EventReporter 6.1

User Manual

By Adiscon

Contents

About EventReporter 6.1

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 smaller size 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 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 monitor additional system settings, you might want to have a look at the MonitorWare agent available at www.monitorware.com.

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.

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.

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 - under its previous name EvntSLog - 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' sites 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 log forwarding process "the engine" is implemented as a native multithreaded Windows NT service. It can be controlled via the control panel services applet.

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, German 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!

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 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).

System Requirements

EventReporter has minimal requirements.

The **EventReporter client** needs roughly 10 MB of disk space. The EventReporter client is optional and needs not to be present on a production system.

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 shouldn’t 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 your found this documentation in) 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 are no ALPHA executable for that operating systems.

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.

Installation

Installation is quick and easy. To facilitate mass rollouts, EventReporter has two setup modes:

- Full Install
- Engine-Only Install

The full install includes both the EventReporter client and service. In large environments, this is typically installed on a "master machine" being used to create the configuration parameters. The Engine-Only install includes the EventReporter service only. In large environments, that is the install process used primarily on a large number of target machines.

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 downloadable from the EventReporter web site.

EventReporter must be installed by a user with administrative permissions.

Full Install

The install set (the ZIP file you downloaded) contains a standard setup program and it's necessary helper files. Please unzip the archive to any directory you like. This can be a local drive, a removable one or a remote share on a file server. A Win32 Unzip program can be found at http://www.winzip.com.

After unzipping, simply double-click "setup.exe" and follow the onscreen instructions.

Setup.exe will install the EventReporter client and copy the Service process to disk.

Full Install mode is highly recommend for first time installations!

Engine-Only Install

There is no GUI setup program for an engine-only installation. The main purpose of this install mode is to roll out the product to a large number of machines.

Actual installation is straightforward

1. Copy EvntSLog.exe to any location you like (on the machines *local* hard drive)
2. Install it as a service by running "EvntSlog -i"
3. Use regedit to customize its settings – most importantly the syslog daemon’s IP address (a .reg file can be used for this purpose).

Important

Please be sure to copy EvntSLog.exe to a directory on a local drive. The install process (EvntSLog -i) will install the service to run from the current working directory. If that is not on the local drive, you need to have access privileges to the file server EvntSLog is stored on. The default service account - local system - does not have such privileges. Thus service startup will fail. If you need this setup, be sure to set the service account to someone with sufficient privileges (via control panel services applet).

Customization of the EventReporter service is via the registry. Modifications can be made directly via RegEdit (see documentation on how to do that) or via the EventReporter client (which must then be installed). Please note that the registry "Parameters" key ("HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AdisconEvntSLog\Parameters")can be exported to a .reg file and re-imported by calling regedit. As such, a mass rollout can be fully scripted by the following batch file:

```
copy \\server\share\eventslog.exe c:\some-local-dir
cd \some-local-dir
evntslog -i
regedit \\server\share\configParms.reg
```

Users of any prior NT service versions of EventReporter can uninstall the old version via "EvntSlog -u" before installing the new one. Please note that uninstalling the service will also remove all configuration settings. Uninstallation of old service version is not required. All EventReporter service versions are able to work on the configuration settings of previous ones.

Obtaining a Printable Manual

A printable version of the manual can be obtained at

http://www.monitorware.com/en/Manual/

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

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. For detailed instruction on the configuration of common scenarios, be sure to see “Step-by-Step Guides” on page 28.

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

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 a condition evaluates to true (“matches”), actions associated with this rule are carried out (for example, storing the information unit to disk or emailing an administrative alert.

Note: The screenshots in this tutorial are made with MonitorWare Agent, the user interface of which is similar to the one EventReporter 6.1 uses.

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 operations. Boolean operators like “AND” or “OR” can be used to create complex conditions.

If you are note so sure about the Boolean operations, 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 also true. In all other cases, it is false.

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

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

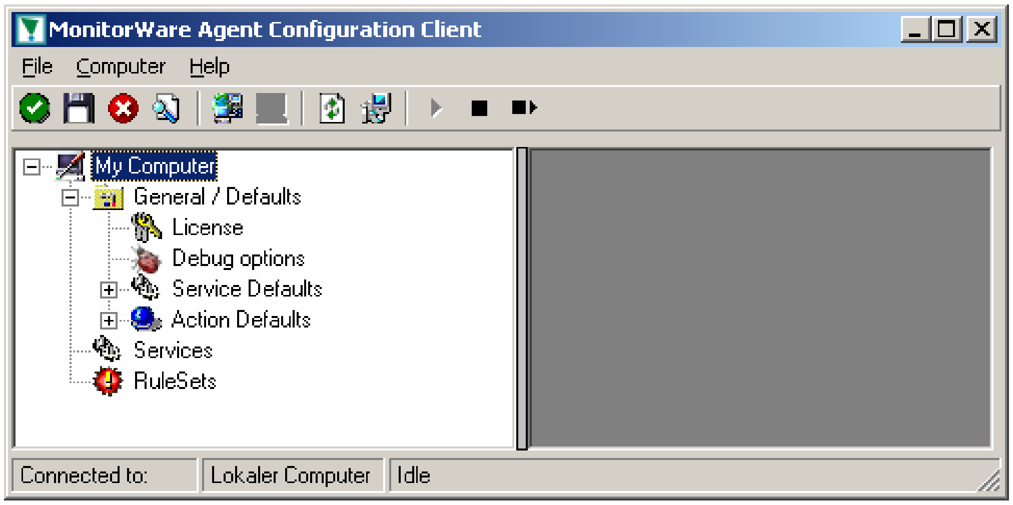
Ignoring Events

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

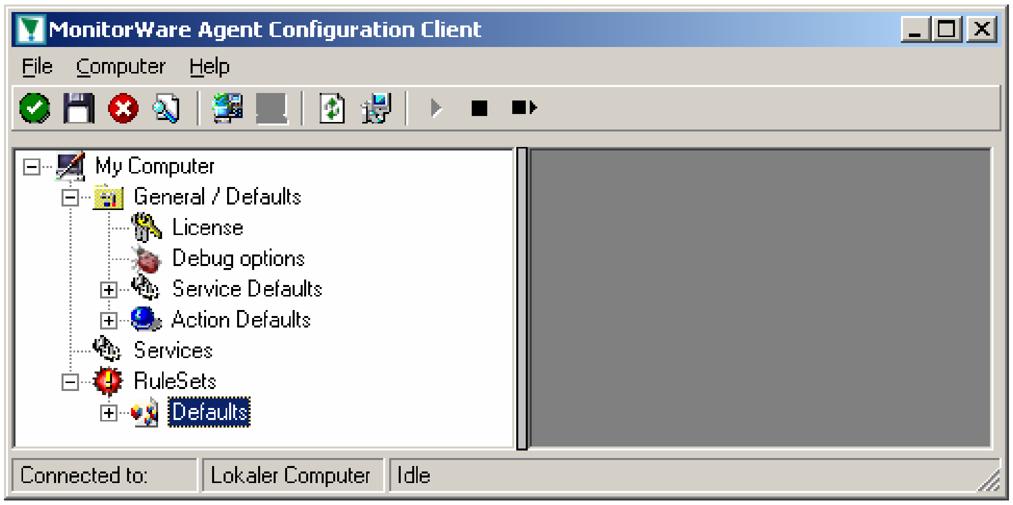
We handle these events on top of our rule set. This ensures that only minimal processing time is needed and they are discarded as soon as possible.

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

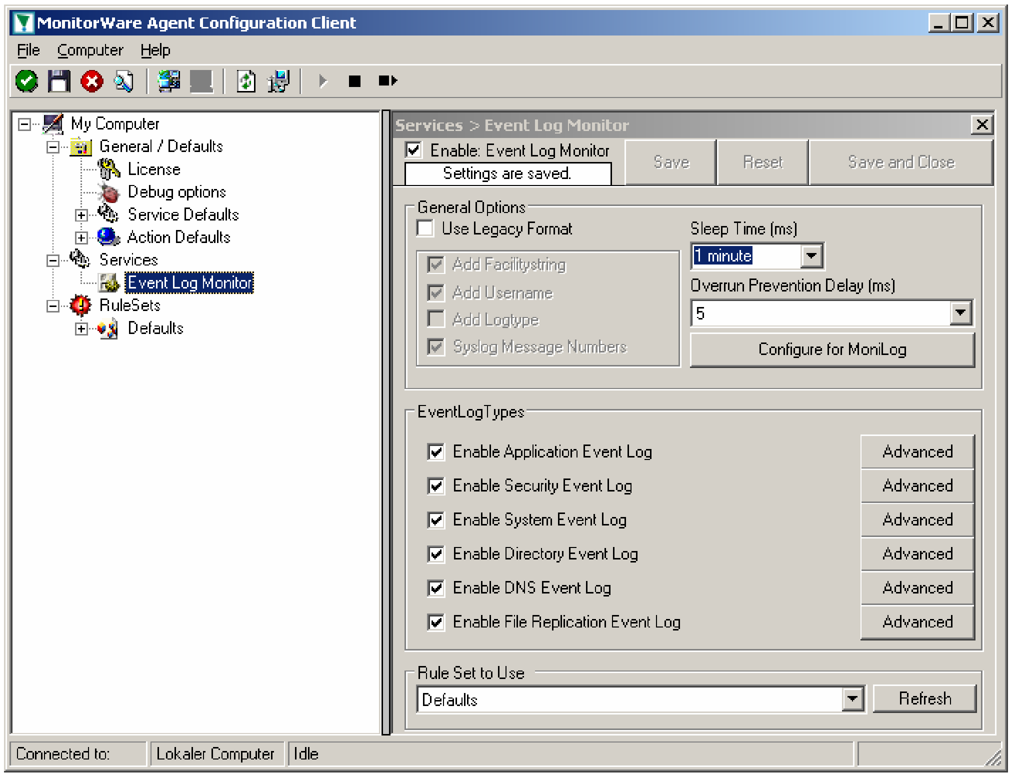
In this sample, no service or rule set is yet defined. It is just a “plain” system right after install, as can be seen in the following screen shot:



