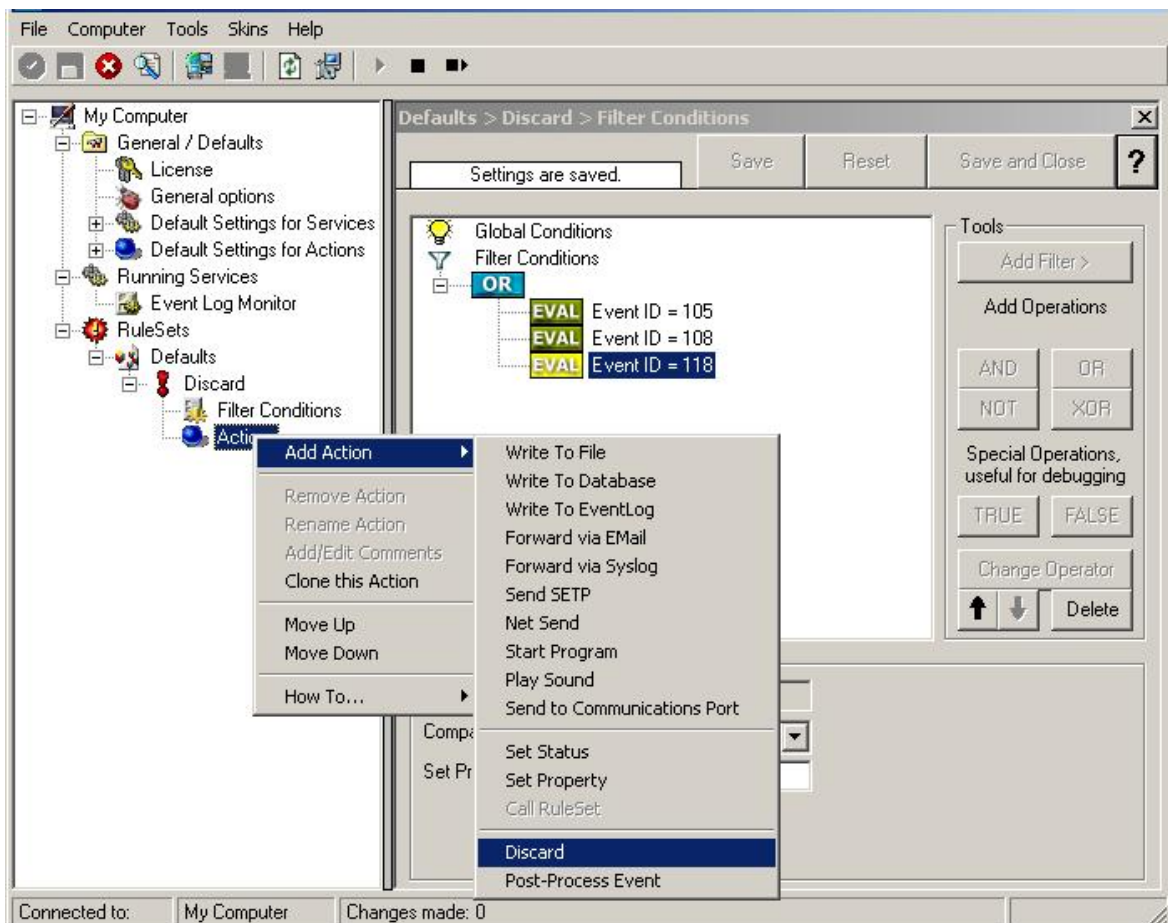
Ignoring Events - Figure 9

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 like follows:
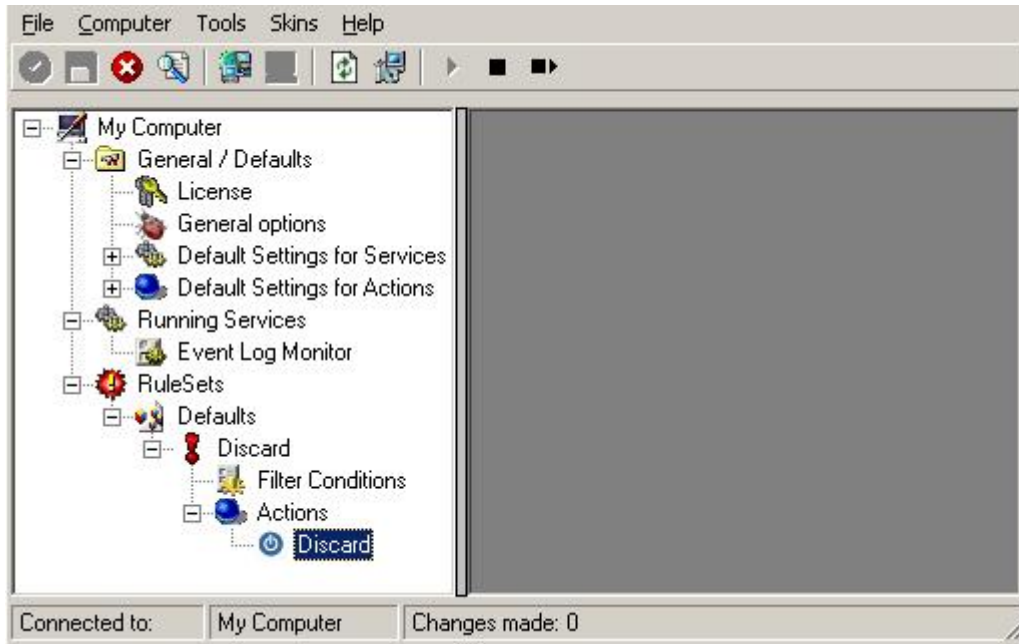
Ignoring Events - Figure 10

This concludes the definition of our first rule.

If we would start EventReporter 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.

2.3.3   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 the need arises. 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 into a log file.

To do so, right click the "Defaults" rule set as shown below. Then, select "Rules" and "Add Rule":
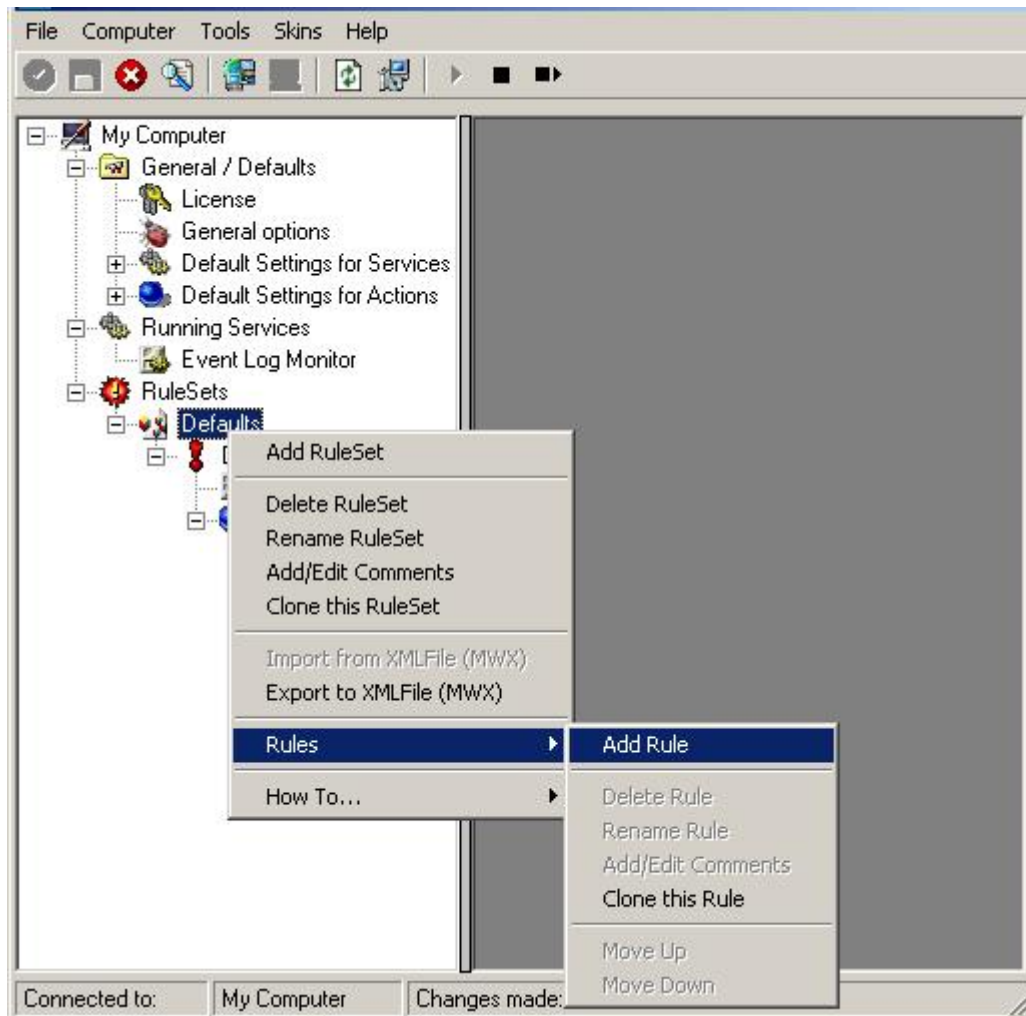
Logging Events - Figure 1

Use a name of you 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:
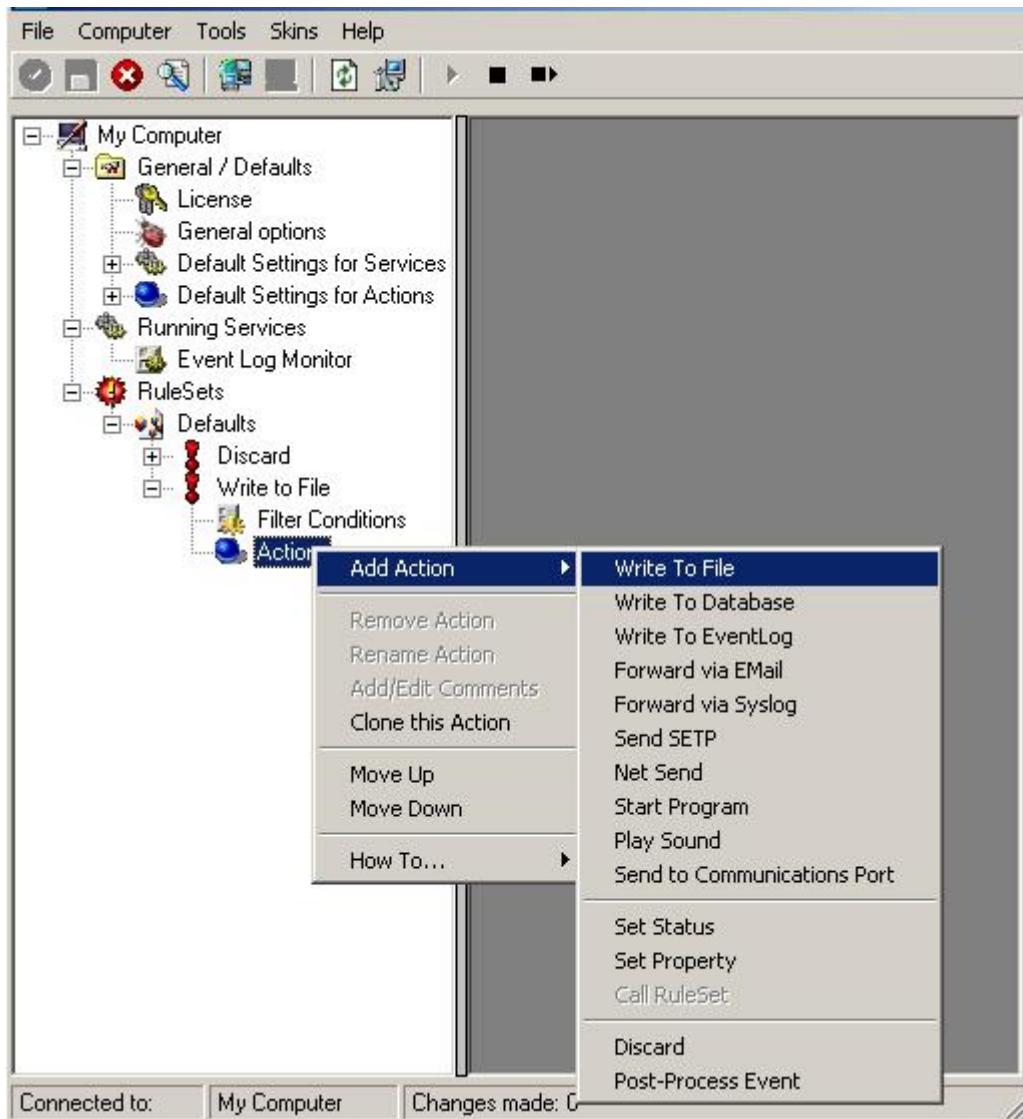
Logging Events - Figure 2

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:
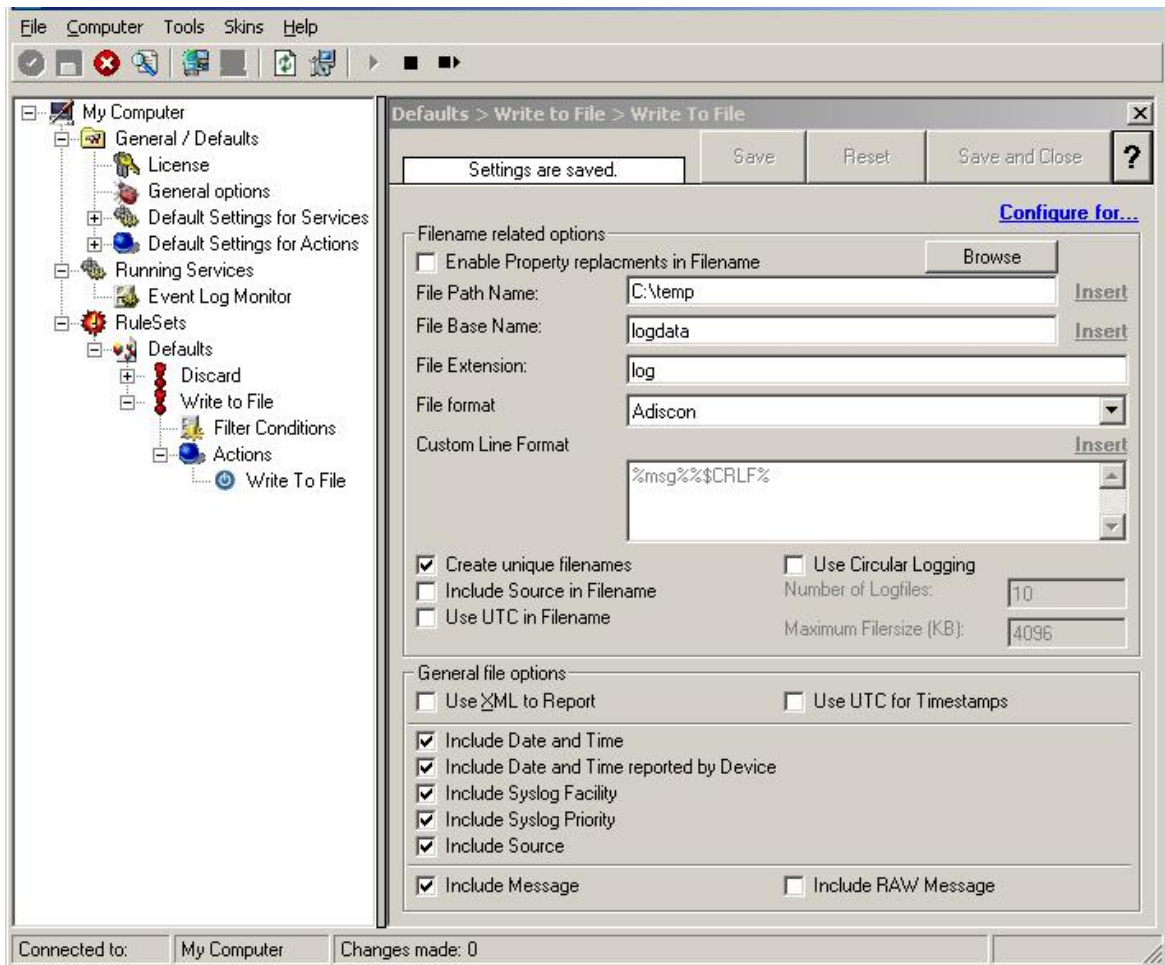
Logging Events - Figure 3

Important

If the configured directories are missing, they will be automatically created by EventReporter i.e. the folder specified in "File Path Name".