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



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


MonitorWare Agent Configuration Client

File Computer Help

- My Computer
  - General / Defaults
    - License
    - Debug options
    - Service Defaults
    - Action Defaults
  - Services
    - Event Log Monitor
  - RuleSets
    - Defaults

Services > Event Log Monitor

[✓] Enable: Event Log Monitor
Settings are saved.
Save | Reset | Save and Close

General Options
[ ] Use Legacy Format
[✓] Add Facilitystring
[✓] Add Username
[ ] Add Logtype
[✓] Syslog Message Numbers
Sleep Time (ms): 1 minute
Overrun Prevention Delay (ms): 5
Configure for MoniLog

EventLogTypes
[✓] Enable Application Event Log — Advanced
[✓] Enable Security Event Log — Advanced
[✓] Enable System Event Log — Advanced
[✓] Enable Directory Event Log — Advanced
[✓] Enable DNS Event Log — Advanced
[✓] Enable File Replication Event Log — Advanced

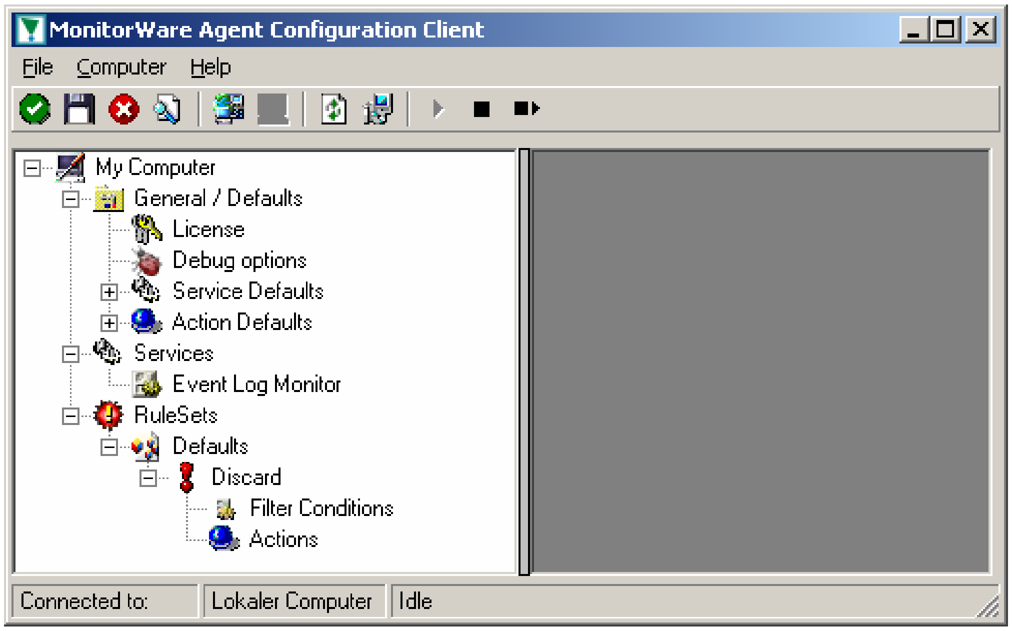
Rule Set to Use: Defaults — Refresh

Connected to: Lokaler Computer Idle

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

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

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

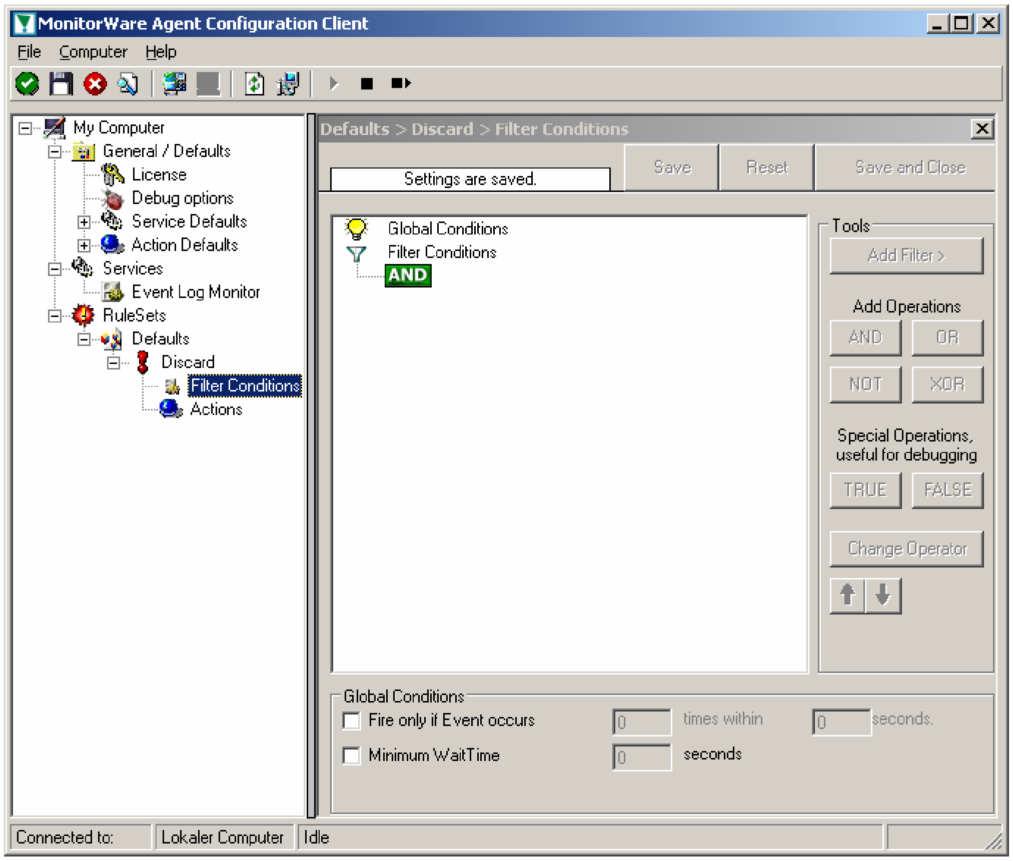

MonitorWare Agent Configuration Client

File Computer Help

- My Computer
  - General / Defaults
    - License
    - Debug options
    - Service Defaults
    - Action Defaults
  - Services
    - Event Log Monitor
  - RuleSets
    - Defaults
      - Discard
        - Filter Conditions
        - Actions

Connected to: Lokaler Computer Idle

Click on “Filter Conditions” to see this dialog:



Settings are saved.

Global Conditions

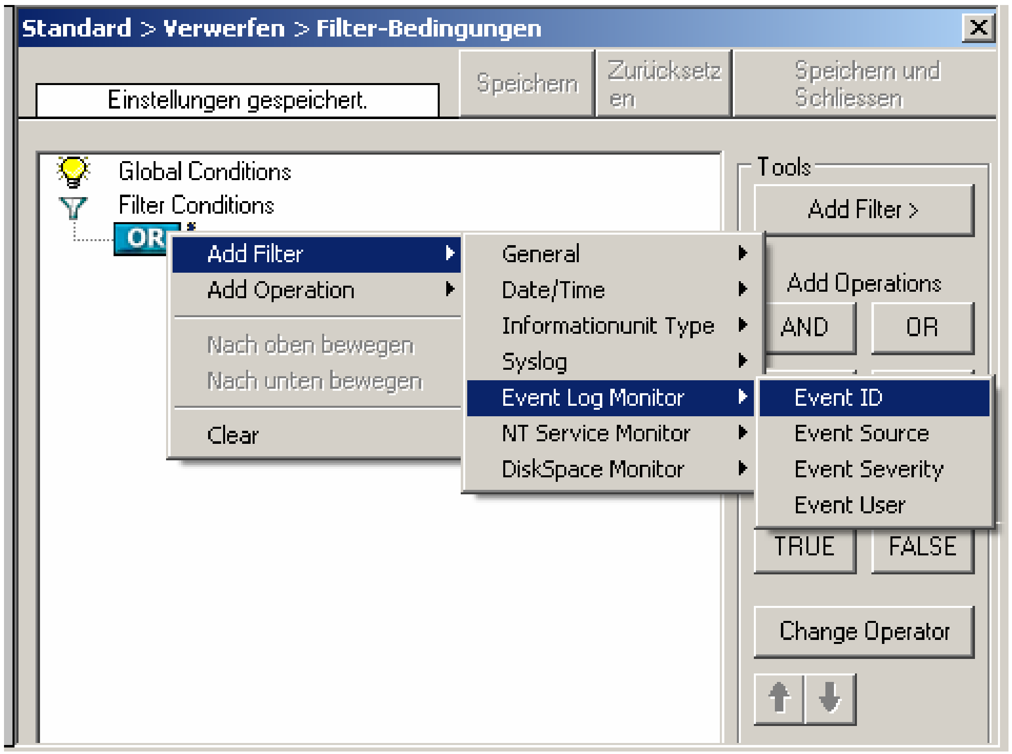
Fire only if Event occurs 0 times within 0 seconds.