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 EventReporter by hitting the start button seen below:

Logging Events - Figure 4

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 MonitorWare Console.

Please note that the current date is appended to the log file. This facilitates file management in 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.

2.3.4 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 night time. 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, EventReporter 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.
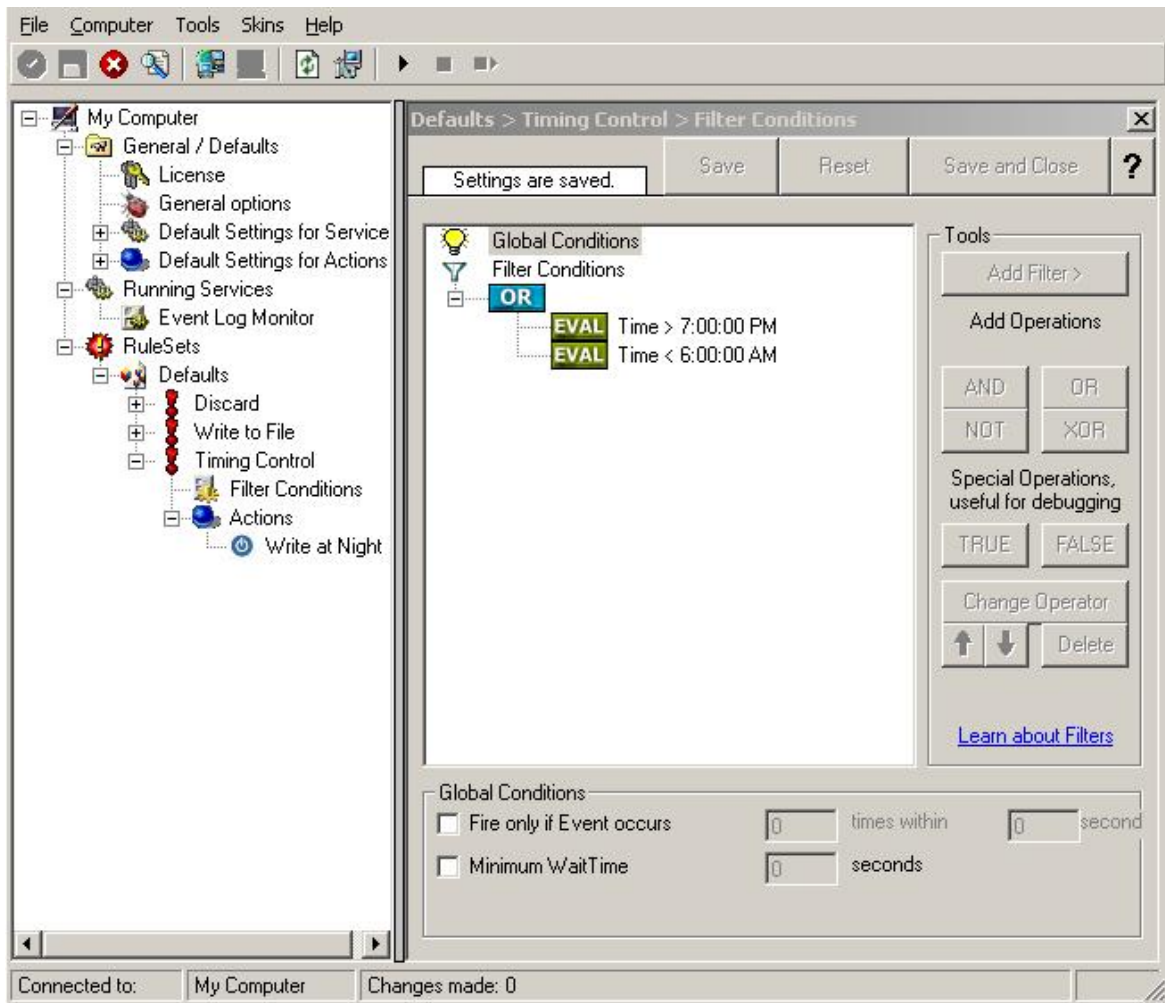
Time-Based Filters - Figure 1

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 night times 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:
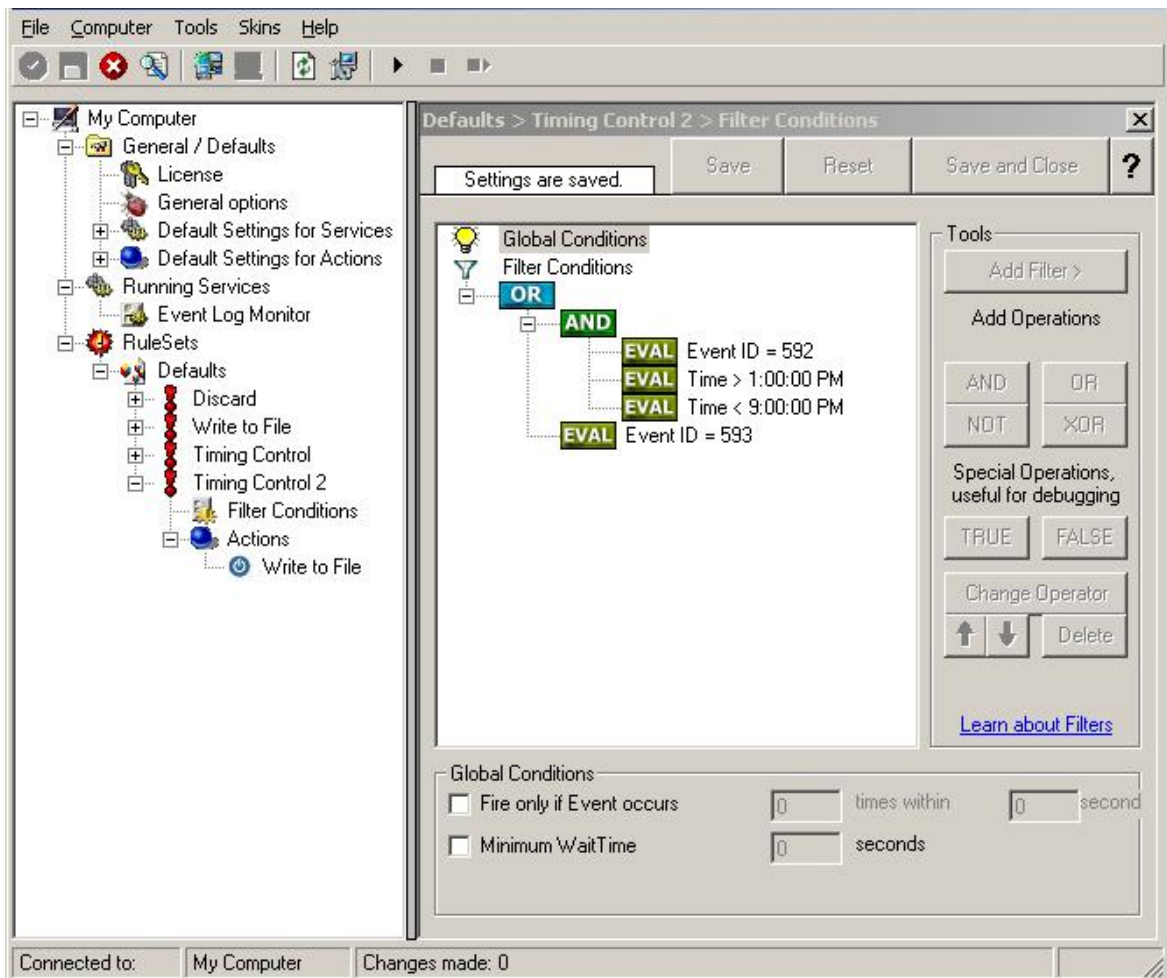
Time-Based Filters - Figure 2

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 = 593   = *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).

2.3.5 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 at least 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.
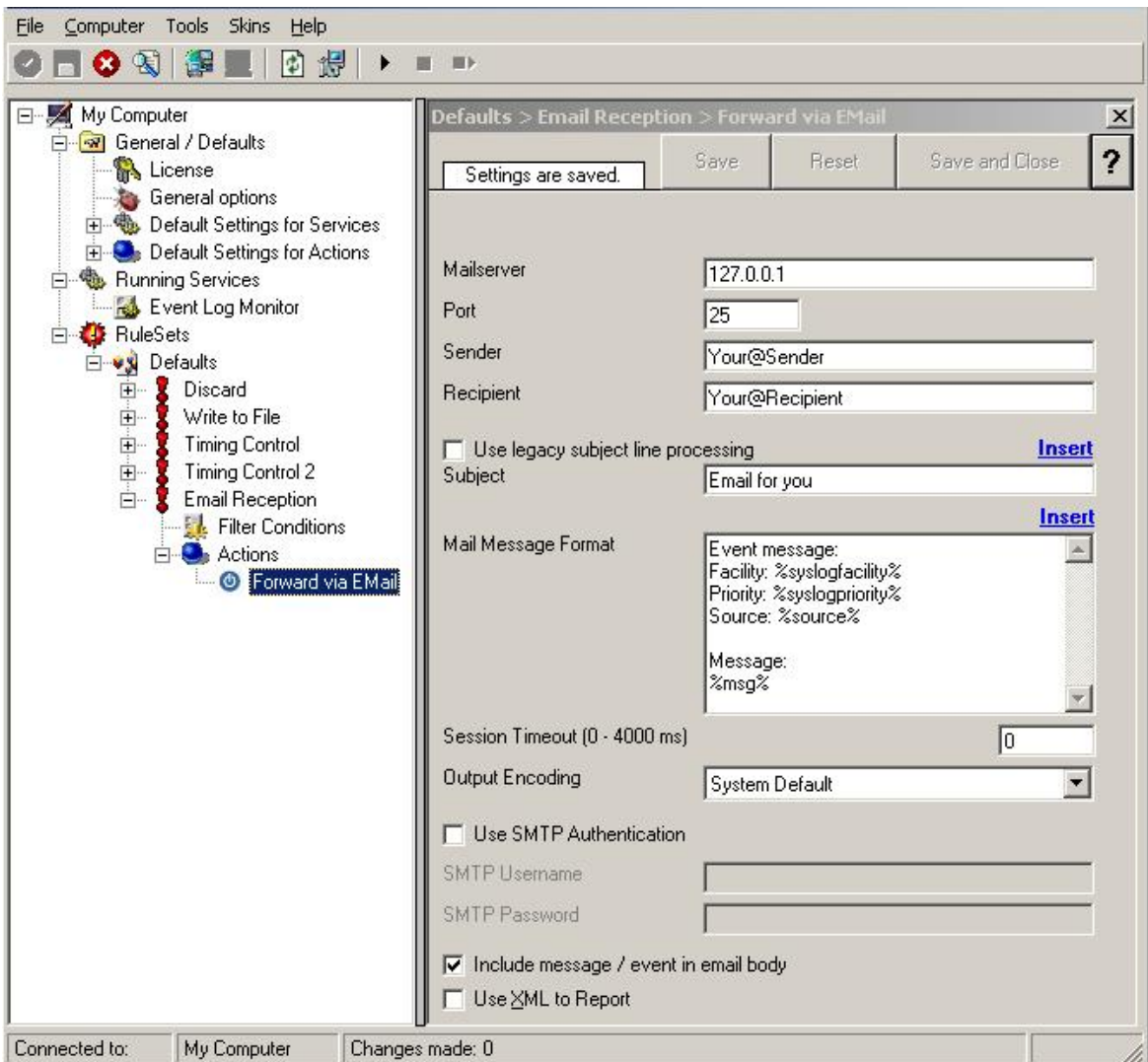
Email Notifications - Figure 1

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:
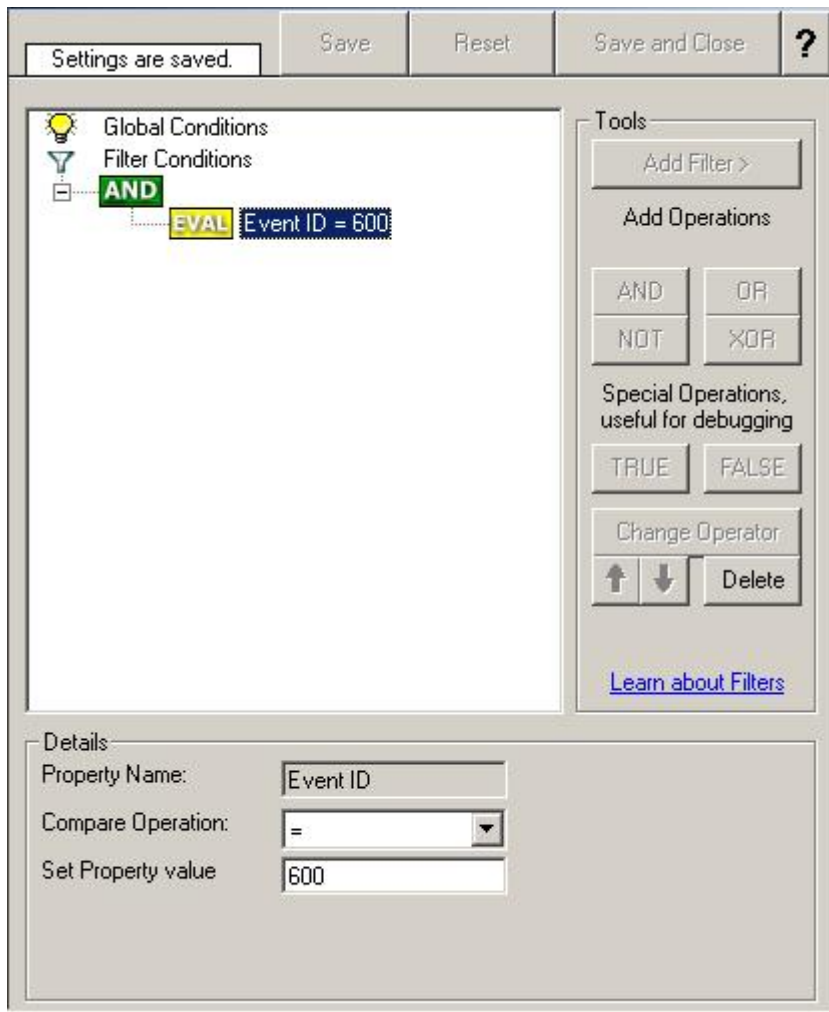
Email Notifications - Figure 2

When you have finished these steps, be sure to save the configuration and re-start the EventReporter 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.