Minimum WaitTime 0 seconds

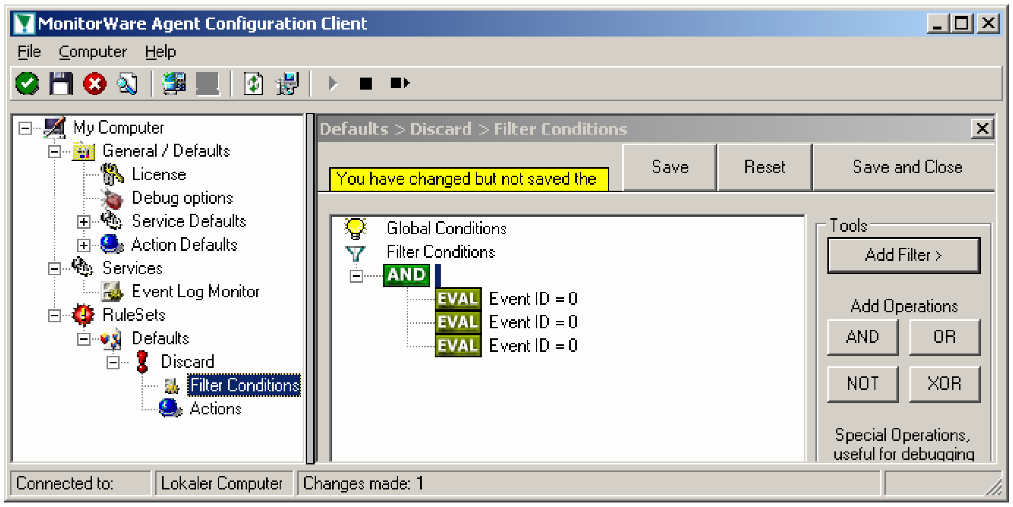
In that dialog, we will define our filter. Remember: we are about to filter those events, that we are **not** interested in. As we would like to discard multiple events, we need the Boolean “OR” operator in the top level node, not the default “AND”. Thus, we need to change the Boolean operator.

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 “AddFilter” from the pop up menu. This can be seen in the screen shot below:

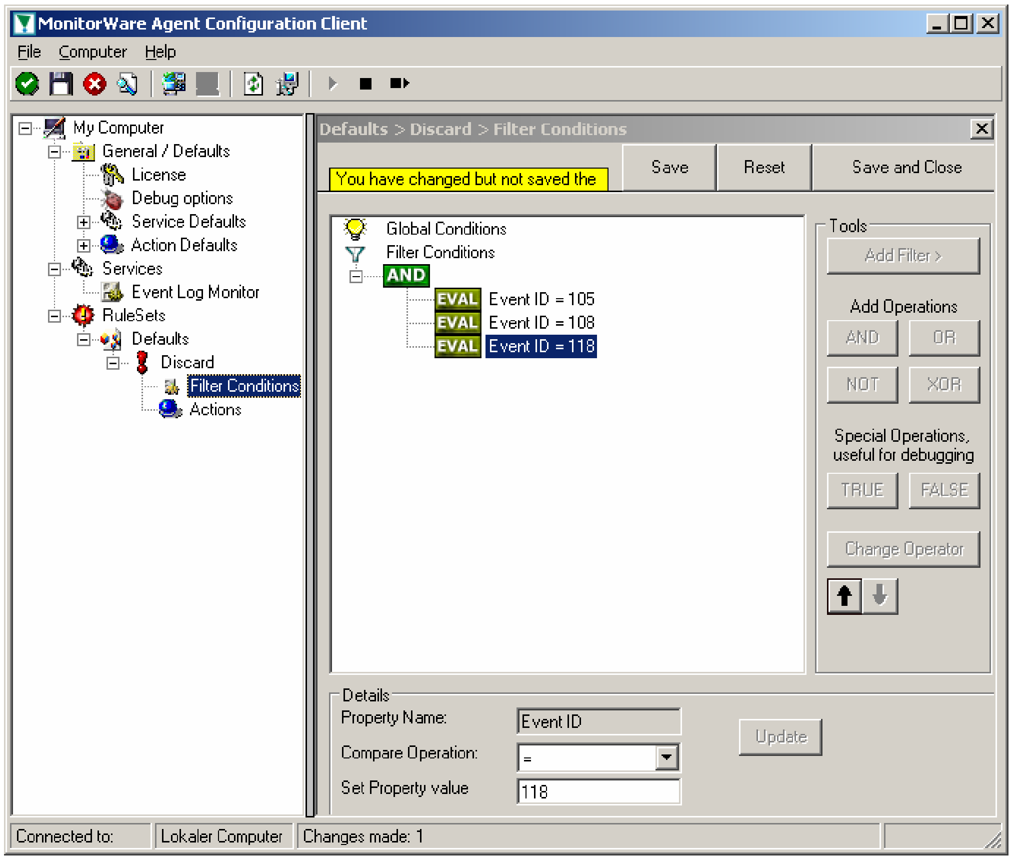


I prefer to add all three event id property filters first and later on change the event id to the actual value I am looking for. When you have added them, it should look as follows:



In order to enter the actual values, select each of the three filter. 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:



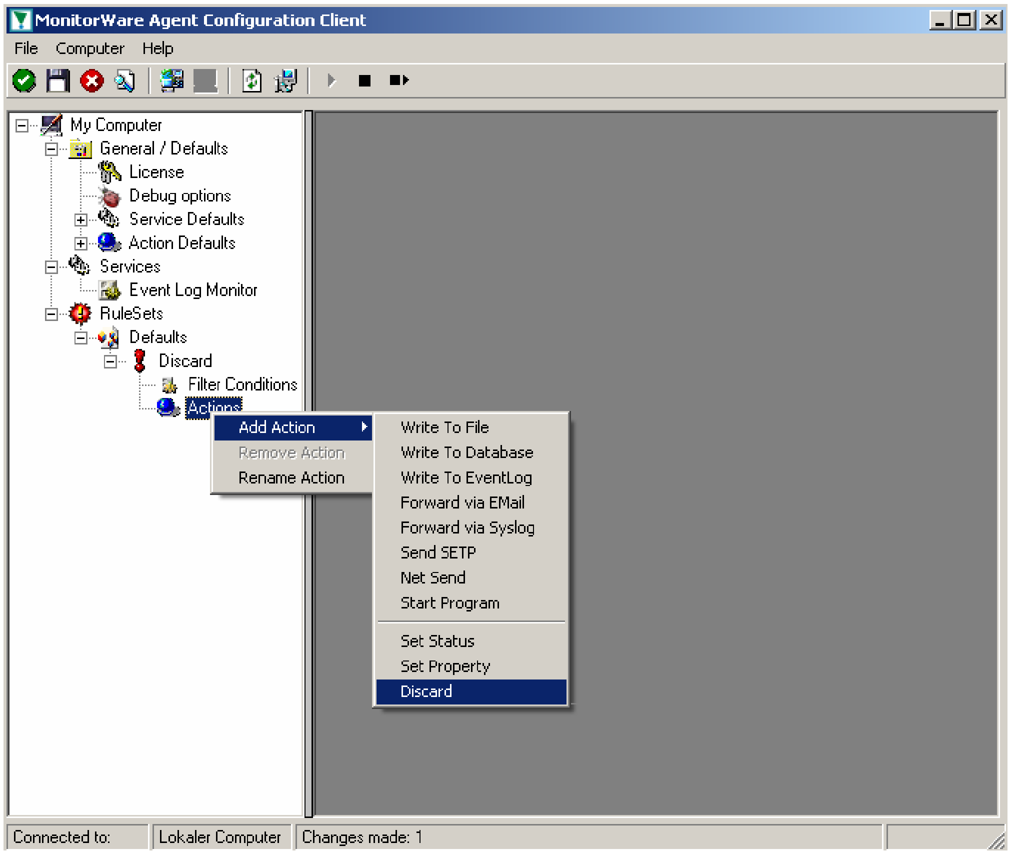
Global Conditions
Filter Conditions
AND
EVAL Event ID = 105
EVAL Event ID = 108
EVAL Event ID = 118

Details
Property Name: Event ID
Compare Operation: =
Set Property value 118

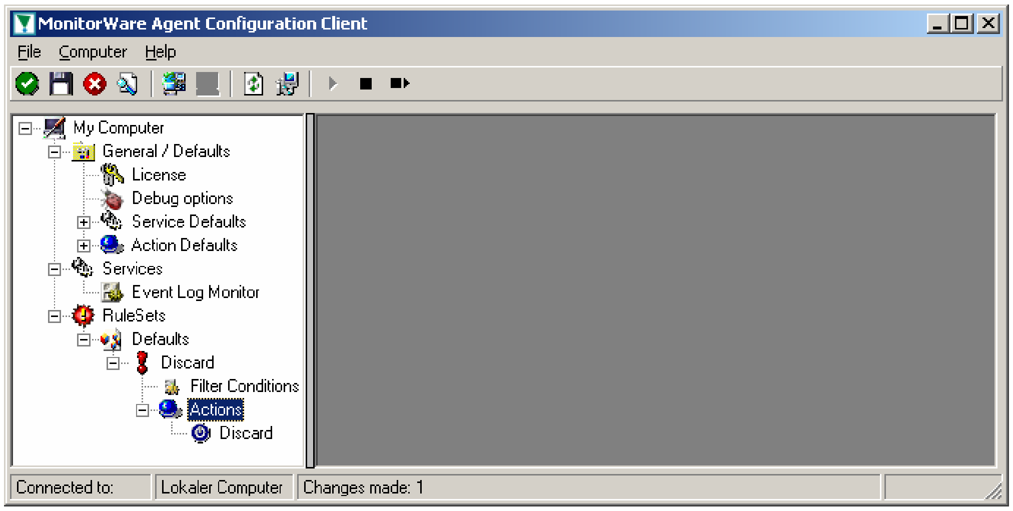
Connected to: Lokaler Computer Changes made: 1

Save the settings by clicking the (diskette-like) “Save” button. We have now selected all events that we would like to be discarded. In reality, these are often far more or a more complicated filter is needed. We have kept it simple so that the basic concept is easy to understand – but it can be as complex as your needs are.

Now let us go ahead and actually discard these events. This is done via an action. To do so, right-click on "Actions" and select "Discard."



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



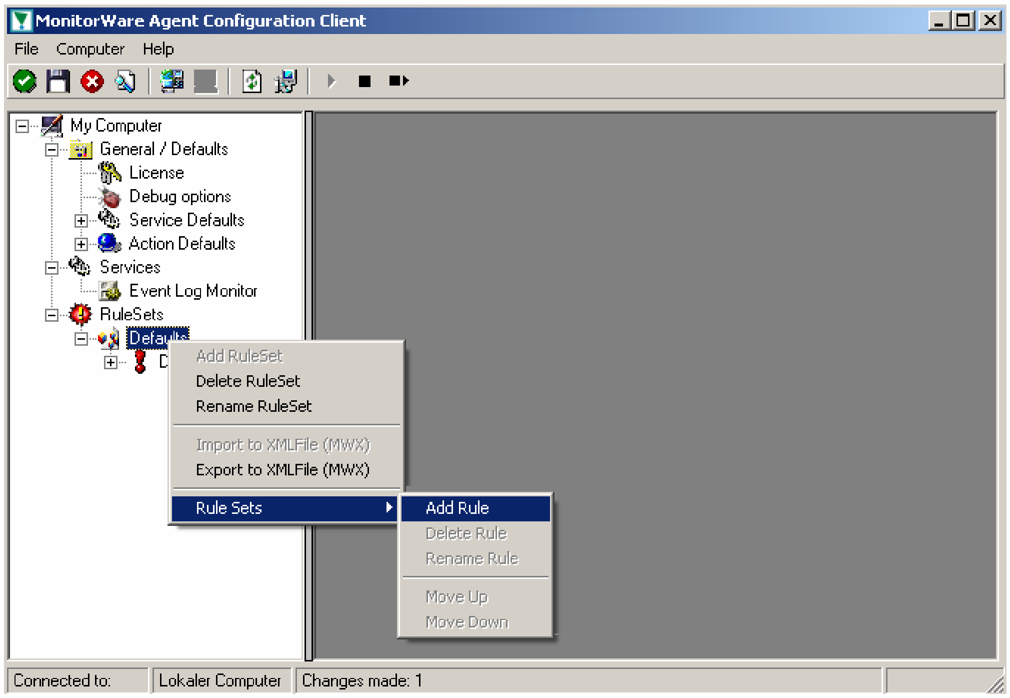
This concludes the definition of our first rule.

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.

Logging Events

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

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

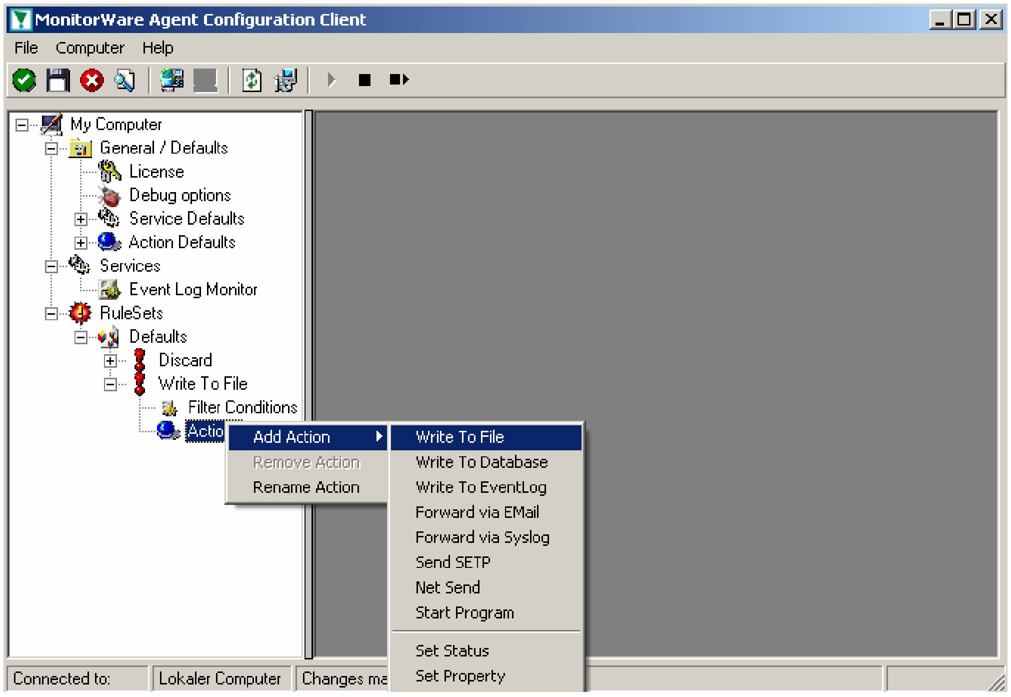


Use a name of you choosing. In our sample, we call this rule "Write To file".

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).

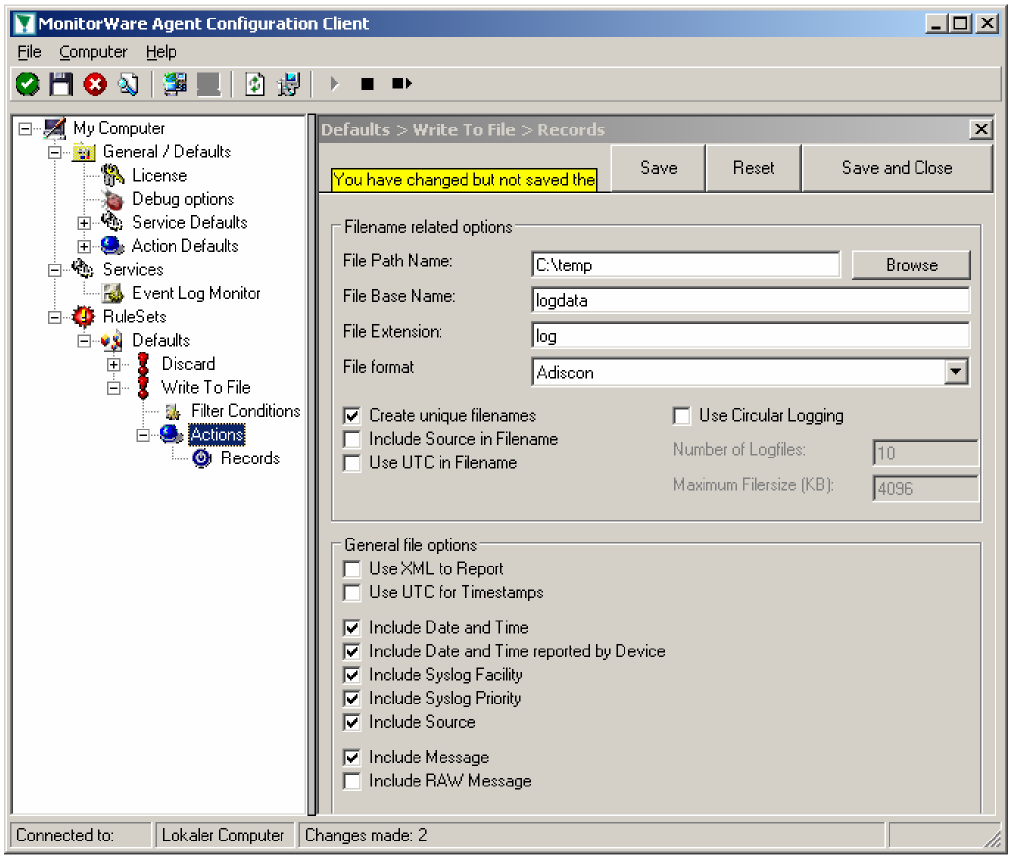
Since we want to store all still open Events with help of this rule, we don't require any filter rules here. However, a corresponding action must be defined. So we just need to define the action:

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



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

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



Filename related options

File Path Name: C:\temp

File Base Name: logdata

File Extension: log

File format: Adiscon

[✓] Create unique filenames
[ ] Include Source in Filename
[ ] Use UTC in Filename
[ ] Use Circular Logging
Number of Logfiles: 10
Maximum Filersize (KB): 4096

General file options

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

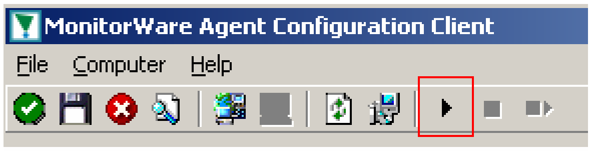
Important: make sure that the folder specified exists! If it does not exist, EventReporter will not write the log file. EventReporter will also **not** create the folder by itself. So if the folder does not exist, be sure to either create it or select a different (existing) one.