2.3.6 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).
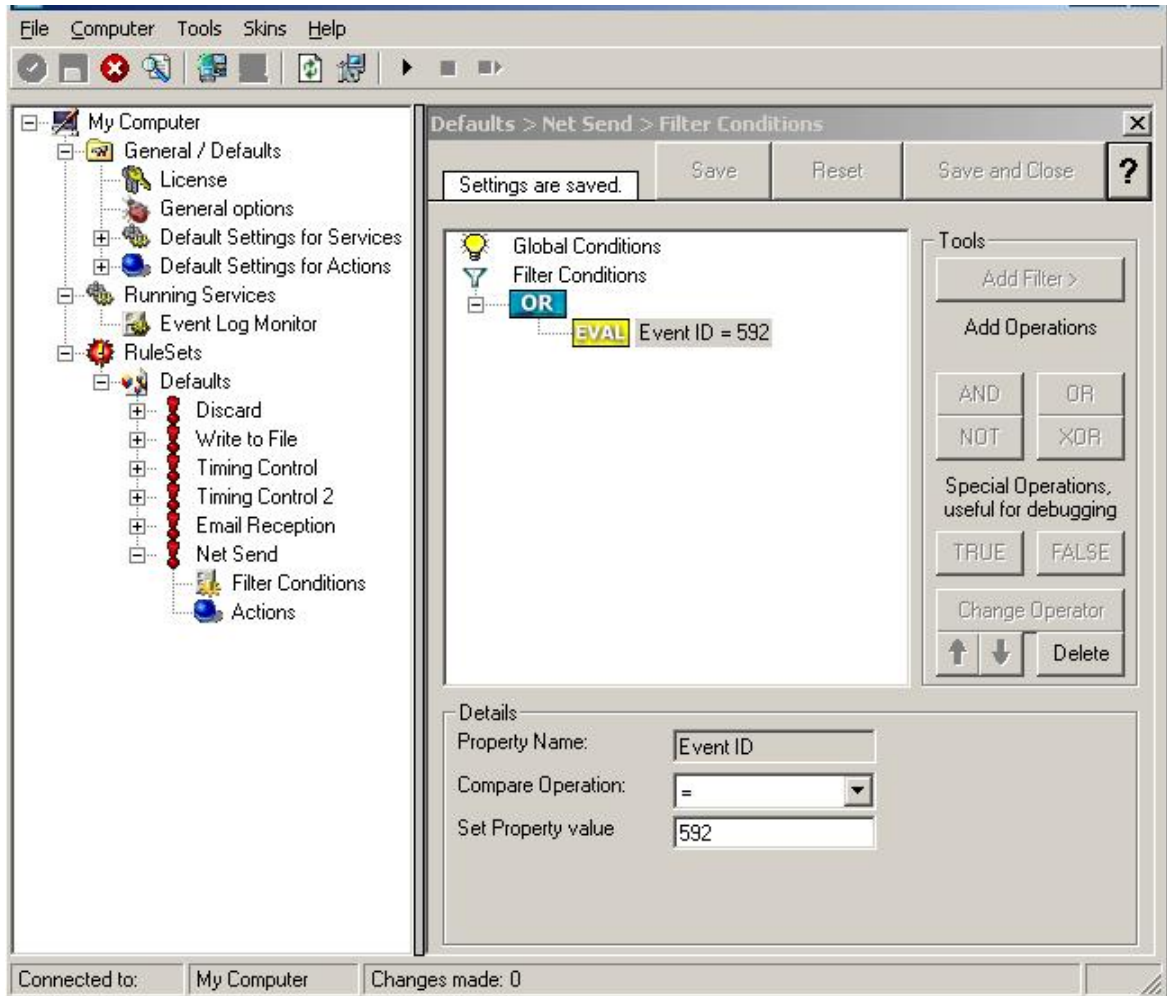
Alarming via Net Send - Figure 1

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".
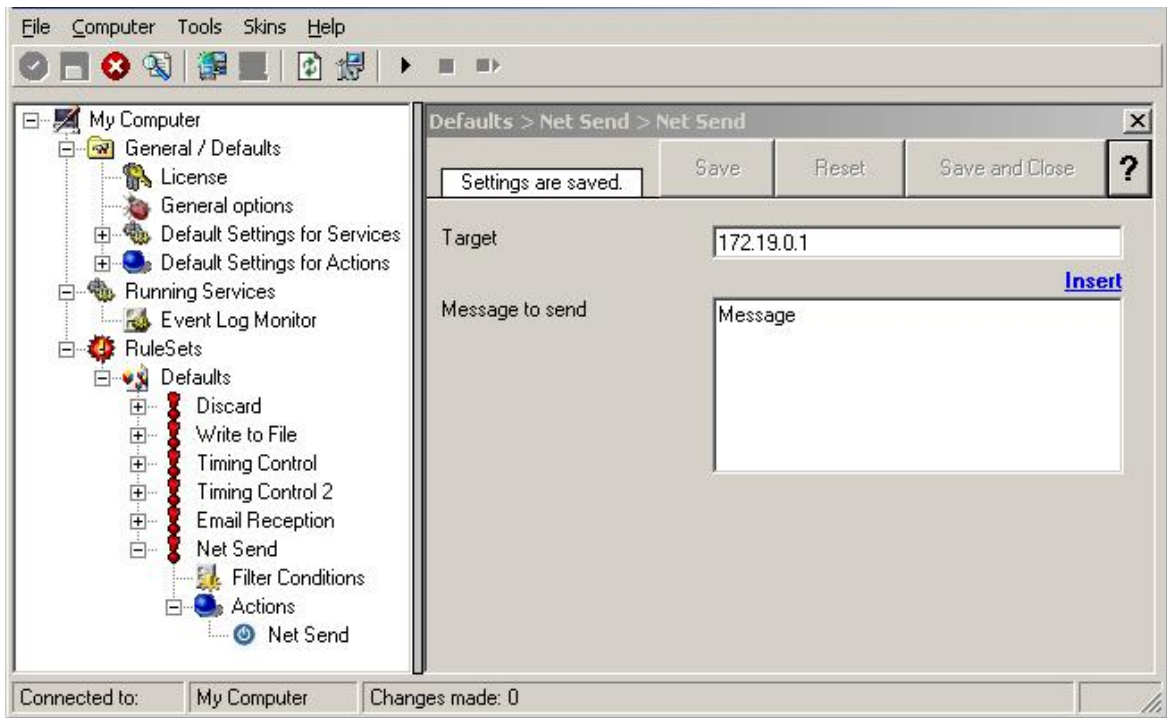
Alarming via Net Send - Figure 2

After saving the configuration and restarting the EventReporter, we will receive notifications if the filter condition evaluates to true. A sample message might look like this (slightly obscured in this sample):
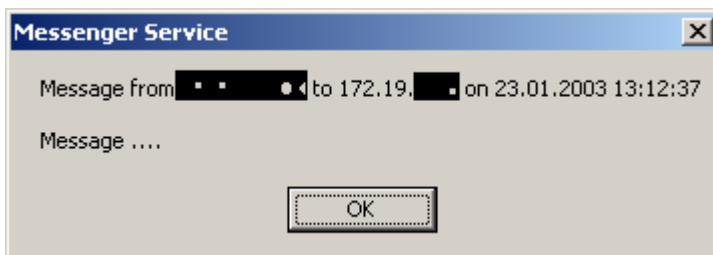
Alarming via Net Send - Figure 3

2.3.7   **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.
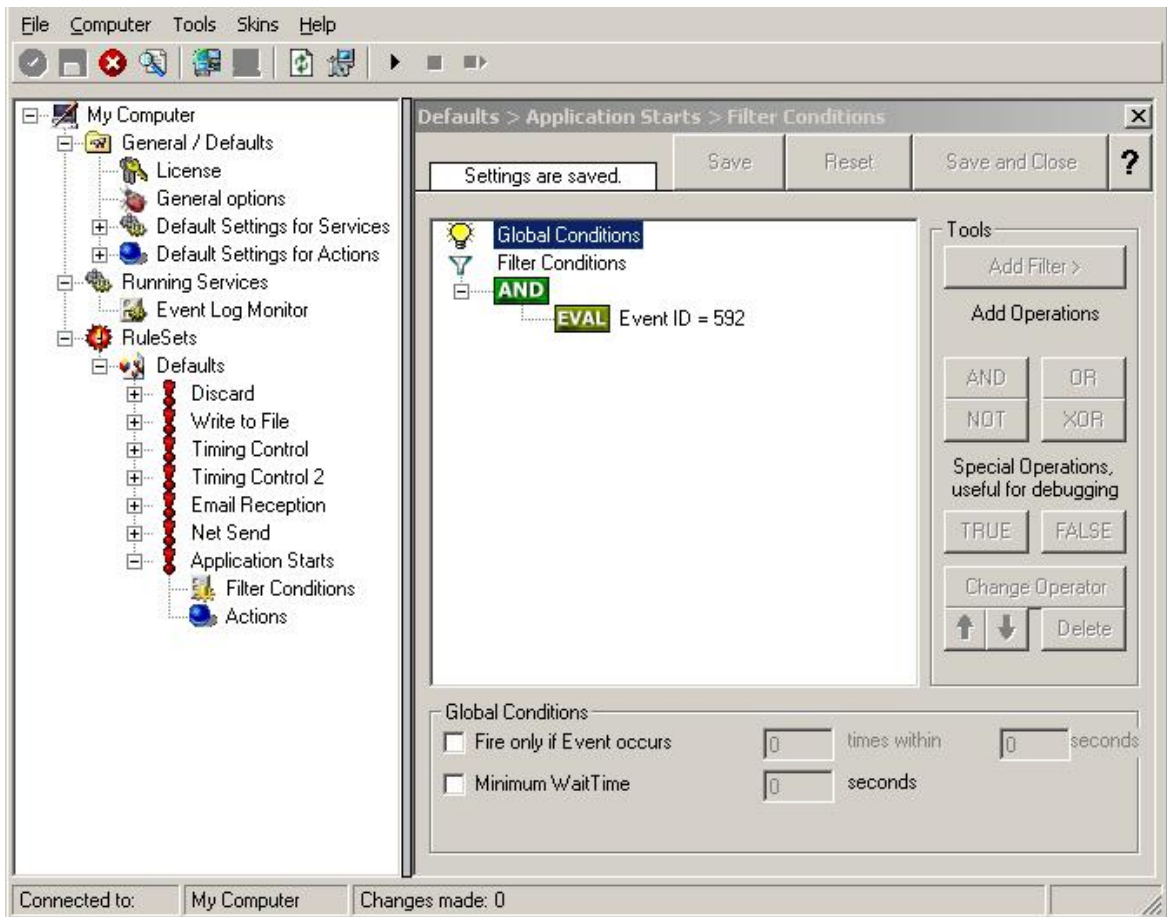
Starting Scripts and Applications in Response to an Event - Figure 1

The start program action is just a "normal" action:
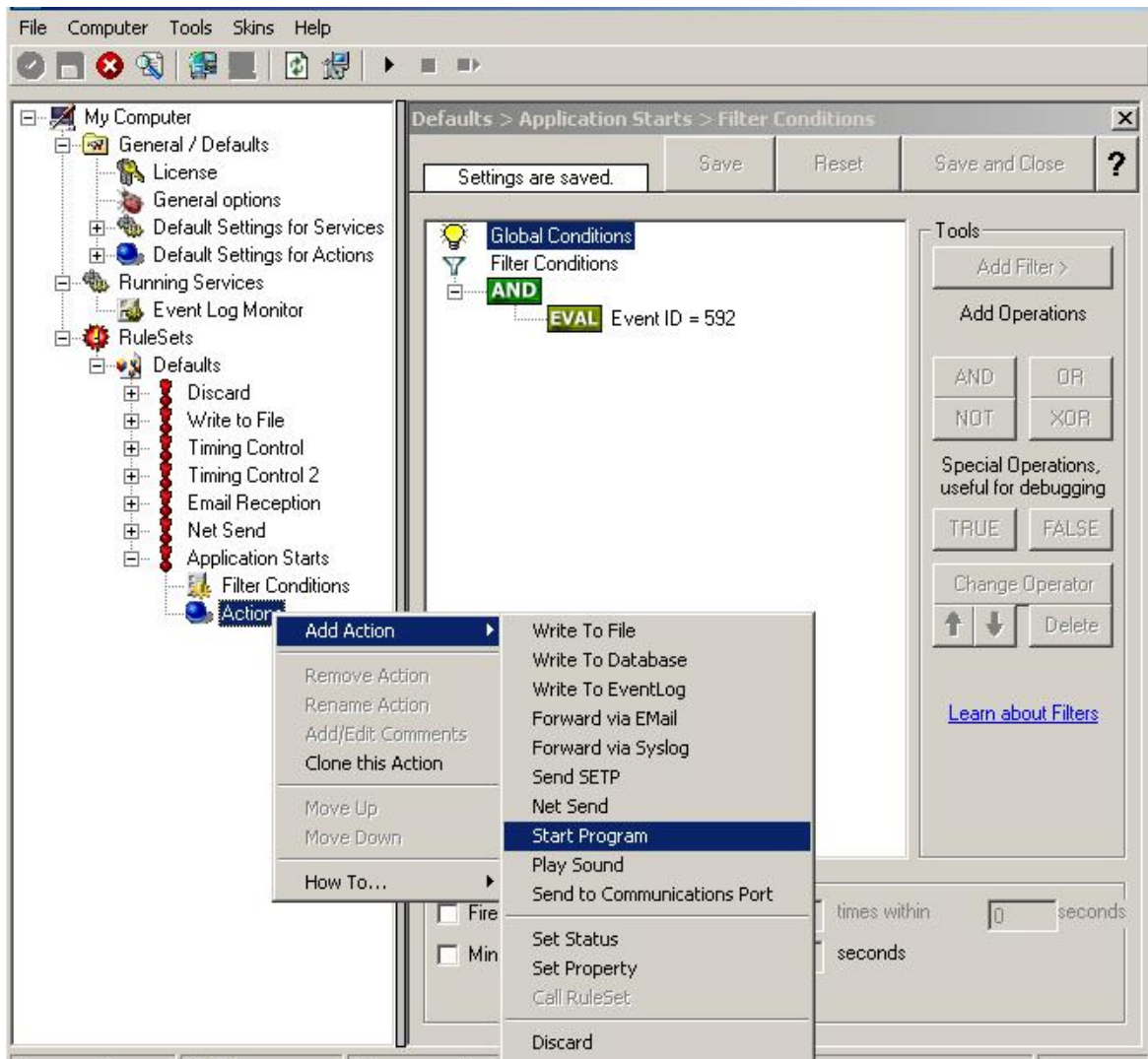
Starting Scripts and Applications in Response to an Event - Figure 2

In the "Start Program" action's parameters, select the file to run as well as all parameters to be supplied to it (if any):
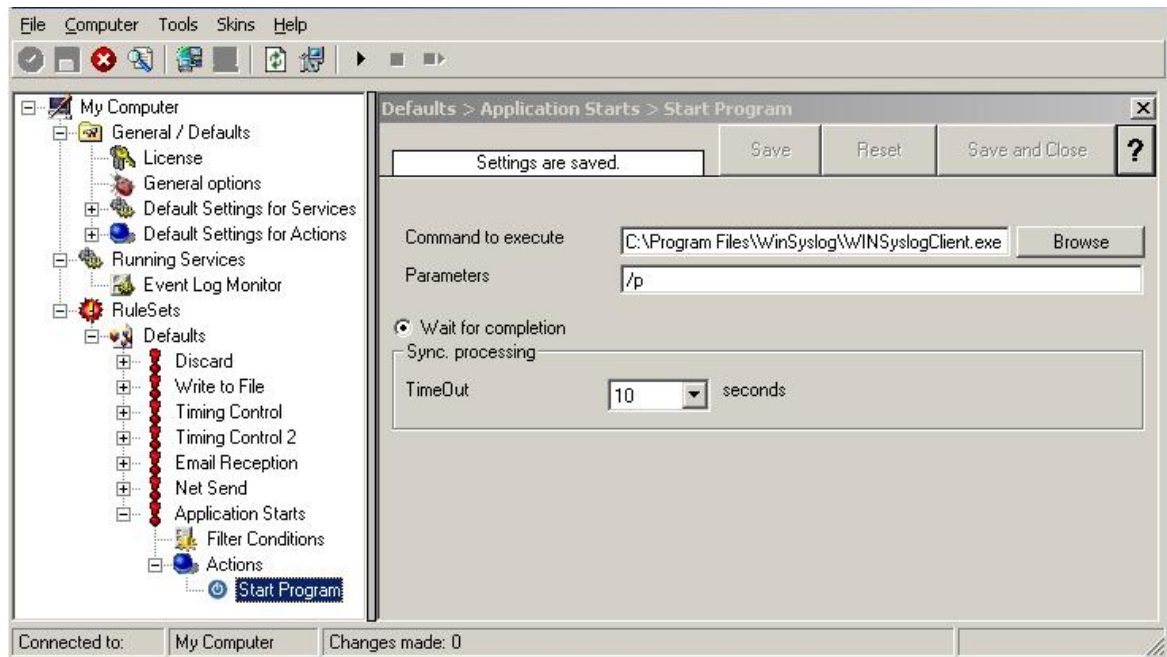
Starting Scripts and Applications in Response to an Event - Figure 3

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

3   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:

Installations and Configurations

- How To Setup MonitorWare Agent/ WinSyslog/ EventReporter
- How To setup MonitorWare Console 2.0
- How To setup MonitorWare Console 1.1
- Configuring Windows for the Event Log Monitor
- Sample Syslog device configurations
- Intrusion detection via the Windows event log
- Firewall setup for MonitorWare Agent
- Creating a hardened log host

Services

- How To Configure a Syslog Server
- How To setup SETP Server Service
- How To setup EventLogMonitor Service
- Creating a simple Syslog server
- Forwarding NT event logs to a Syslog server
- Forwarding NT event logs to an SETP server

Actions

- How To setup an SETP Action
- Creating a rule set for database logging

Centralized Monitoring / Reporting

- How do I apply filters for database reports? (MonitorWare Console 2.0)
- How To Schedule Reports with MonitorWare Console 2.0
- How To Generate PIX Reports with MonitorWare Console 2.0 Manually
- How To Generate Windows Reports with MonitorWare Console 2.0 Manually
- How To setup PIX centralized Monitoring
- Centralized Event Reports with MonitorWare Console
- Centralized event reports with Monilog
- How To Report Log Truncation
- How To Generate PIX Reports with MonitorWare Console 1.1 Manually
- How To Generate Windows Reports with MonitorWare Console 1.1 Manually
- How To Schedule Reports with MonitorWare Console 1.1

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.
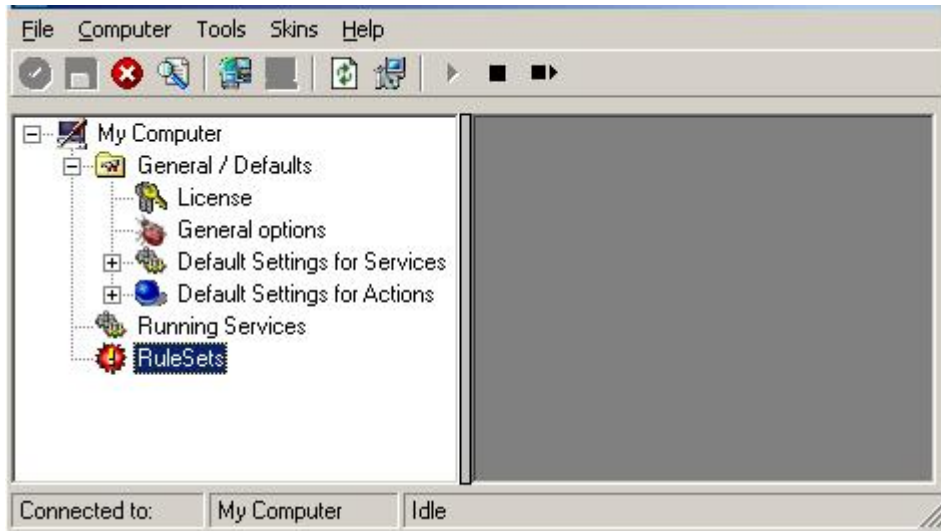
4 Configuring EventReporter

EventReporter is easy to use and is powerful.