In our sample, we also change the file base name to “logdata”. This was just done out of personal preference. There is no need to do so, but it may be convenient for a number of reasons.

Summary

What did we do so far? All events from the Windows event log are passed through our rule engine and rule filters. Certain events are discarded and the remaining events are stored to a text file on the local disk (for later review or post-processing).

We can now do a quick test: Start EventReporter by hitting the start button seen below:



The log file should be created in the path you have specified. Open it with notepad. You should see many events originating from the event log. When you re-open the log file, new events should appear (if there were any new events in the Windows event log). The file is not easily readable. Most probably you have created it for archiving purposes or to run some external scripts against it.

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

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.

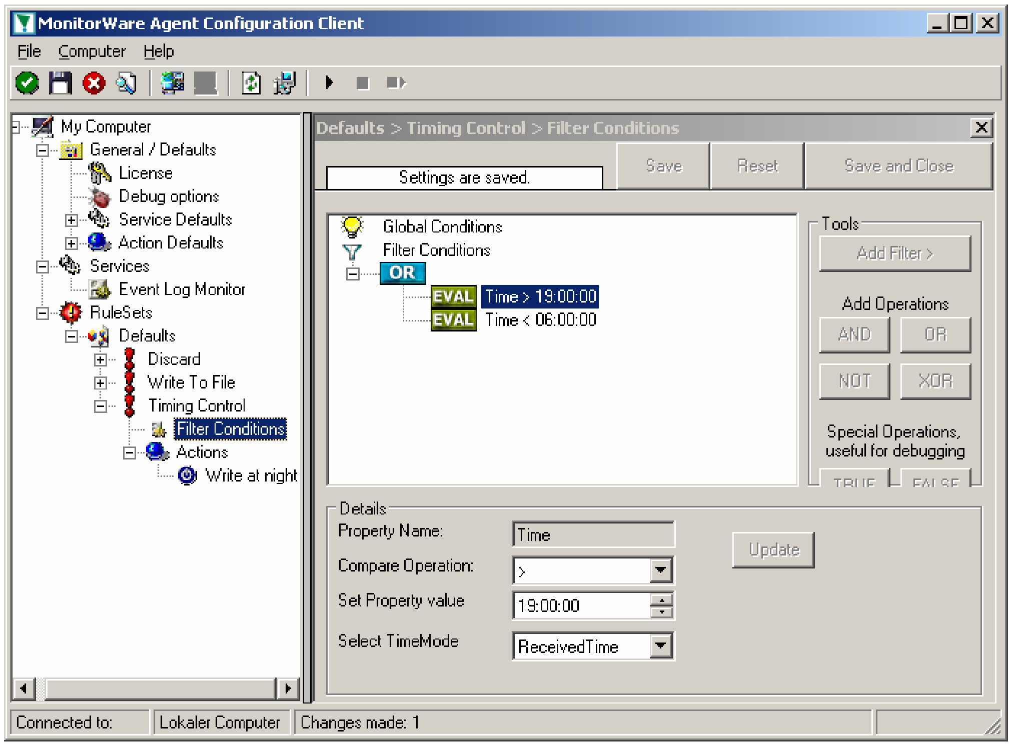
Similarly, there are a number of other good reasons why specific actions should only be applied during specific time frames. Fortunately, EventReporter allows to define complex time frames. In this tutorial, though, we focus on the simple ones.

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 realty, time based filters are often combined with other conditions to trigger time based alerts. But this would complicate things to 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.

Please note: we use the 24 hour clock system below. As this manual is read by a world-wide community, this provides easier understanding. Our apologies to those using 12 hour clock systems (as a quick reminder, 1a is 01:00 while 1p is 13:00 – the hours are just counted forward until 24:00 which is midnight).



Settings are saved.

Time > 19:00:00
Time < 06:00:00

Property Name: Time
Compare Operation: >
Set Property value 19:00:00
Select TimeMode ReceivedTime

Connected to: Lokaler Computer Changes made: 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 nighttimes between 19:00:01 (7p) and 5:59:50 (5:59a), 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 8:00 in the morning, the action will not be called – it is outside of the specified time frame:

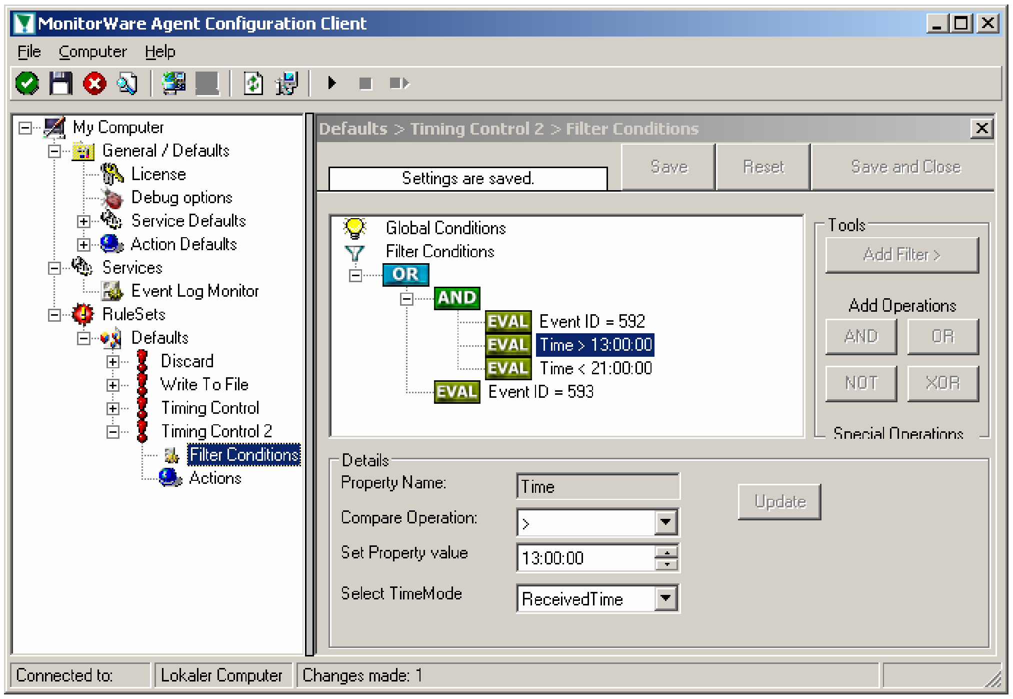
08:00:00 > 19:00:00 = *false*

08:00:00 < 06:00:00 = *false*

If the very same event comes in at 8:00 in the evening (20:00 hours in the 24 hour clock system), the filter condition evaluates to true and the action will be executed.

20:00:00 > 19:00:00 = *true*
20:00:00 < 06:00:00 = *false*

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


MonitorWare Agent Configuration Client

File Computer Help

My Computer
General / Defaults
License
Debug options
Service Defaults
Action Defaults
Services
Event Log Monitor
RuleSets
Defaults
Discard
Write To File
Timing Control
Timing Control 2
Filter Conditions
Actions

Defaults > Timing Control 2 > Filter Conditions

Settings are saved.

Save | Reset | Save and Close

Global Conditions
Filter Conditions
OR
AND
EVAL Event ID = 592
EVAL Time > 13:00:00
EVAL Time < 21:00:00
EVAL Event ID = 593

Tools
Add Filter >
Add Operations
AND | OR
NOT | XOR
Special Operations

Details
Property Name: Time
Compare Operation: >
Set Property value: 13:00:00
Select TimeMode: ReceivedTime
Update

Connected to: | Lokaler Computer | Changes made: 1

In this example, we will call the configured actions if events with ID 592 occur between 13:00:01 (1p) and 20:59:59 (roughly 9p). 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 7:00a sharp:

Event ID = 592	= *true*
07:00:00 > 13:00:00	= *false*
07:00:00 < 21:00:00	= *false*
“AND” Branch	= *false*
Event ID = 593	= *false*

In all, the filter condition is false.

Now, the same event comes in at 14:00 (2p):

Program start ID = 592 = *true*

Event ID = 592	= *true*
14:00:00 > 13:00:00	= *true*
14:00:00 < 21:00:00	= *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 her/it 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 7:00 in the morning:

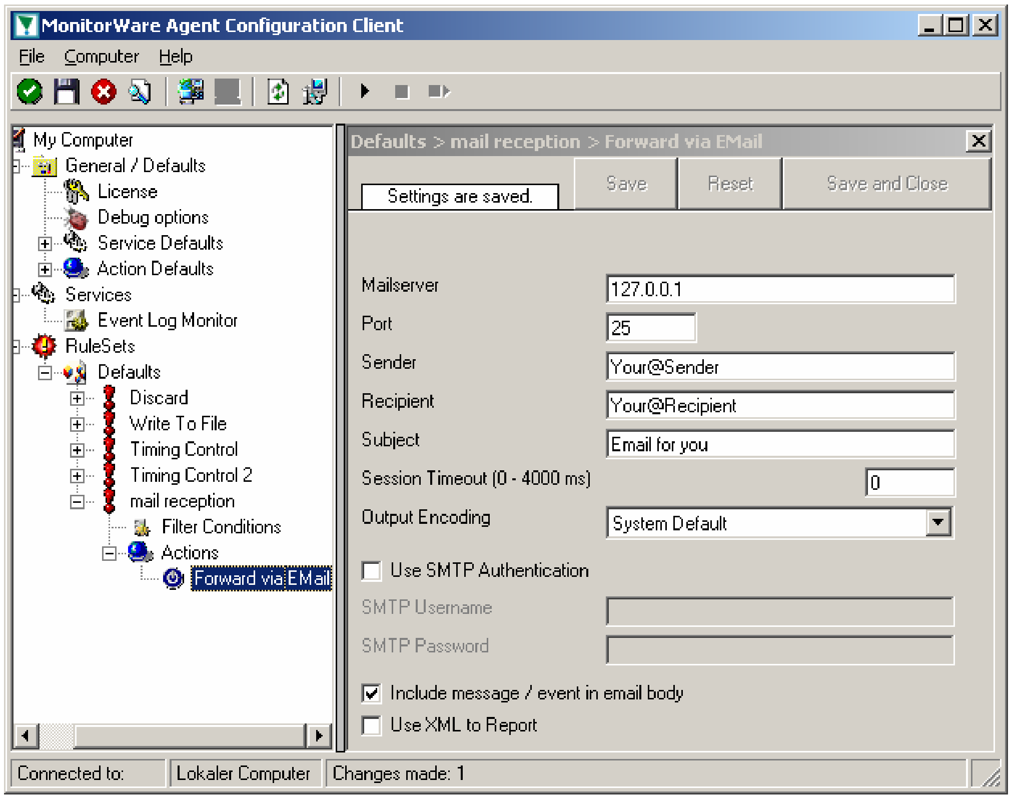
Program start ID = 592	= *true*
Event ID = 592	= *false*
07:00:00 > 13:00:00	= *false*
07:00:00 < 21:00:00	= *false*
"AND" Branch	= *false*
Event ID = 593	= *true*

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).

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 "mail 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.


MonitorWare Agent Configuration Client

Defaults > mail reception > Forward via EMail

Settings are saved.

Mailserver	127.0.0.1
Port	25
Sender	Your@Sender
Recipient	Your@Recipient
Subject	Email for you
Session Timeout (0 - 4000 ms)	0
Output Encoding	System Default

☐ Use SMTP Authentication

SMTP Username

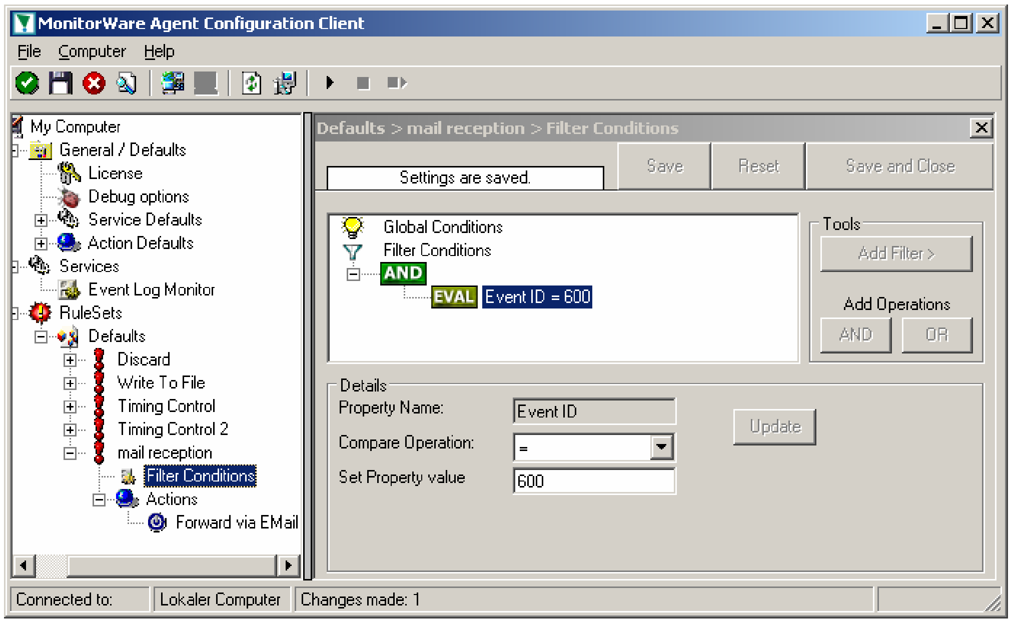
SMTP Password

☑ Include message / event in email body

☐ Use XML to Report

Connected to: Lokaler Computer Changes made: 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:


MonitorWare Agent Configuration Client

Defaults > mail reception > Filter Conditions

Settings are saved.

Global Conditions
Filter Conditions
AND
EVAL Event ID = 600

Property Name:	Event ID
Compare Operation:	=
Set Property value	600

Connected to: Lokaler Computer Changes made: 1

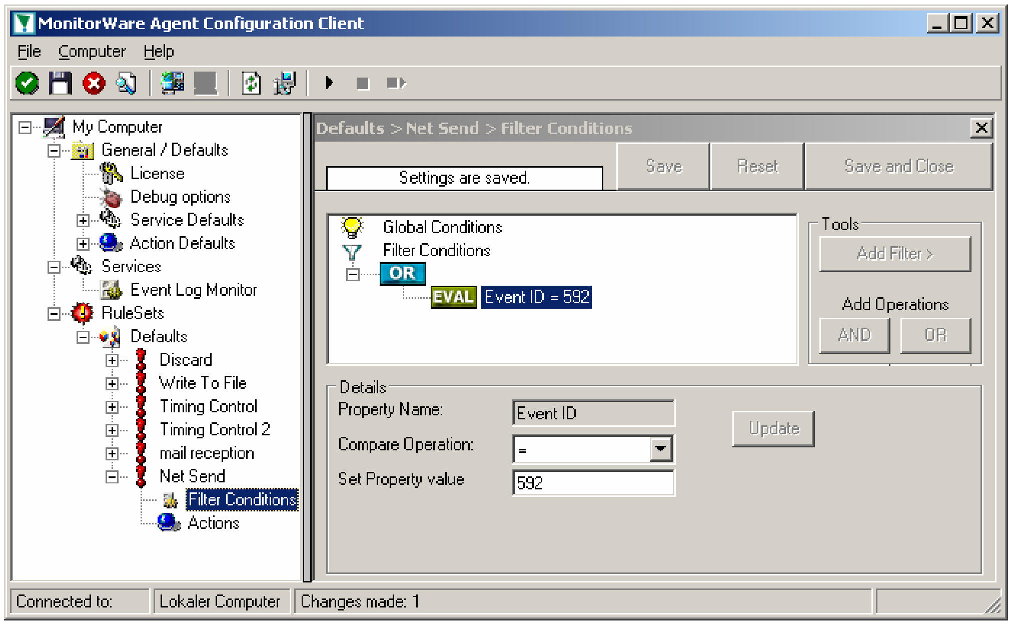
When you have finished this steps, be sure to save the configuration and re-start the EventReporter service. After the restart, the newly extended rule set will be executed. In addition the rules defined so far, the new one will be carried out, emailing all events with ID 600 to the specified recipient.

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).



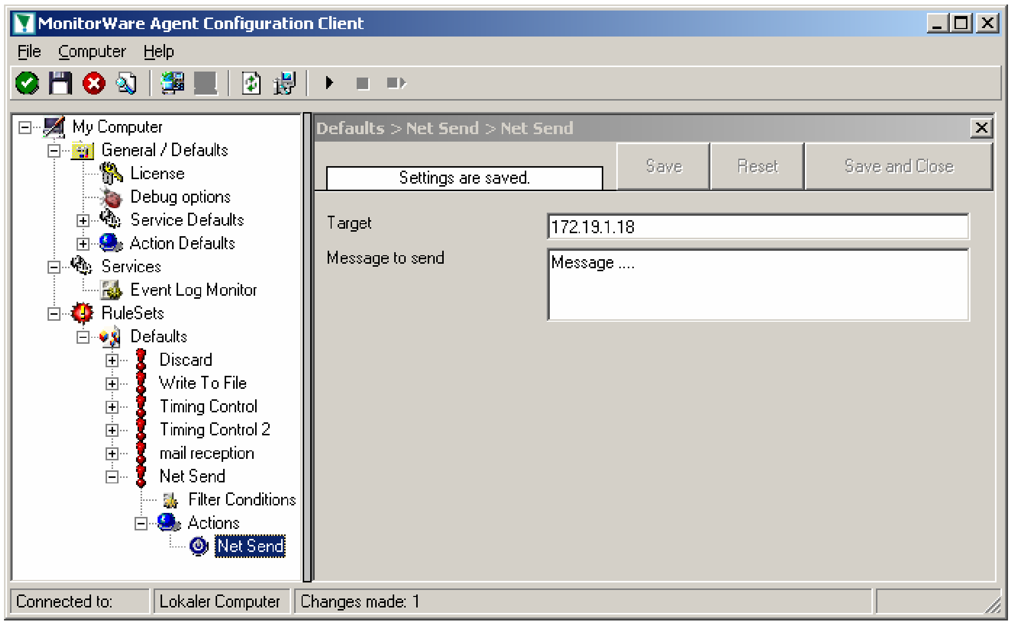
Settings are saved.

Global Conditions
Filter Conditions
OR
EVAL Event ID = 592

Details
Property Name: Event ID
Compare Operation: =
Set Property value 592

Connected to: Lokaler Computer | Changes made: 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 send to. The message text itself goes into “Message to send”.