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

The EventReporter 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 EventReporter configuration client application. It is used to configure the service settings.

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

EventReporter 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 EventReporter 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. EventReporter 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 EventReporter that limits from doing so.

The service definition looks like this:
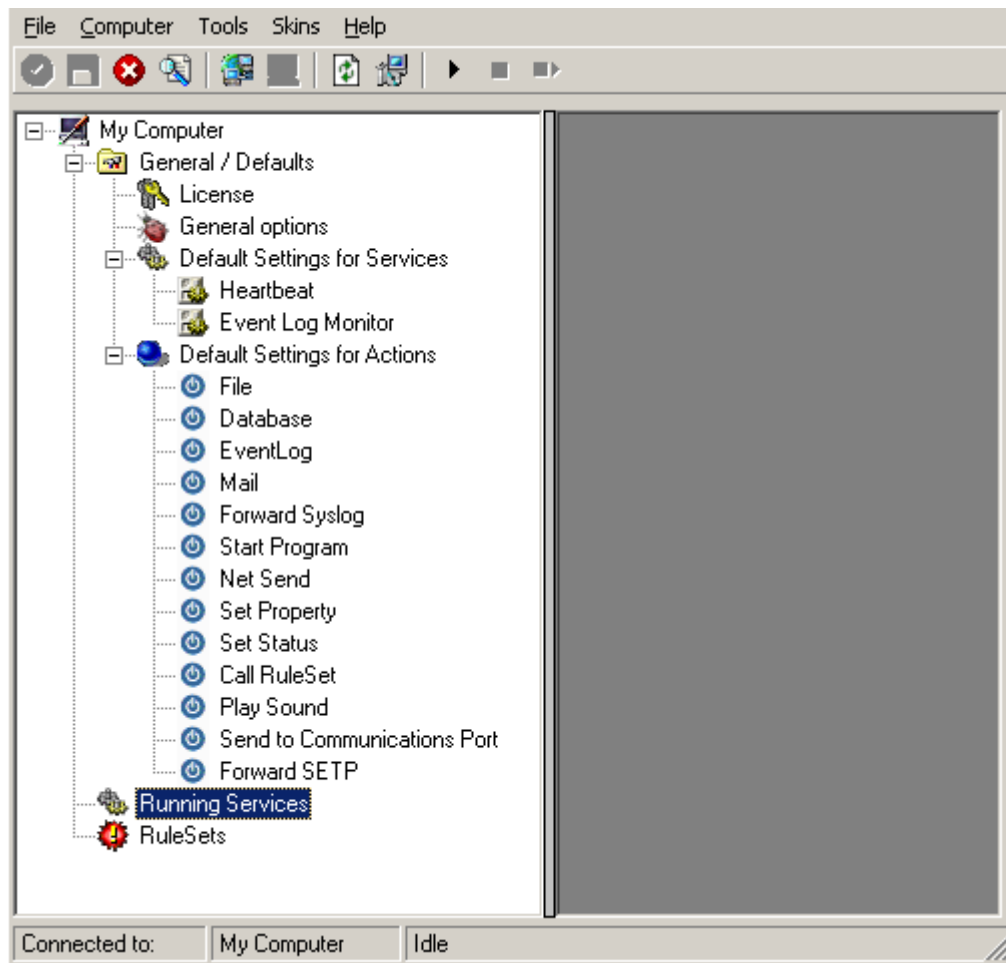
EventReporter Configuration Client - Service Definition View

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.

4.1 License Options

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

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 that 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.

4.2   General Options

This tab can be used to debug rule bases. Especially with complex bases, it might be necessary to learn what EventReporter 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.**

Debug Options Parameters

General Options

The General Options available on this form are explained below:

Process Priority

Configurable Process Priority to fine-tune MonitorWare Agent behavior.

Queue Limit

This value is a maximum number of internal Queue elements MonitorWare Agent can hold. By default this is 0 which means unlimited.

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.

Default TimeValues based on UTC or Local Time

This option allows you to configure the timemode that will be used e.g. in File Logging, Database Logging, Send Email actions and everywhere, where time is used. In addition to this it will also set the time mode in all those places where time mode either UTC or Local Time is not defined.

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 the specify a fully qualified file name including the drive.

Note: If the configured directories are missing, they will be automatically created by MonitorWare Agent i.e. the folder specified in "File and Path Name".

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.

4.3 Services

4.3.1 Understanding Services

Services gather event data. For example, the Syslog Server service accepts incoming syslog messages and the 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.

4.3.2   Event Log Monitor

This dialog configures the Windows Event Log Monitor service.

This service was intially 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.

Event Log Monitor Properties

General Options

The General Options available on this form are explained below:

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 miliseconds.

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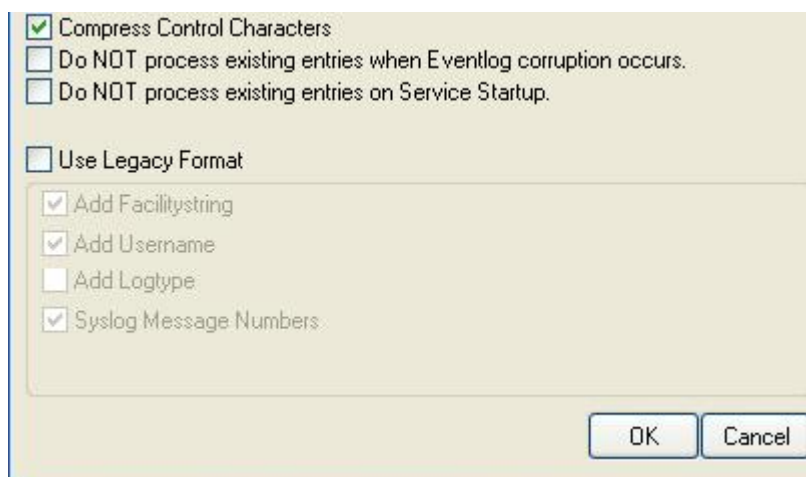
This property allows configuring a delay after generating an event. The time is the delay in milliseconds.

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.

Advanced Options

Once you click on the Advanced Options Button a screen-shot similar to the one below is opened. The options available on this form are explained below:

Event Log Monitor - Advanced Options

Compress Control Characters

This option allows you to control the control character removal and space compression. If checked, control characters (e.g. CR, LF, TAB - nonprintable characters in general) are removed. Also multiple spaces are compressed to a single one. By default this is checked. We recommend keeping it checked for most cases as it provides better display.

Please note that it should be unchecked if events should be forwarded via email. And it MUST be turned off if double-byte character sets are being processed (e.g. Japanese).

Do NOT process existing entries when Eventlog corruption occurs

When this option is chekced, it prevents from reprocessing of the whole Windows EventLog when it is corrupted or truncated. So EventReporter / MonitorWare Agent don't processes all entries again.

Do NOT process existing entries on Service Startup

When this option is chekced, it prevents from reprocessing of the whole Windows EventLog when the EventReporter / MonitorWare Agent service is restarted.

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 implemtations. 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 perpended 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.

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.

Add Log Type

If checked, then Log types e.g. System, security etc. etc. is perpended to the generated message.

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 perpended 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.

Event Log Types

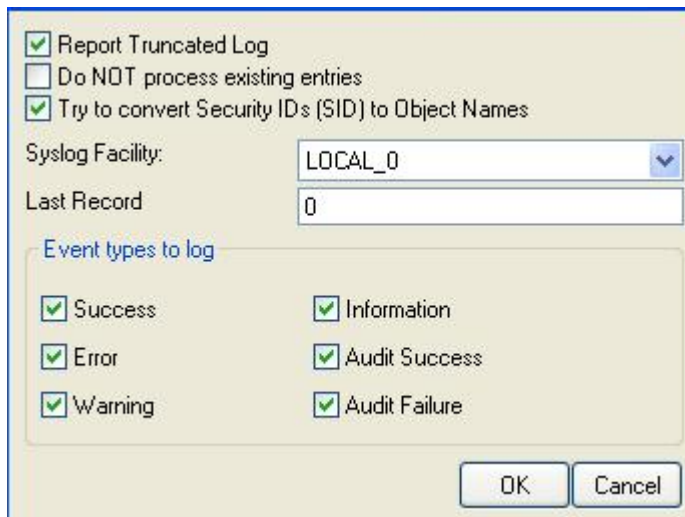**The "Event Log Types"** configure per-event-log settings. The corresponding log will only be processed if the respective "Enable" checkbox is checked. The parameters are common to all logs and will be explained only once. Each dialog looks similar:

Event Log Monitor - EventLog Types Options

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 loosing log entries.

Do not Process Existing Entries

If you don't want to get a dump of an existing specific Windows event log then use this option. When MonitorWare Agent / EventReport are restarted it will start processing after that last entry and will not look for the prevouis entries.

Try to Convert Security IDs (SID) to Object Names

With this option you can convert Security ID's (SIDs) to object names. You can enable this feature in the advanced configuration of each event log type in the Event Log Monitor service. Simple check the "Try to convert Security Ids (SID) to Object Names" option.

Note that only for the Security Event Log this feature is enabled by default. For all other event log types this feature is disabled by default.

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 a valid Rule Set.

Please Note if you intend to make Event ID part of the actual Syslog message while forwarding to a Syslog Server then you have to make some changes in the Event Log Montior Settings. [CLick here](#) to know the settings.

4.3.3   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.

Heartbeat Properties

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 side 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 a valid Rule Set.

4.4   Filter Conditions

4.4.1   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 tree 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:
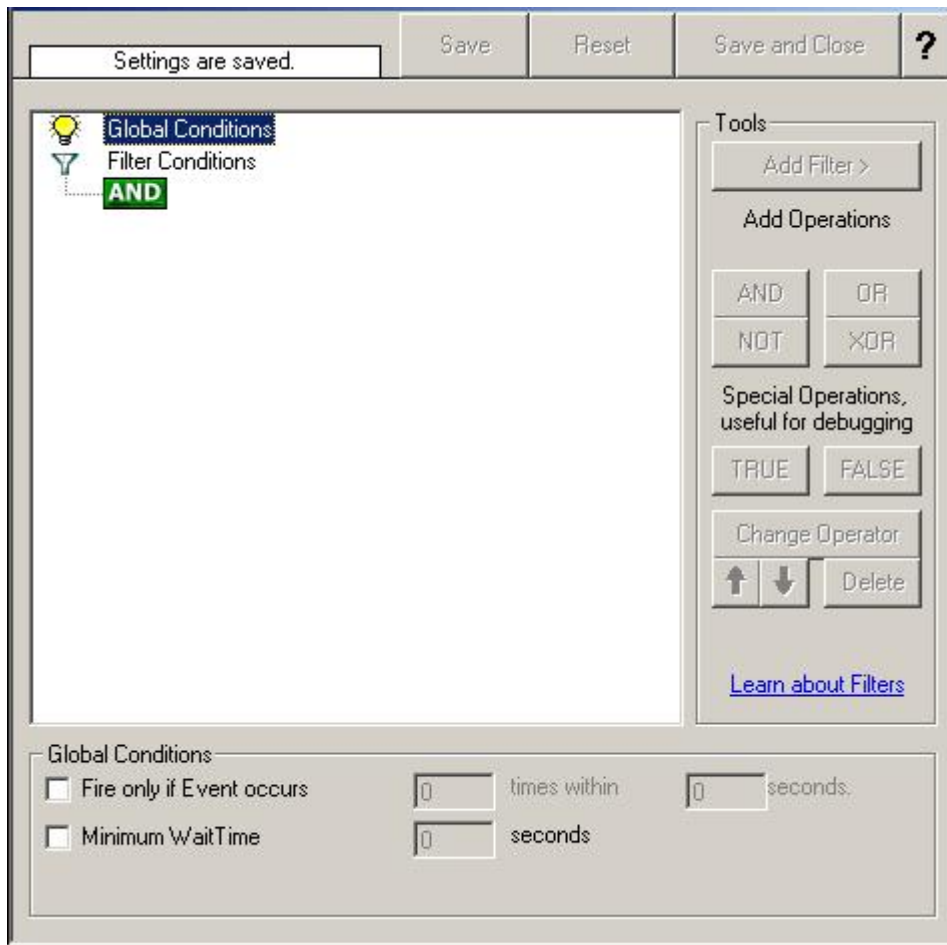
Filter Conditions - Display form

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:
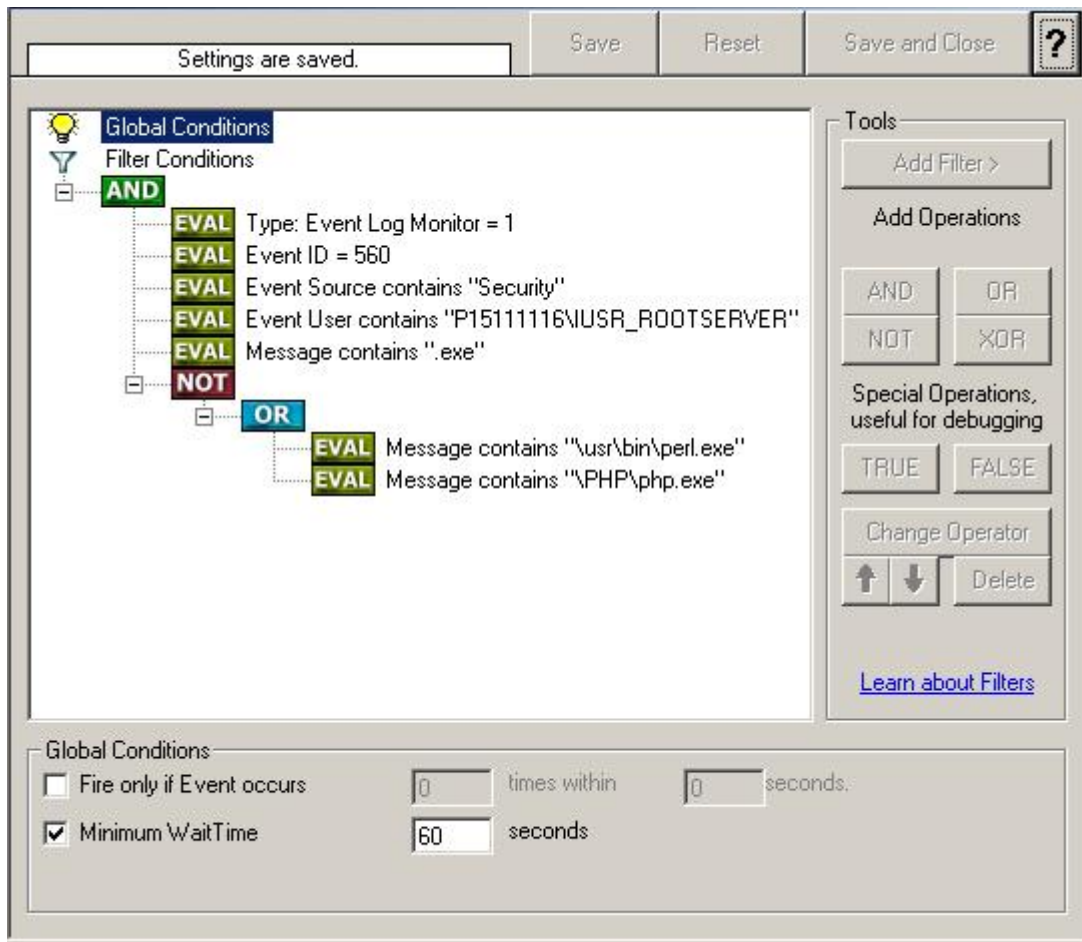
Filter Conditions - Complex Filter

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 (IIS). 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.
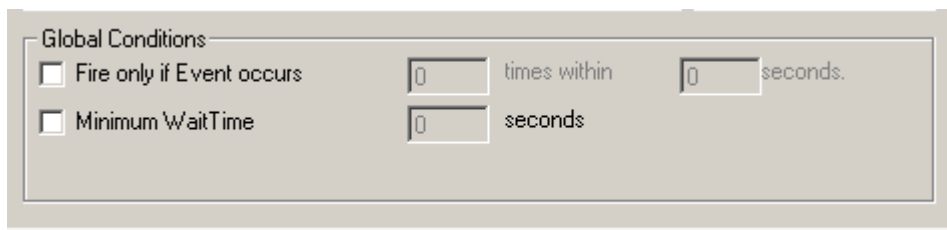
Note: If you want to know more about complex filter conditions you can click on the "Learn about Filters" link.

String comparison in Filter Conditions are "Case Sensitive"! For example, if the Source System name is "ws01" and you had written "WS01" while applying the filter, then this filter condition would **"NEVER"** evaluate to True! Please double check before proceeding further!

If you aren't still sure about what to do, you can drop a word about your requirements to support@adiscon.com, and we will look into it!

4.4.2   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.

Filter Form - General Conditions

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. For 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.

Note: 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 need 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.

4.4.3   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 of the 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.

4.4.4 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.

What happens with Filters that are not available in an "Information Unit"?

Every filter that is not found in an Information Unit will ignored in the filtering process. If you want to create filters specialized for Types of Information Units, always make sure to add an "Information Unit Type" filter.

An example, you have one ruleset, rule and action. In the filters you have one EventID filter. Then you have two Services, one Eventlog Monitor and a Heartbeat monitor both pointing to this ruleset. The Information Units from the Eventlog Monitor would be filtered correctly, but those from the Heartbeat monitor would not be filtered as they don't have an EventID property. The EventID filter would be ignored and the actions would be executed every time.

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.

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 "=".

The List of possible filters, which can be evaluated is described in the upcoming sections.

4.4.5   General

These are non-event log specific settings.
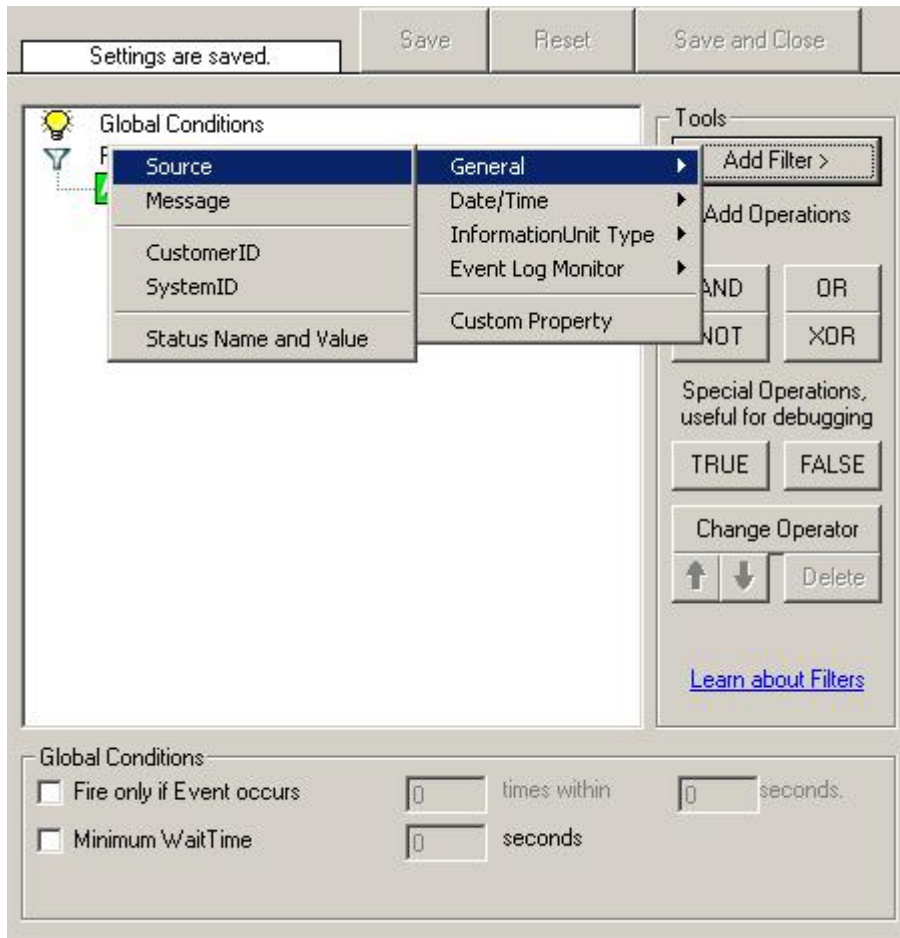
Filter Conditions - General

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 by choosing the "**contains within**

range" compare opertaion. This can be done by specifying the start range and end range into the respective boxes.

Please note that you can enter the character position you desire in these fields. The default "Start Range" and "End Range" are set to 0.

If you would like to search for a string just between positions 10 and 50, specify these values as start and end values, respectively. Similarly if you want to recieve all logs from 192.168.0.1 then set this as:

Property value = 192.168.0.0
Range Start = 0
Range End = 10

Which means 10 charcters starting at zero ("192.168.0."). Please note that the final DOT must be included. If you just used range "9", then 192.168.010 would also match.

This filter is of type string.

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.

CustomerID (Type=Number).

SystemID

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

SystemID (Type=Number).

Status Name and Value

These filter type corresponds to "Set Status" Action. Status Name and Value (Type=String)

4.4.6   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).
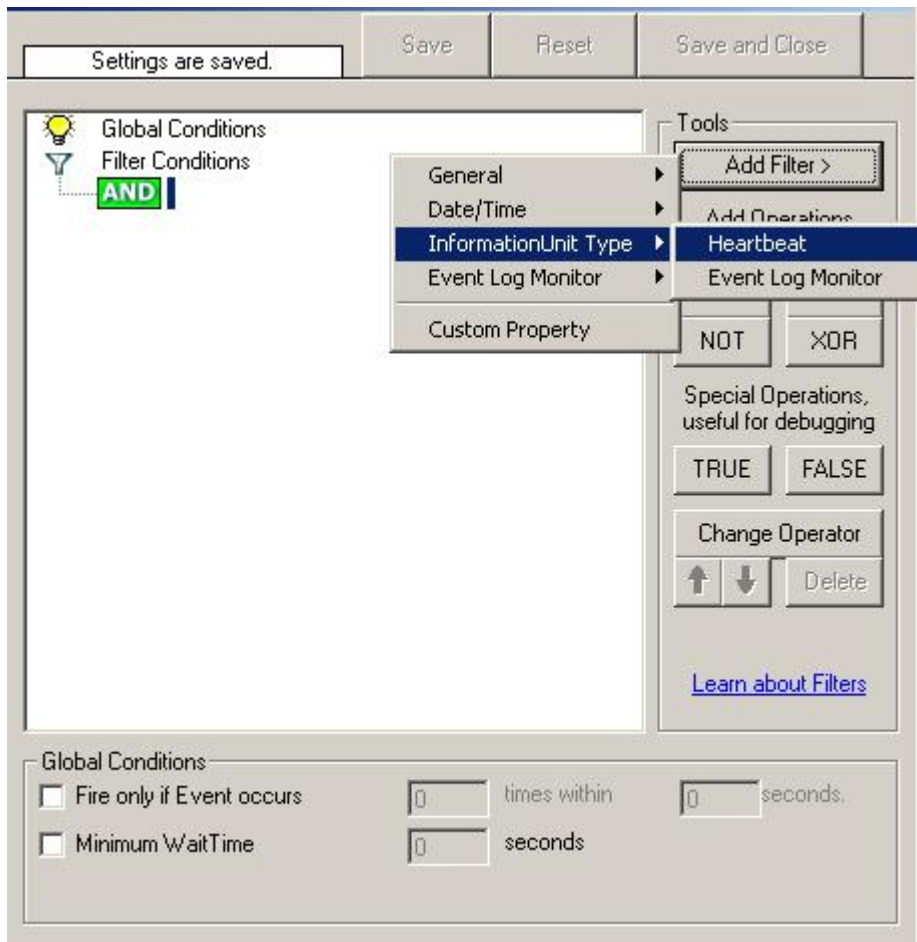
Filter Conditions - InformationUnit Type

The following filters are available:

1.   Heartbeat (Type=Boolean)
2.   Event Log Monitor (Type=Boolean)

4.4.7 Event Log Monitor

Event log monitor specific filters are grouped here.
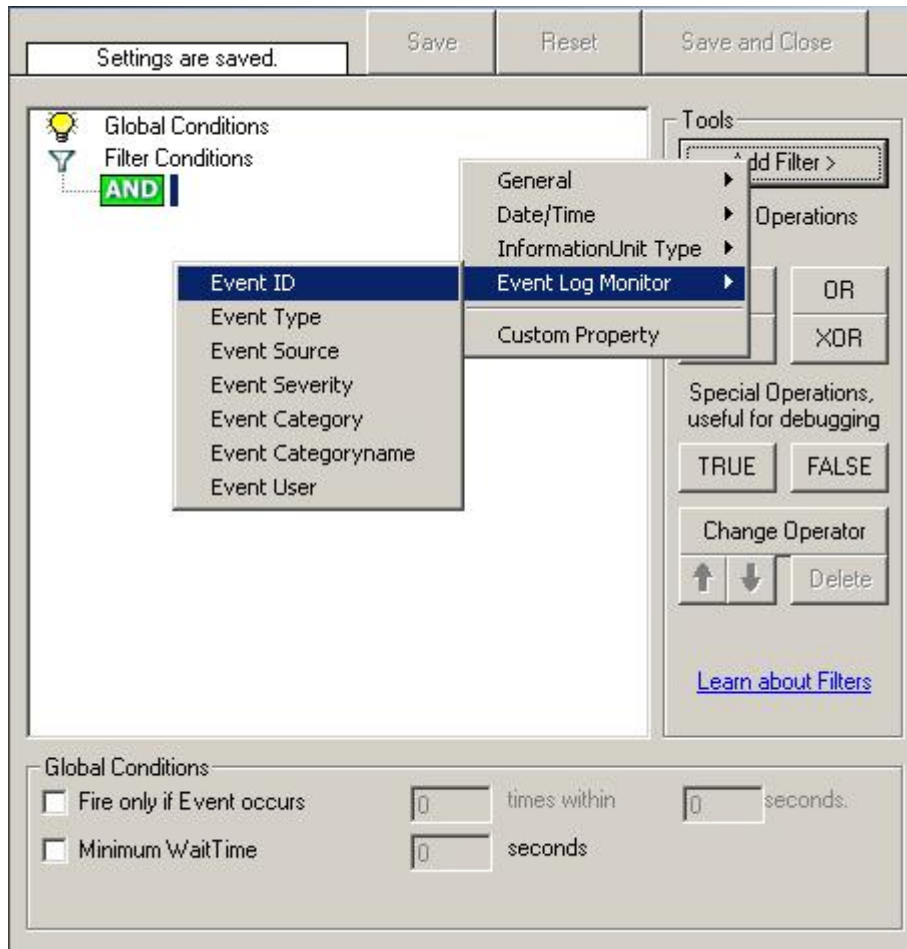
Filter Conditions - Event Log Monitor

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 Categoryname

This value contains the Category value as string if it can be resolved. Otherwise it will contain the category number.

This filter is of type string.

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.

4.4.8   Custom Property

Custom Property specific filter is described here.
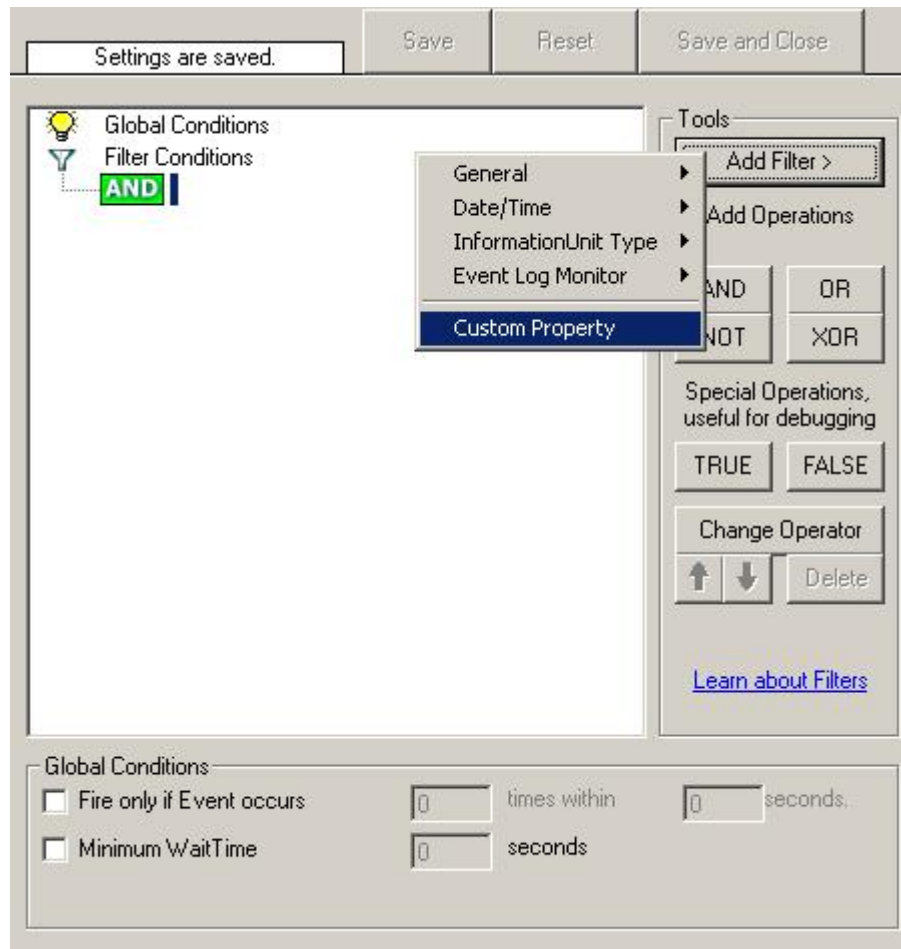
Filter Conditions - Custom Property

Custom Property

As the name suggests it is a "Custom Property". Internally in MonitorWare Agent all values are stored in properties. For example the main message is stored in a property called "msg". By using this dialog you can access properties which are dynamic (Like those from SNMP Trap Monitor when using v2 protocol).

This filter is of type string.

4.5 Actions

4.5.1 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. **However you can change the order of the actions by Moving then Up or Down.**

4.5.2 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.

File Logging Options

Configure For ...

If you want to generate the Reports on log files using Monilog or MonitorWare Console, then its absolutely necessary that the log files are in a specific format. This option allows you to configure the File Logging format for Monilog and MonitorWare Console.

If the log file entries are not in the correct format for MonitorWare Console (for PIX or Windows Reports), then it will write error messages for first 50 lines in Windows Event Log and will ignore them for the generation of report, resulting in a generation of empty report.

And, if the log file entries are not in the correct format for Monilog, then an empty report woud be generated.