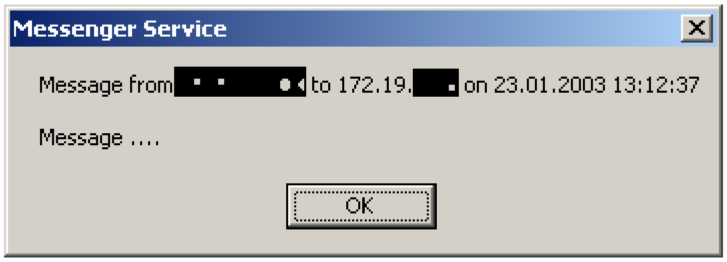
Settings are saved.

Target: 172.19.1.18

Message to send: Message ....

Connected to: Lokaler Computer | Changes made: 1

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):



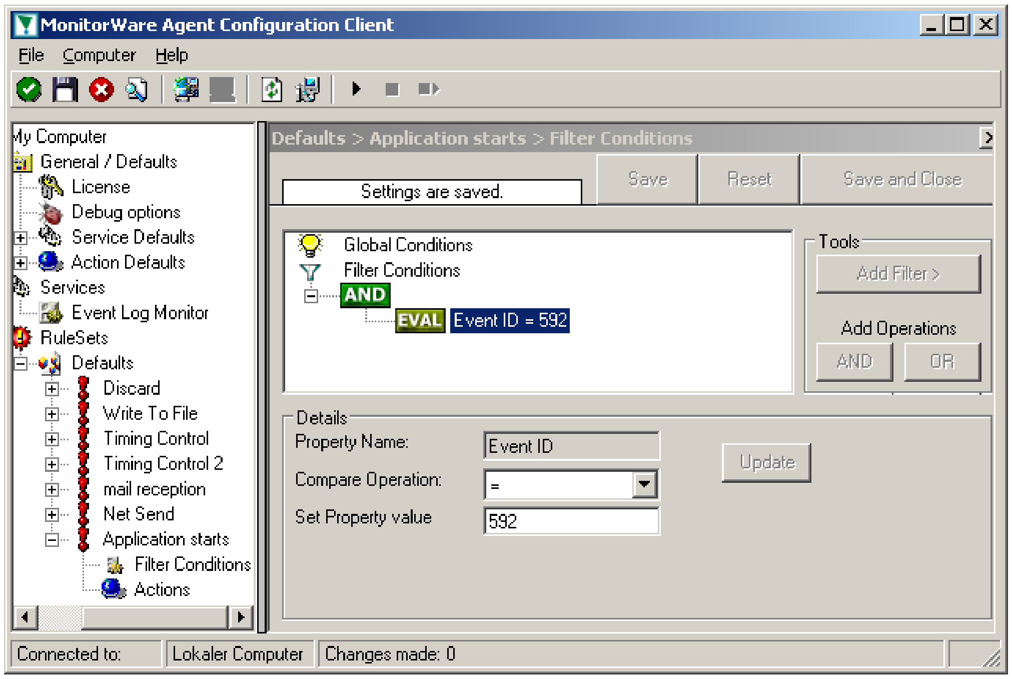
Message from [redacted] to 172.19.[redacted] on 23.01.2003 13:12:37

Message ....

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.

Or 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. So the application will start whenever event 592 comes in.

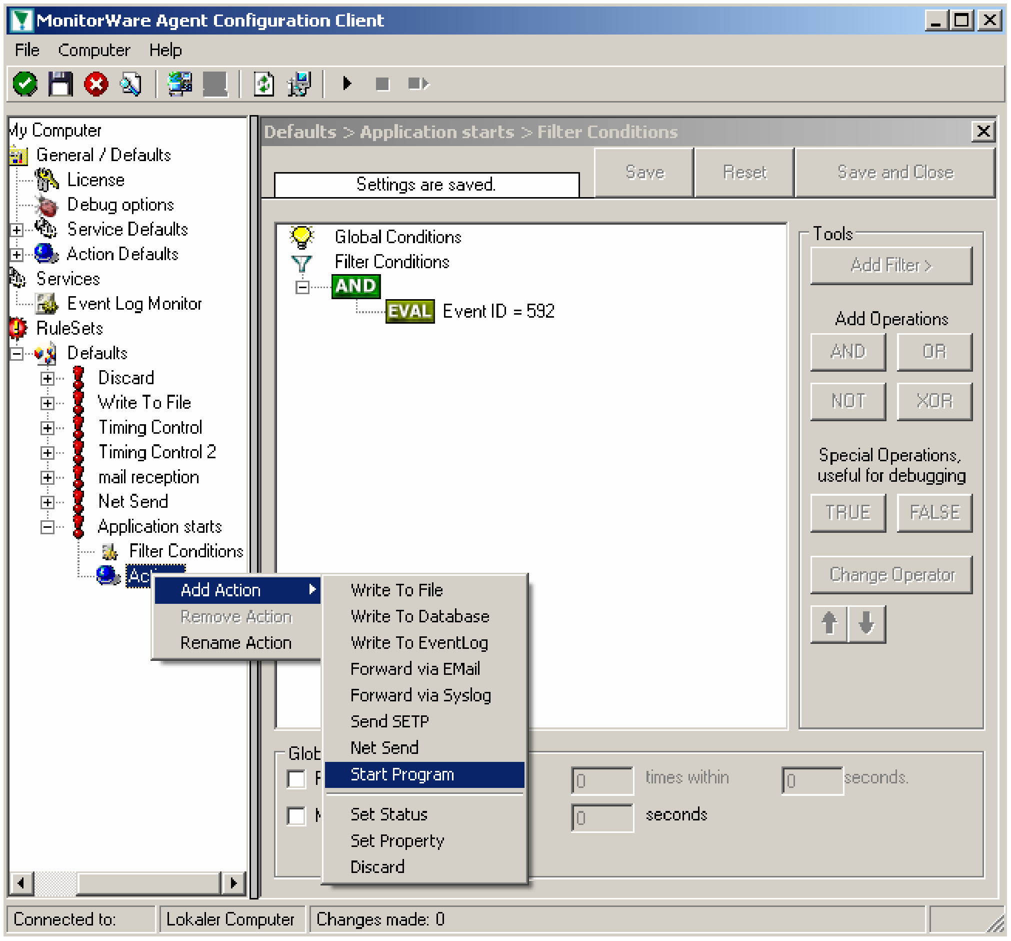


Settings are saved.

Global Conditions
Filter Conditions
AND
EVAL Event ID = 592

Details
Property Name: Event ID
Compare Operation: =
Set Property value 592

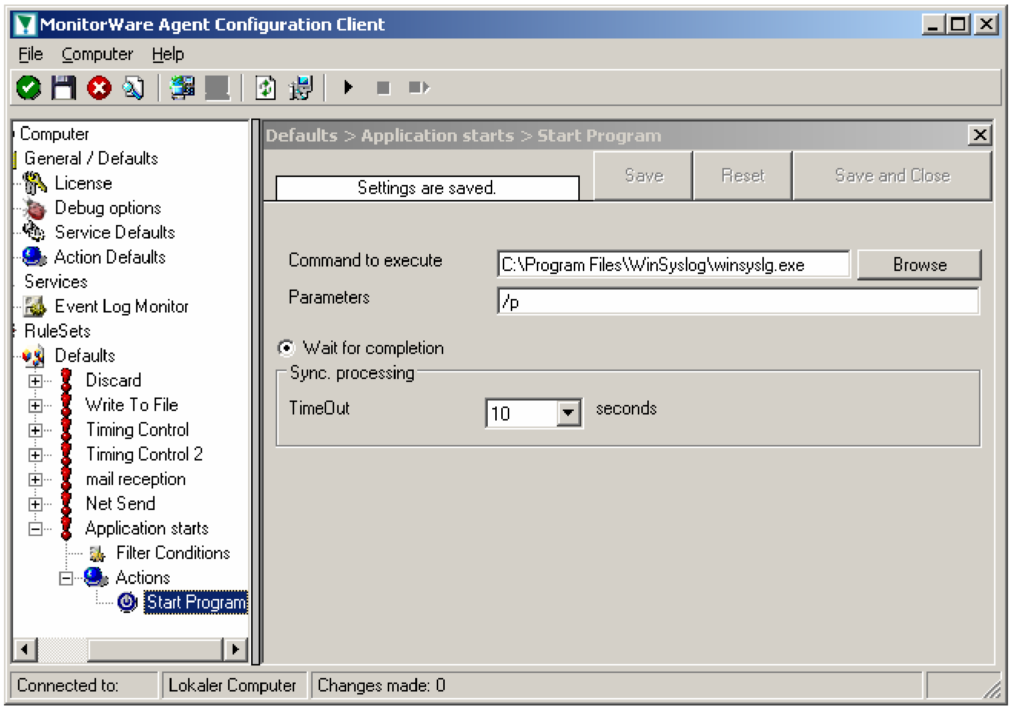
The start program action is just a "normal" action:



Settings are saved.

Global Conditions
Filter Conditions
AND
EVAL Event ID = 592

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



Defaults > Application starts > Start Program

Settings are saved.

Command to execute: C:\Program Files\WinSyslog\winsyslg.exe

Parameters: /p

[✓ Wait for completion]

Sync. processing

TimeOut: 10 seconds

Connected to: Lokaler Computer | Changes made: 0

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

Common Uses

EventReporter can be used in a multitude of ways to perform well in many different environments serving many different needs. This chapter describes some typical use cases. It includes pointers what to set up and also what can be achieved. This chapter provides an overview of the scenario. Detailed setup and configuration instructions can be found in the “Step-by-Step Guides” on page 28.