Following three options are available:

1.   Configure for MonitorWare Console PIX Reports
2.   Configure for MonitorWare Console Windows Reports

3.   Configure for Monilog

Configure for MonitorWare Console PIX Reports

This option changes the File Logging format of MonitorWare Agent to the correct fromat expected by MonitorWare Console for PIX Report generation.

Configure for MonitorWare Console Windows Reports

This option changes the File Logging format of MonitorWare Agent to the correct fromat expected by MonitorWare Console for Windows Report generation.

Configure for Monilog

This option changes the File Logging format of MonitorWare Agent (i.e. custom line format) to the correct fromat expected by Monilog for Report generation.

Enable Property replacements in Filename

By activating this option, you can use properties within the file or pathname like %Source% and all the others. For example:

File Path Name can be **F:\syslogs\%source%**
File Base Name can be **IIS-%source%**

If your source is 10.0.0.1, that will write the following file:

F:\syslogs\10.0.0.1\IIS-10.0.0.1.log

Please note that the path f:\syslogs\10.0.0.1 was generated because the source poperty was used inside the path.

Note: You can use ANY property inside the path and base name. Event properties are described in the <u>property replacer section</u>.

File Path Name

The base path (directory) of the file. Please see above for exact placement. Default is "c:\temp". The Insert Menu entry allows you to create "**Dynamic Directories**". For example:

File Path Name can be **F:\syslogs\%source%**

Event properties are described in the <u>property replacer section</u>.

File Base Name

The base name of the file. Please see above for exact placement. Default is "MonitorWare". The Insert Menu entry allows you to reate "Dynamic Base Filenames". For example:

File Base Name can be **IIS-%source%**

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" format 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.

The "Custom" format allows you to customize formats to increase log file compatibility for third party applications. When you choose this option then Custom Line Format is enabled.

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.

Custom Line Format

Custime Line Format enables you to fully customize the output for the log file. The Insert Menu entry provides further options and it only works in custom line format. Default value is "%msg%%$CRLF%".

Event properties are described in the <u>property replacer section</u>.

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.

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 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.

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.

4.5.3   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.

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), Microsoft SQL Server and MySQL. 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.

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.

Enable Detail Property Logging

If this option is enabled, additional detail data will be logged into the SystemEventsProperties table.

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.

Please note that The message content of the Message field can now be configured. Event properties are described in the <u>property replacer section</u>.

4.5.4   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.

Custom Event Source

EventSource is now fully configurable with all posibilities the Property Engine gives you. **Please note that content of this field can be configured. Event properties are described in the <u>property replacer section</u>.**

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.

Message to Log

It is the message which will be logged into the Windows Event Log. It is fully configurable what is logged into the Eventlog.

Please note that Insert Menu entry allows you to add replacement characters e.g. %msg% - you can write the actual message of an event into the Windows Event Log.

Please note that The message content of the Message field can be configured. Event properties are described in the <u>property replacer section</u>.

4.5.5   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

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).

Please note that Insert Menu entry allows you to add replacement characters e.g. %msg% - you can send out the actual message of an event in the subject line.

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.

Please note that The message content of the Message field can be configured. Event properties are described in the <u>property replacer section</u>.

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 replacment 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:

%s	IP address or name (depending on the "resolve hostnames" setting) of the source system that sent the message.
%f	numeric facility code of the received message
%p	numeric priority code of the received message
%m	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.
%%	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 "Mesg: '%msg:1:15%' From: %fromhost%". If the message "This is a lengthy test message" were receive from "172.16.0.1", the resulting email subject would read: "Mesg: 'This is a lengt' From: 172.16.0.1". Please note that the message is truncated because you only extracted the first 15 charactes 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 miliseconds, 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 <u>non-legacy mode</u>.

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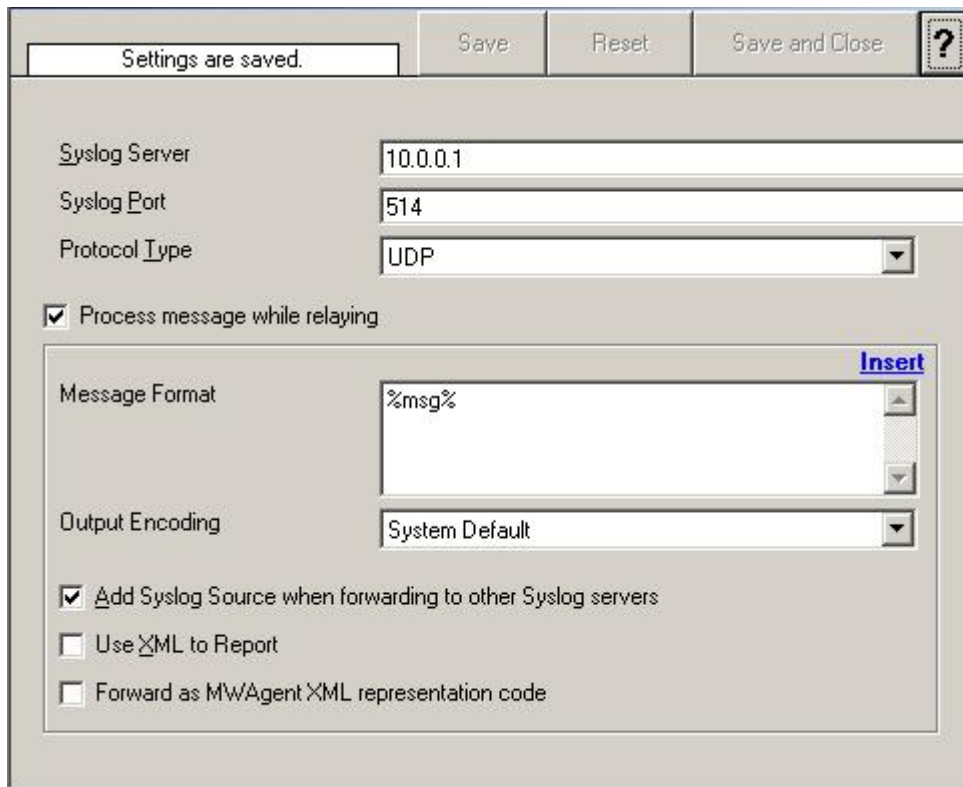
If checked, the received event will be included in XML format in the mail. If so, the event will include **all** information, like the original timestamp, the facility, priority etc. XML format is especially useful if the mail is sent to an automated system, which will

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.

4.5.6   Forward Syslog Options

This dialog controls Syslog forwarding options.

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 [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. MonitorWare Agent also can receive Syslog messages via TCP and reliable Syslog messages via TCP using the new [RFC 3195](#) standard.

Message Format

You can change the message format. By default the original message is forwarded.

Please note that The message content of the Message field can now be configured. Event properties are described in the property replacer section.

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.
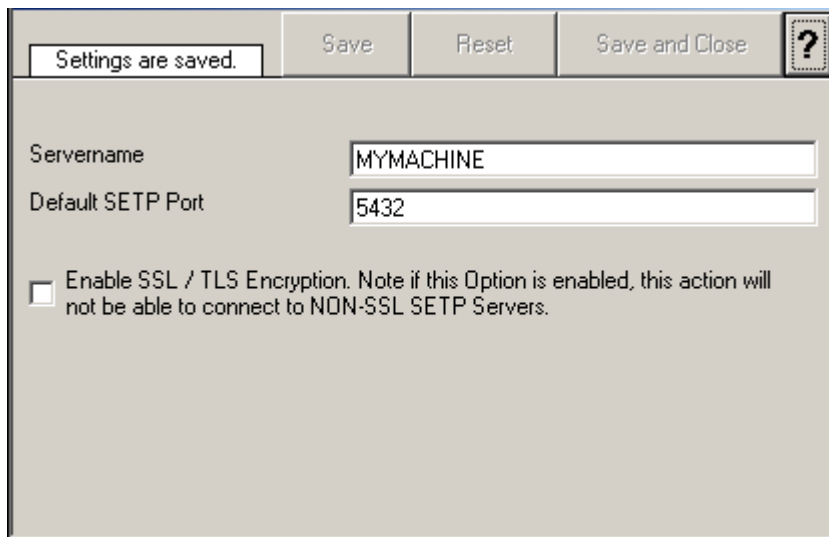
Forward as MW Agent XML Representation Code

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. **Please note that this option is only "experimental" and is not an official standard.**

Please note you can also make Event ID part of the actual Syslog message while forwarding to a Syslog Server then you have to make some changes in the Forward Syslog Action. Click here to know the settings.

4.5.7 Forward SETP Options

This dialog controls the Send options. With the "Send SETP" action, messages can be sent to a SETP server.

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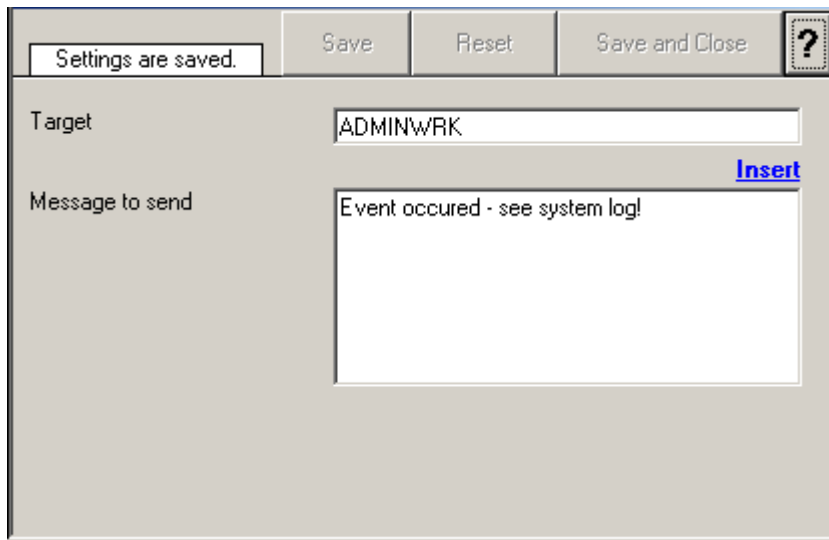
If this option is enabled then this action will be able to connect to SSL/TLS SETP servers. Please make sure that you want this option to be enabled.

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

4.5.8   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".

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.

Please note that The message content of the Message to send field can now be configured. Event properties are described in the <u>property replacer section</u>.

4.5.9   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.
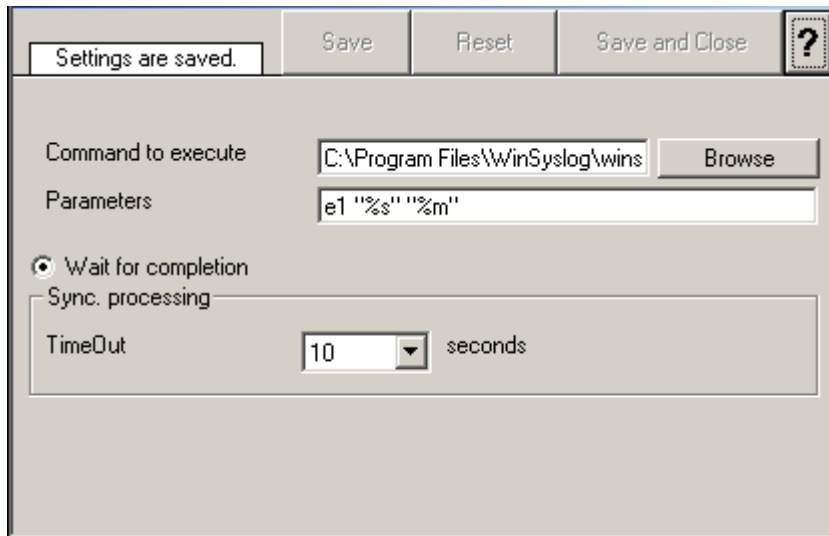
Start Program Dialog

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:

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

4.5.10 Play Sound

This action allows you to play a sound file.
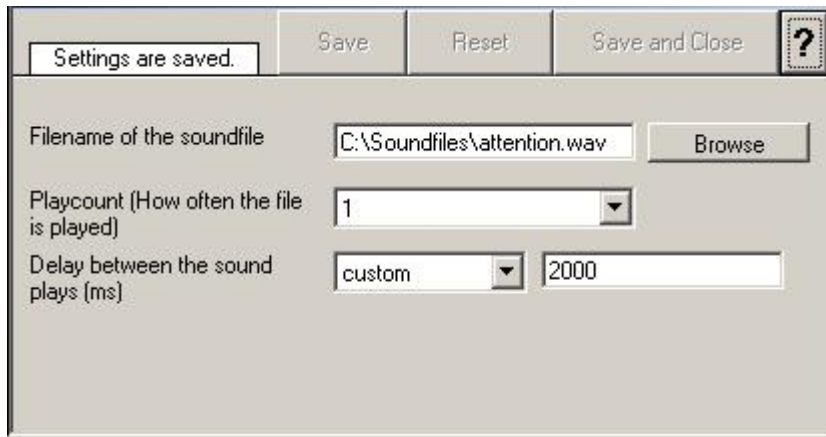
Play Sound Dialog

Please note: if your machine has multiple sound cards installed, the "Play Sound" action will always use the card, that was installed first into the system, only.

However there is a work around if you want to use [Play Sound Action](#) for a second sound card!

Filename of the Soundfile

Please enter the name of the sound file to play. **This must be a .WAV file**, other formats (like MP3) are **not** supported. While in theory it is possible that the sound file resides on a different machine, we highly recommend using files on the local machine only. Using remote files is officially not supported (but currently doable if you are prepared for some extra effort in getting this going).

If the file can either not be found or is not a valid format, a system beep is emited instead (this should - by API definition - be possible on any system).

Playcount

This specifies how many times the file is played. It can be re-played up to a hundered times.

Please note: Playing sounds is performance intense and MonitorWare Agent will block other actions while sounds are being played. As such, we recommend to limit the duration and repeat count of sounds played.

Delay between Plays

If multiple repeats are specified, this is the amount of time that is waited between each individual play.

4.5.11   Send to Communications Port

This action allows you to send a string to an attached communications device, that is it sends a message through a Serial Port.

Send to Communications Port Options

Timeout Limit

The maximum time allowed for the device to accept the message. If the message could not be send within that period, the action is aborted. Depending on the device, it may be left in an unstable state.

Port to Send To

Specify the port to which your device is being attached. Typically, this should be one of the COMx: ports. The listbox shows all ports that have been found on your local machine. You may need to adjust this to a different value if you are configuring a remote machine.

1. MSFAX
2. COM1
3. COM2
4. COM3
5. COM4
6. FILE
7. LPT1
8. LPT2
9. LPT3
10. AVMISDN1
11. AVMISDN2
12. AVMISDN3
13. AVMISDN4
14. AVMISDN5
15. AVMISDN6
16. AVMISDN7
17. AVMISDN8
18. AVMISDN9

Port Settings

Use those settings that your device expects. Please consult your device manual if in doubt.

Port Settings

These settings must be set as expected by your device. If in doubt, consult your device manual.

Bits per Seconds

Bits per second can be 110 and go up to 256000, by default 57600 is selected.

Databits

Databits define the number of bits in the bytes transmitted and received.

Parity

With Parity you can configure the Parity scheme to be used. This can be one of the following values:

1. Even
2. Mark
3. No parity
4. Odd
5. Space

Stop bits

You can configure the Number of stop bits to be used. This member can be one of the following values:

1.   1 stop bit
2.   1.5 stop bits
3.   2 stop bits

DTR Control Flow

DTR (data-terminal-ready) flow control. This member can be one of the following values:

1.   DTR_CONTROL_DISABLE - Disables the DTR line when the device is opened and leaves it disabled.
2.   DTR_CONTROL_ENABLE - Enables the DTR line when the device is opened and leaves it on.
3.   DTR_CONTROL_HANDSHAKE - Enables DTR handshaking.

RTS Control Flow

RTS (request-to-send) flow control. This member can be one of the following values:

1.   RTS_CONTROL_DISABLE - Disables the RTS line when the device is opened and leaves it disabled.
2.   RTS_CONTROL_ENABLE - Enables the RTS line when the device is opened and leaves it on.
3.   RTS_CONTROL_HANDSHAKE - Enables RTS handshaking. The driver raises the RTS line when the "type-ahead" (input) buffer is less than one-half full and lowers the RTS line when the buffer is more than three-quarters full.
4.   RTS_CONTROL_TOGGLE - Specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be low.

Message to Send

This is the message that is to be send to the device. You can enter text plainly and you can also include all properties from the current event. For example, if you have a serial audit printer and you would just plainly like to log arrived messages to that printer, you could use the string "%msg%%$CRLF%" to write the actual message arrived plus a CRLF (line feed) sequence to the printer.

Please note that The message content of the Message field can now be configured. Event properties are described in the <u>property replacer section</u>.
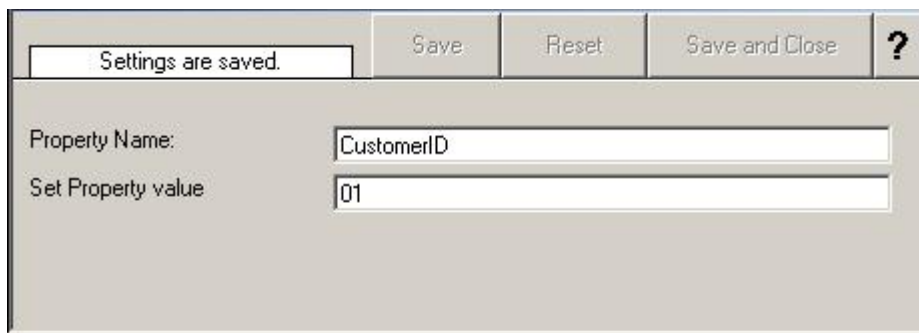
4.5.12 Set Status

This dialog controls the set status options.

Each Information unit have specific Properties e.g. EcentID, Priority, Facility etc. These properties have some values. Lets suppose that EventID has Property value 01. Now If you want to add "**a new property of your own choice**" in the existing set of properties then Set Status action allows you to accomplish this!

You can Create a new property and assign the any valid desired value to it e.g. we had created a new Property as CustomerID and set its value to 01 in the screen-shot below. After you have created the property through this action, then you can define filters for them. There is an internal Status List within the product which you can use for more complex filtering.

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!

Set Status Dialog

Property Name

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

Set Property Type

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

4.5.13 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!

Set Property Dialog

Select Property Type

Select the property type to be changed. The list box contains all properties that can be changed. By default it is set to "Source System". Available options are:

1.   SourceSystem
2.   Importance
3.   SyslogFacility
4.   SyslogPriority

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 use in the filter conditions and actions.
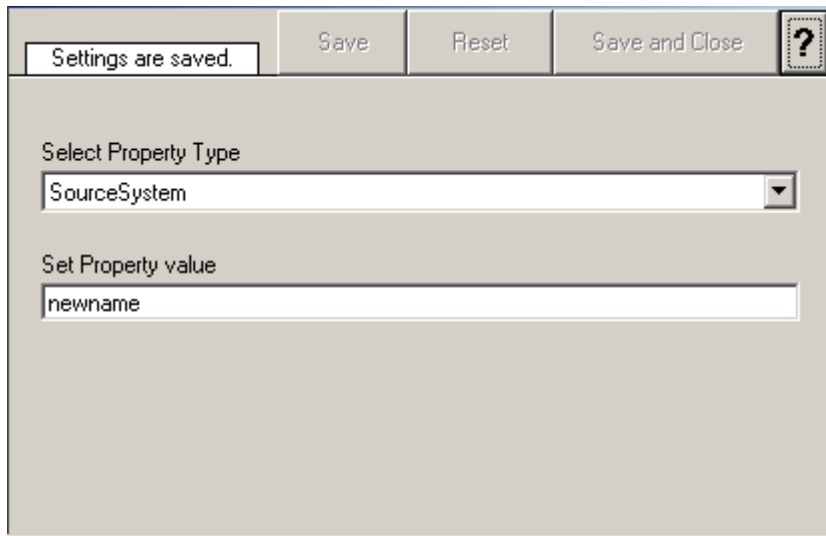
4.5.14 Call RuleSet

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 go 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 Rule 1 has two actions - Action 1 and Action 2. The Action 1 of Rule 1 is an include (Call Ruleset) 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.
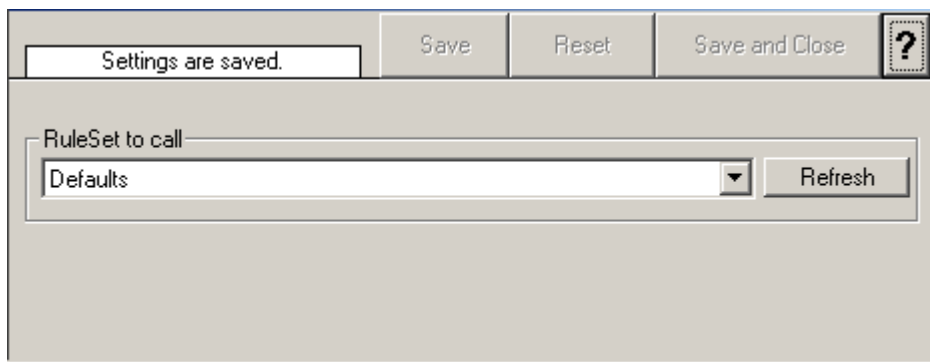
Call Ruleset Dialog

Ruleset to Call

Select the Ruleset to be called.

Note Call RuleSet stays disabled until you have more then "One" RuleSet!

4.5.15 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.

Click here to see an example about the Discard Action.

5 Getting Help

The EventReporter 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.eventreporter.com/en/FAQ/. The FAQ area is continuously being updated

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/forum,3.html.

Customer Support System

Our customers service and support system is available at http://custservice.adiscon.com. With it, you can quickly open a support ticket via a web-based interface. This system can be used to place both technical support calls as well as general and sales questions. We would appreciate if you select the appropriate category when opening your ticket.

Please note: the customer service system asks you for a userid and password when you open it. If you do not have a userid yet, you can simply follow the "register" link (in the text part) to create one. You can also open a ticket without registering first, in which case the system will create one for you. You will receive the generated userid as part of the email notifications the system generates.

Why using the customer support system? As you see further below, we also offer support by email. In fact, email is just another way to create a ticket in the customer support system. Whenever we reply to your ticket, the system automatically generates an email notification, which includes a link to your ticket as well as the answer we have provided. So for the most cases, you can use email, only. However, there are some situations where the support system should be used:

- **Email notifications do NOT include attachments!** If we provide an attachment, you must login into the ticket in order to obtain this. For your convenience, each email notification contains an active link that allows you to login immediately.
- **If you seem to not receive responses from us, it is a very good idea to check the web interface.** Unfortunately, anti-SPAM measures are being setup more and more agressive. We are noticing an increasing number of replies that simply do not make it to your mailbox, because some SPAM filter considered it to be SPAM and removed it. Also, it may happen that your support question actually did not get past our own SPAM filter. We try very hard to avoid this. If we discard mail, we send a notification of this, so you should at least have an indication that your mail did not reach us. Using the customer support system via its own web interface removes all SPAM troubles. So we highly recommend doing this if communication otherwise seems to be distrubed. In this case, please remember that notification emails may also get lost, so it is a good idea to check your ticket for status updates from time to time.

MonitorWare Agent Web Site

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

Email

Please address all support requests to support@adiscon.com. An appropriate subject line is highly appreciated.

Please note: we have increasingly seen problems with too-agressive SPAM filtering, resulting in loss of our replies. If you do not receive a response from us within two working days, we highly recommend re-submitting your support call via the customer support system.

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 **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.

Click here to learn more about UpgradeInsurance.

Non-Technical Questions

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

Please note: we have increasingly often problems with too-agressive SPAM filtering, resulting in loss of our replies. If you do not receive a response from us within two working days latest, we highly recommend re-submitting your question via the customer support system.

Product Updates

The MonitorWare line of products is being developed since 1996. New versions and enhancements are made available continuously.

Please visit www.eventreporter.com for information about new and updated products.

6 Purchasing EventReporter

All EventReporter 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.

Pricing & Ordering

Please visit http://www.eventreporter.com/en/intermediate-order.php 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.

7   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 EventReporter Service
- Support for Mass Rollouts
- Formats (XML and Database)
- Version History
- Property Replacer

Note: Please go through the Formats (XML and Database) specifically "Database Formats", sometimes looking into it can solve your problems!

7.1   Comparison of properties Available in MonitorWare Agent, EventReporter and WinSyslog

The property replacer is a reference - the actual properties are very depending on the edition purchased. We have just included information on what is available in which products for your ease and convenience.

Properties Available	**MonitorWare Agent**	**WinSyslog**
Standard Property	Yes	Yes
Windows Event Log Properties	Yes	
Syslog Message Properties	Yes	Yes
Disk Space Monitor	Yes	
File Monitor	Yes	
Windows Service Monitor	Yes	
Ping Probe	Yes	
Port Probe	Yes	
Database Monitor	Yes	
Serial Port Monitor	Yes	
MonitorWare Echo Request	Yes	
System Properties	Yes	Yes
Custom Properties	Yes	Yes

7.2   Event Properties

Events have certain properties, for example the message associated with the event or the time it was generated. Each of this properties has an assigned name. The actual properties available depend on the type of event. The following sections desribe both how to access properties as well as properties available.

Knowing about event properties is important for building complex filter conditions, customized actions as well as for integrating into a third-party system. Event properties provide a generic way to look at and process the events generated. Thus we highly recommend that you at least briefly read this reference section.

7.2.1   Acessing Properties

Properties are accessed by their name. The component used for this is called the "property replacer" It 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.

The property replacer provides very powerful ways to acces the properties: they can not only be accessed as one full property. They can also be acceessed as substrings

and even be reformatted. As such, the property replacer provides a specifc syntax to access properties:

%property:fromPos:toPos:options%

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

FromPos and ToPos can be used to copy a substring from a lenghty property. The options allow to specify some additional formating.

7.2.1.1 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 event property, a custom property, a dynamic property or a system property.

If a property is selected that is **not** present, the result will always be an empty string, no matter which other options have been selected.

7.2.1.2 FromPos

If you do not want to use the full string from the property, you can specify a start position here. There are two ways to specify the start location:

Fixed Character position

If you know exactly on which position the string of interest begins, you can use a fixed location. In this case, simply specify the character position containing the first character of interest. Character positions are counted at 1.

Search Pattern

A search pattern is specified as follows:

/<search-pattern>/<options>

If a search pattern is specified, the property value is examined and the first occurence of <search-pattern> is detected. If it is not found, nothing is returned. If it is found, the position where the pattern is found is the start postion or, if the option "$" is specified, the position immediately after the pattern.

The search pattern may contain the "?" wildcard character, which represents any character. Other wildcards are not supported with the property replacer.

Default Value

If the FromPost is not specified, the property string is copied starting at position 1.

7.2.1.3 ToPos

If you do not want to use the full string from the property, you can specify the highest character position to be copied here.

Absolute Position

Specify a simple integer if you would like to specify an absolute ending position.

Relative Position

This is most useful together with the serach capabilities of "FromPos". A relative position allows you to specify how many characters before or after the FromPos you would like to have copied. Relative positions are specified by putting a plus or minus ("+"/"-") in front of the integer.

Please note: if you specify a negative postion (e.g. -20), FromPos and ToPos will internally be swapped. That is the property value will not be (somehow) reversely copied but they will be in right order. For example, if you specify %msg:30:-20% actually character positons 10 to 30 will be copied.

Default

If not specified, the ending position will be the last character.

7.2.1.4 Options

Options allow you to modify the the contents of the property. Multiple options can be set. They are comma-separated. If conflicting options are specified, always the last option will be in effect (e.g. specifying "uppercase,lowercase" will lead to lowercase conversion of the property value).

The following options are available with this release of the product:

lowercase	All characters in the resulting property extract will be converted to lower case.
uppercase	All characters in the resulting property extract will be converted to upper case.
uxTimeStamp	This is a special switch for date conversions. It only works if the extracted property value is an ISO-like timestamp (YYYY-MM-DD HH:MM:SS). If so, it will be converted to a Unix-like ctime() timestamp. If the extracted property value is not an ISO-like timestamp, no conversion happens.

Important: All option values are case-sensitive. So "uxTimeStamp" works while

"uxtimestamp" is an invalid option!

7.2.1.5 Examples

Simple 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.

More complex Examples

If you would like to extract the 50 characters from the message after the word DROP, you would use the following replacer string:
%msg:/DROP/$:+50%

If you would like to have the first 40 characters in front of the string "- aborted" (including that string):
%msg:/- aborted/$:-40%

And if you would like to receive everything starting from (and including) "Log:":
%msg:/Log%

If you would just like to log lowercase letters in your log messages:
%msg:::lowercase%

And if you would just like to have the first 50 characters (and these in lower case):
%msg:50:::lowercase%

If you need to change a timestamp to a UNIX-like timestamp, you could use this:
%datereceived:::uxTimeStamp%

A real world Sample
We use the following template to generate output suitable as input for MoniLog:

%timegenerated:1:10%,%timegenerated:12:19%,%source%,%syslogfacility%,%syslogpriority%,EvntSlog: %severity% %timereported:::uxTimeStamp%:
%source%/%sourceproc% (%id%) - "%msg%"%$CRLF%

Please note: everything is on one line with no line breaks in between. This example is from the "write to file" action (with custom file format).

7.2.2 System Properties

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

$CRLF	A Windows newline sequence consisting in the characters CR and LF. If you just need CR, you can use %$CRLF:1:1% and if you need use LF you can use %$CRLF:2:2%
$TAB	An US-ASCII horizontal tab (HT, 0x09) character
$HT	same as $TAB
$CR	A single US-ACSII CR character (shortcut for %$CRLF:1:1%)
$LF	A single US-ACSII LF character (shortcut for %$CRLF:2:2%)
$xNN	A single character, whoms value (in hexadecimal) is given by NN. NN **must** be two hexadecimal digits - a leading zero must be used if a value below 16 is to be represented. The value 0 (%x00) is invalid and - if specified - replaced by the "?" character. As an example, $CR could also be expressed as %$x0d%. Please note that only **one** character can be represented. If you need to specify multiple characters, you need multiple $xNN sequences. An example may be $CRLF which could also be specified as %$x0d%%$x0a% (but **not** as %$x0d0a%).

7.2.3 Custom Properties

Users can create an unlimited number of custom properties. These can be created with for example the "PostProcess" action (if the product edition purchased supports this action).

Custom properties can theoretically have any name, but Adiscon highly recommends to prefix them with "u-" (e.g. "u-MyProperty" - "u" like "user"). This ensures that no compatibility problems will arise in current and future versions of the software. Adiscon guarantees that it will never use the "u-" prefix for Adiscon-assigned properties.

Custom properties can be used just like regular properties. Wherever you can specify a property, you can also specify a custom property.

7.2.4 Event-Specific Properties

Each network event is represented by a so-called "Event Record" (sometime also named an "InfoUnit", an "Unit of Information"). Data obtained from all services will end up as an event. For example, Windows Event Log data, syslog data and a file line obtained by the file monitor will all be an event. That kind of generalization make it easy to deal with all of these events in a consistent way.

Each event has a set of properties which in turn have values. For example, there is a property named "source" and it will always contain an indication of which system the

event orginated on. Obviously, not every event source does support all properties. For example, a syslog message does not contain a Windows NT Event ID - simply because there is no such thing as an event ID in syslog. So, depending on the type of event, it may contain different properties.

In order to make the product really generally useful, some few properties have been defined in a generic way and are guarantted to be present in every event, no matter what type it may have. Sometimes this is a "natural" common property, like the "fromhost". Sometimes, though, it may look a bit artificial. An example of the later is the "syslogfacility" property. It is guaranteed to be present in every event - but actually this is a syslog-only thing. The non-syslog event sources either emulate this property (in a consistent manner) or allow the user to configure a syslogfacility that should be used for all events generated by that service. At the bottom line, this will ensure that the property is available in all events and - given proper configuration - that can be extremely helpful for the administrators to set up things in a powerful and generic way.

7.2.4.1 Standard Properties

As outlined under <u>Event Properties</u>, these are properties present in all types of events. Some event types have only these standard properties. Others have additional properties. Those with additional properties are documented in the other sections. If there is no specific documentation for a specific event type, this means that it supports the standard properties, only.

msgPropertyDescribed	A human-readable representation of the message text. While this is genereally available, the exact contents largely depends on the source of the information. For example, for a file monitor it contains the file line and for a syslog message it contains the parsed part of the syslog message.
source	The source system the message orignated from. This can be in various representations (e.g. IP address or DNS name) depending on configuration settings.
syslogpriority	The severity of a syslog message. For non-syslog messages, this should be a close approximation to what a syslog severity code means.
syslogfacility	The facility of a syslog message. For non-syslog messages, the value is provided based on configuration. In essence, this is simply an integer value that can be used for quick filtering inside your rules.
syslogtag	The syslog tag value, a short string. For non-syslog messages, this is provided based on configuration. In most cases, this is used for filtering.
resource	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
CustomerID	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
SystemID	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
timereported	The time the originator tells us when this message was reported. For example, for syslog this is the timestamp from the syslog message (if not configured otherwise). Please note that timereported eventually is incorrect or inconsistent with local system time - as it depends on external devices, which may not be properly synchronized. For Windows Event Log events, timereported contains the timestamp from the event log record.
timegenerated	The time the event was recorded by the service. If messages are forwarded via SETP, this timestamp remains intact.
importance	Reserved for future use.
iut	Indicates the type of the event. Possible values are: 1- syslog message 2- hearbeat 3- Windows Event Log Entry 4- SNMP trap message 5- file monitor 8- ping probe 9- port probe 10- Windows service monitor 11- disk space momitor 12- database monitor 13- serial device monitor
iuvers	Version of the event record (info unit). This is a monitorware interal version identifier.

7.2.4.2 Windows Event Log Properties

id	Windows Event ID
severity	severity as indicated in the event log. This is represented in string form. Possible values are: [INF] - informational [AUS] - Audit Success [AUF] - Audit failure [WRN] - Warning [ERR] - Error [NON] - Success (called "NON" for historical reasons)
severityid	The serverity encoded as a numerical entity (like in Windows API)
sourceproc	The process that wrote the event record (called "source" in Windows event viewer).
category	The category ID from the Windows event log record. This is a numerical value. The actual value is depending on the event source.
user	The user name that was recorded in the Windows event log. This is "N\A" if no user was recorded.
NTEventLogType	The name of the Windows event log this event is from (for example "System" or "Security").
bdata	Windows event log records sometimes contain binary data. The event log monitor service can be set to include this binary data into the event, if it is present. If it is configured to do so, the binary data is put into the "bdata" property. Every byte of binary data is represented by two heaxdecimal characters. Please note that it is likely for bdata **not** to be present. This is because the binary data is seldomly used and very performance-intense.

7.2.4.3 Syslog Message Properties

rawmessage	The message as it was received from the wire (unparsed).

7.2.4.4 Disk Space Monitor

currusage	The currently used disk space.
maxavailable	The overall capacity of the (logical) disk drive.

7.2.4.5 File Monitor

genericfilename	The configured generic name of the file being reported.

7.2.4.6 Windows Service Monitor

sourceproc	The name of the service whoms status is being reported (from the Windows service registry).

7.2.4.7 Ping Probe

echostatus	Status returend for the echo request
roundtriptime	Round trip time for the ping packet (if successful)

7.2.4.8 Port Probe

responsestatus	The status of the probe.
responsemsg	The response message received (if any)

7.2.4.9 Database Monitor

Database-Monitor created events are a bit different than other events. The reason is that the database fields themselves become properties - but obviously these are not fixed but depend on what you monitor.

All queried data fields are available as properties via their database field name **prefixed with "db-"**.

An example to clarify: we assume the following select statement is used for the database monitor:

select name, street, zip, city from addresses

There is also an ID column named "ID". So the event generated by this database monitor will have the following specific properties:

- db-ID
- db-name
- db-street
- db-zip
- db-city

These properties will contain the field values as they are stored in the database. Please note that NULL values are translated into empty strings (""), so there is no way to differentiate a NULL value from an empty string with this version of the database monitor.

Other than the custom "db-" properties, no specific database monitor properties exist.

7.2.4.10 Serial Monitor

| **portname** | The name of the port that the data orginated from (typical examples are COM1, COM2). The actual name is taken from the configuration settings (case is also taken from there). |

7.2.4.11 MonitorWare Echo Request

| **responsestatus** | The status of the echo request. Possible values:
0 - request failed (probed system not alive)
1 - request succeeded

If the request failed, additional information can be found in the *msg* standard property. |

7.3 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 Micorosoft 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 provinding 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.

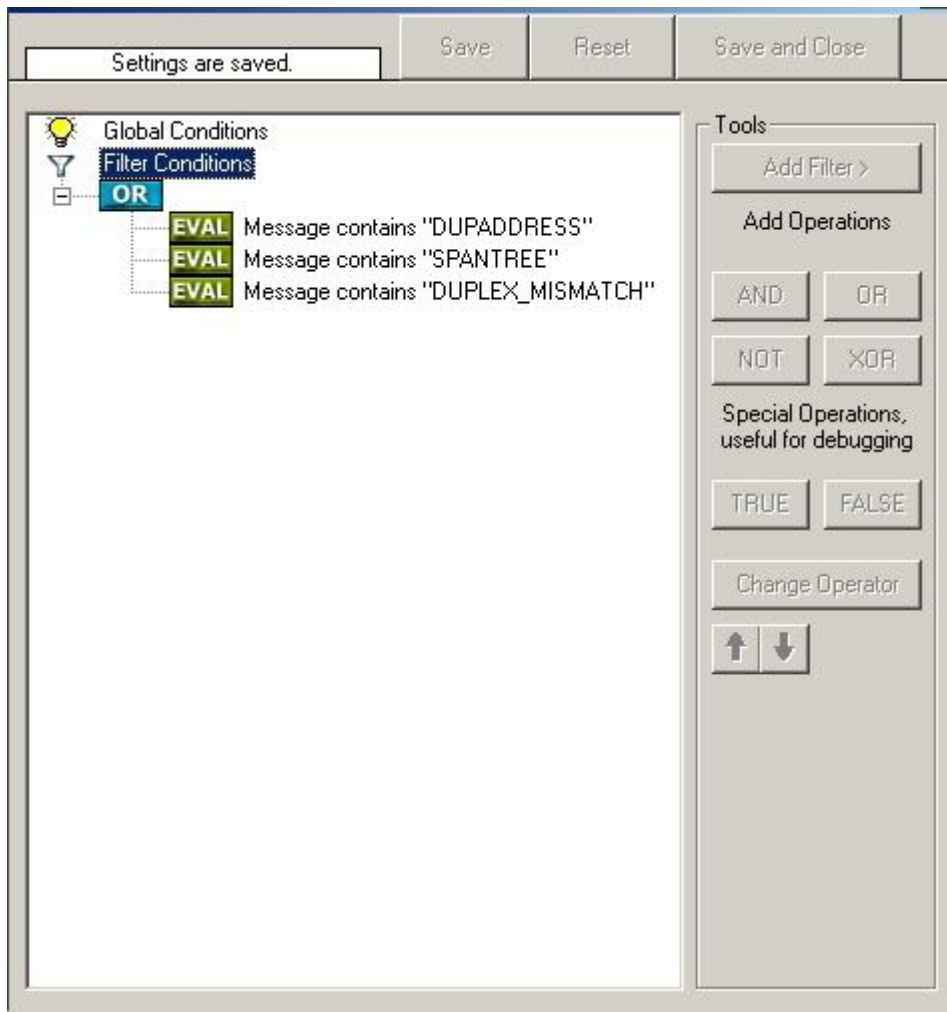In the filter dialog, this pseudo code looks as follows:

Figure 1 - Example 1

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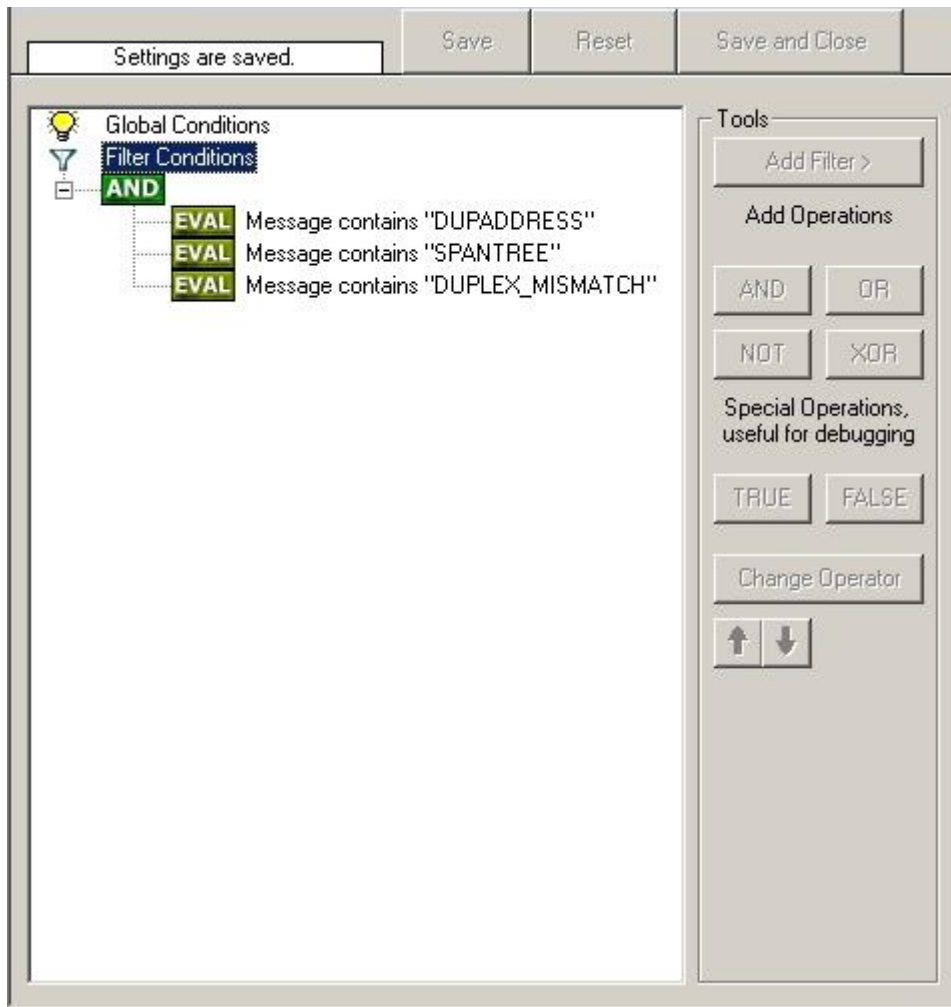
In the filter dialog, this pseudo code looks as follows:

Figure 2 - Example 2

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

In the filter dialog, this pseudo code looks as follows:
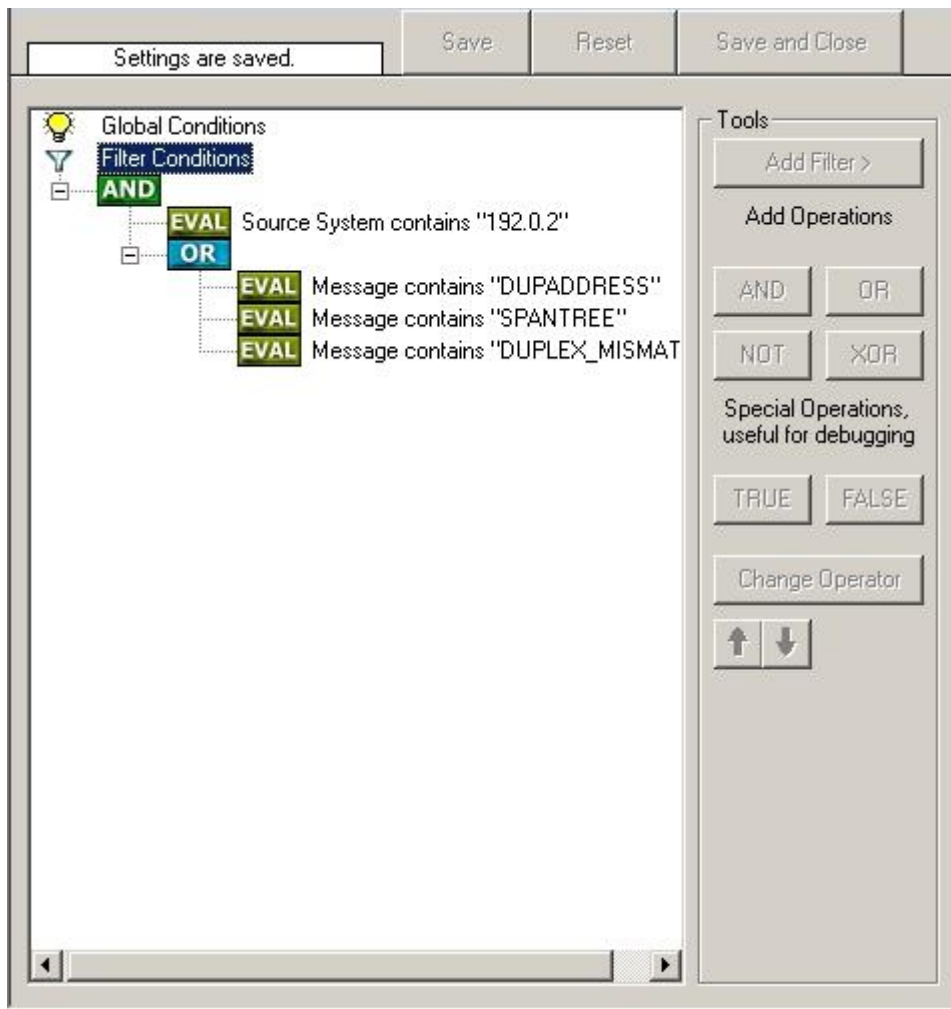
Figure 3 - Example 3

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](#)

7.4   EventReporter Shortcut Keys

Use shortcut keys as an alternative to the mouse when working in EventReporter Client. Keyboard shortcuts may also make it easier for you to interact with EventReporter. All these shortkeys are usually available in textboxes only. Listed below are the available short keys:

Press	To
CTRL+S	Save
CTRL+X	Cut
CTRL+C	Copy
CTRL+V	Paste
CTRL+Z	Undo

Note: This is in synchronization with most major Windows applications.

7.5   Version Comparison

EventReporter comes in different versions. Some of them are more feature-richer then others. The manual covers description about the full feature set. In order to remove confusion we have created a Product Comparison Sheet which identifies the differences between different available versions. Click here to see that which Version provides which services, actions and other features.

8   Copyrights

This documentation as well as the actual EventReporter 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.

We acknowledge using these following third party tools. Here are the download links:

Openssl-0.9.7d: http://www.adiscon.org/3rdparty/openssl-0.9.7d.tar.gz
Ucd-snmp-4.2.3:   http://www.adiscon.org/3rdparty/ucd-snmp-4.2.3.tar.gz
Liblogging: http://www.adiscon.org/3rdparty/liblogging.zip
VB6 NeoCaption: http://www.adiscon.org/3rdparty/VB6_NeoCaption_Full_Source.zip

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.

9 Glossary of Terms

The Glossary of Terms is also available on the Web:

http://www.monitorware.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.

9.1 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.

Click here for more Information about EventReporter.

9.2 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.

Click here for more Information about Milliseconds.

9.3 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/en/logger/)
- ActiveLogger (www.activelogger.com)
- EventReporter (www.eventreporter.com)
- IISLogger (www.iislogger.com)
- MoniLog (www.monilog.com)
- MonitorWare Agent (www.mwagent.com)
- MonitorWare Console (www.mwconsole.com)
- WinSyslog (www.winsyslog.com)

There is also an open source syslog library available for programmers whishing to

integrate syslog into their C/C++ programs:

- Liblogging (www.liblogging.org)

New products are continously being added - please be sure to check www.monitorware.com from time to time for updates.

Click here for more Information about the MonitorWare Line of Products.

9.4 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.

Click here for more Information about the Resource ID:

9.5 SETP

SETP is the "Simple Event Transfer Protocol". SETP allows reliable delivery of events between SETP supporting systems. EventReporter, WinSyslog and MonitorWare Agent support SETP. EventReporter works as SETP Client Only. As such, it can forward events generated and gathered by them to central or intermediary SETP servers. WinSyslog EnterPrise Edition works as SETP client and server, only. 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.

Click here for more Information about SETP.

9.6   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.

Click here for more Information about SMTP.

9.7   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. Tradidionally, 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 distinguise 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.

Click here for more Information about Syslog Facility.

9.8 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.

Click here for more Information about TCP.

9.9 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.

Click here for more Information about UDP.

9.10 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.

Click here for more Information about Upgrade Insurance.

9.11 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 centrel 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.

Click here for More Information about UTC.