This chapter is organized among the four main use cases, which are

- Analysis
- Event Archival
- Alerting
- Solving Problems

Besides this main benefits, there are also some other scenarios, like relaying event data. They are also described.

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

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 not include all information that might be relevant to the situation. For details on the configuration properties, please see "Configuring " on page 29.

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. Information on installing can be found in "**Fehler! Verweisquelle konnte nicht gefunden werden.**" on page **Fehler! Textmarke nicht definiert.**.

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

Forwarding NT event logs to a syslog server

Forwarding NT event logs to an SETP server

Creating a rule set for database logging

Centralized event reports with Monilog

Intrusion detection via the Windows event log

Firewall setup for MonitorWare Agent

Configuring Windows for the Event Log Monitor

Creating a hardened log host

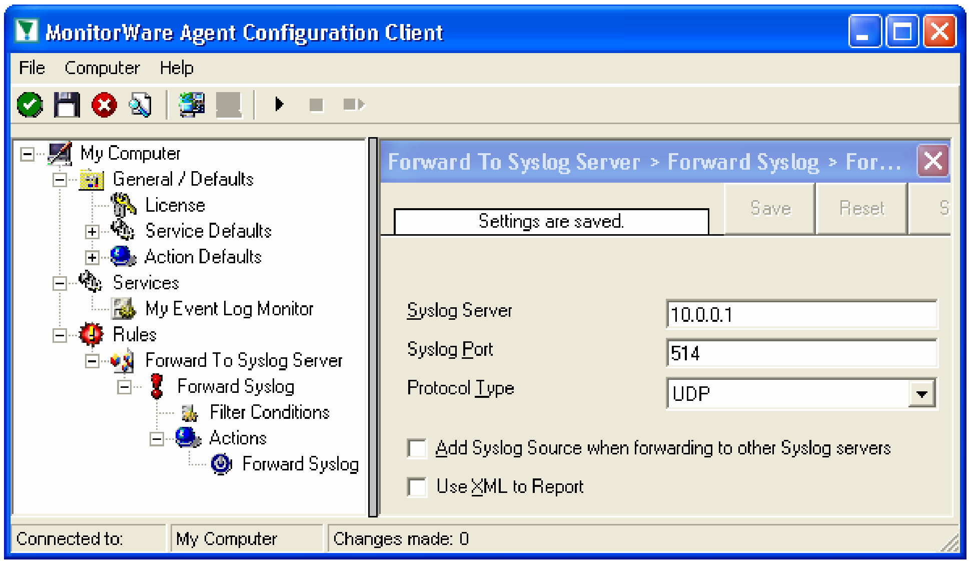
Configuring EventReporter

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:



Settings are saved.

Syslog Server: 10.0.0.1

Syslog Port: 514

Protocol Type: UDP

☐ Add Syslog Source when forwarding to other Syslog servers

☐ Use XML to Report

Connected to: My Computer | Changes made: 0

EventReporter Configuration Client

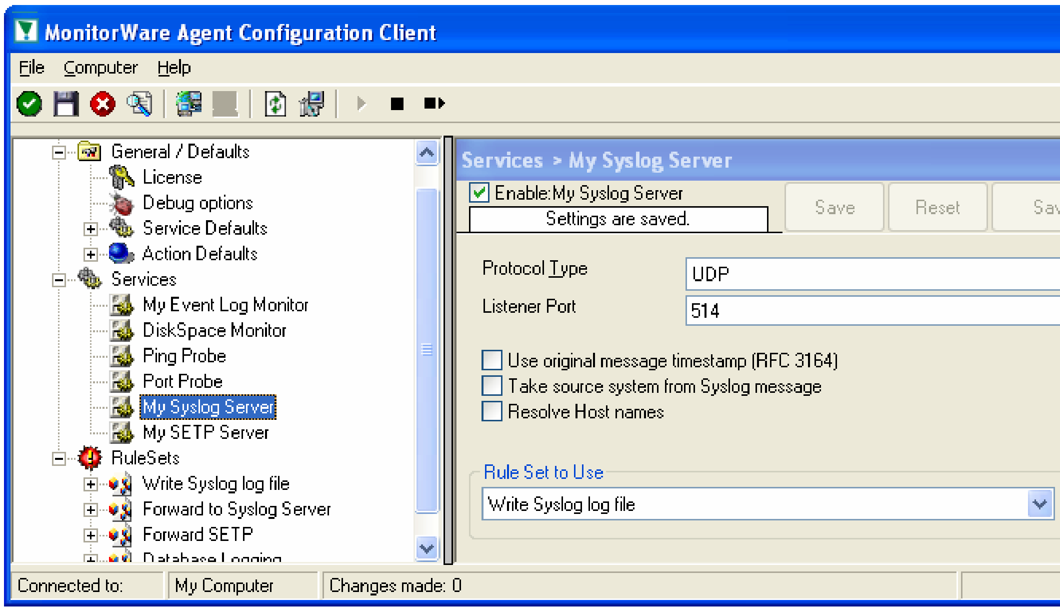
The configuration client ("the client") has two elements. On the left hand side is a tree view that allows you to select the various elements of the EventReporter system. On the right hand side are parameters specific to the element selected in the tree view. In the sample above, the right hand side 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 Agent that limits from doing so.

The service definition looks like this:



Enable:My Syslog Server ✓
Settings are saved.

Protocol Type: UDP
Listener Port: 514

- [ ] Use original message timestamp (RFC 3164)
- [ ] Take source system from Syslog message
- [ ] Resolve Host names

Rule Set to Use: Write Syslog log file

Connected to: My Computer | Changes made: 0

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).

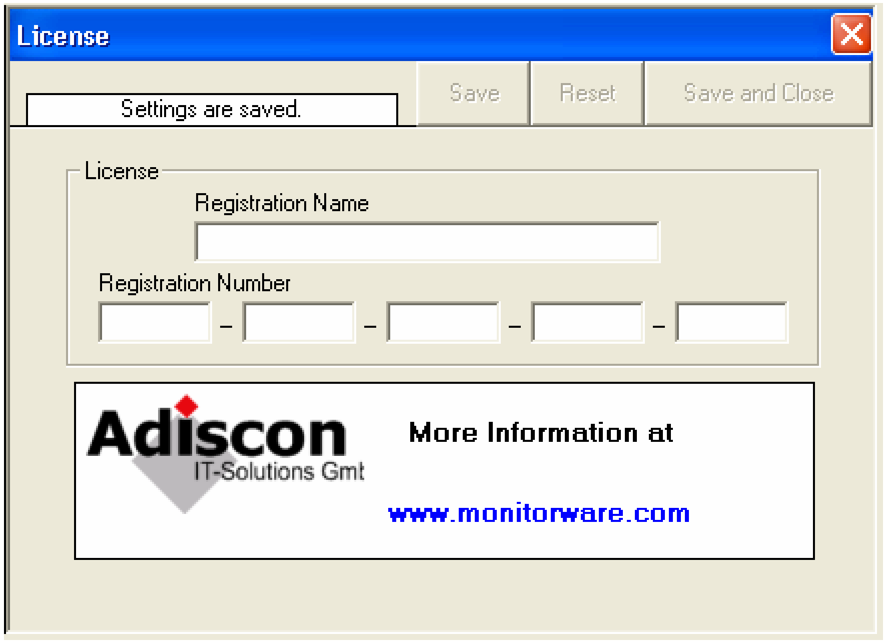
Beneath each rule set are the individual rules. As described in “**Fehler! Verweisquelle konnte nicht gefunden werden.**” on page **Fehler! Textmarke nicht definiert.**, a rule’s position in the list is vitally important. Rules at the top of the rule set are executed before those further down. To move 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.

License Options

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



License Option Parameters

Registration Name

The registration name is chosen by the user. It should correspond to your organization name, e.g. a company called "AA Carpenters, Inc." should not choose "AA" as registration name. This can easily be mistaken and most probably will be rejected by Adiscon for that reason. With the above scenario, we recommend using the full company name "AA Carpenters, Inc.".

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

Registration Number

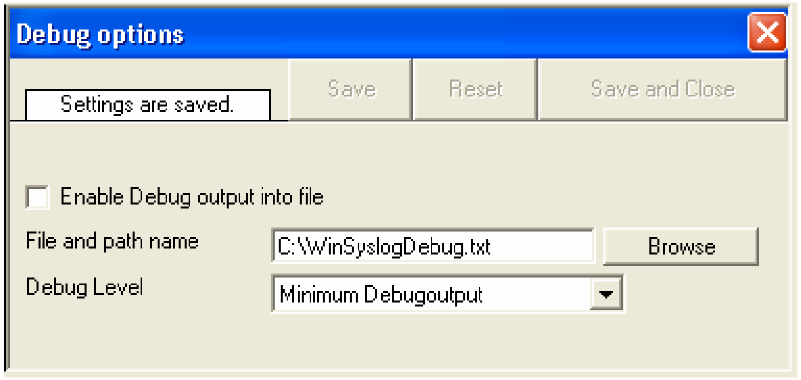
Adiscon provides this number. It is valid for a specific registration name. Be sure to enter the correct registration number. The client will detect invalid registration numbers and report and corresponding error.

Debug 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 this 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. But even the lowest level considerable slows down the service. As such, **we highly recommend turning debug logging off for normal operations.**



Settings are saved.

☐ Enable Debug output into file

File and path name: C:\WinSyslogDebug.txt

Debug Level: Minimum Debugoutput

Debug Options Parameters

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 file 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

Debug Level

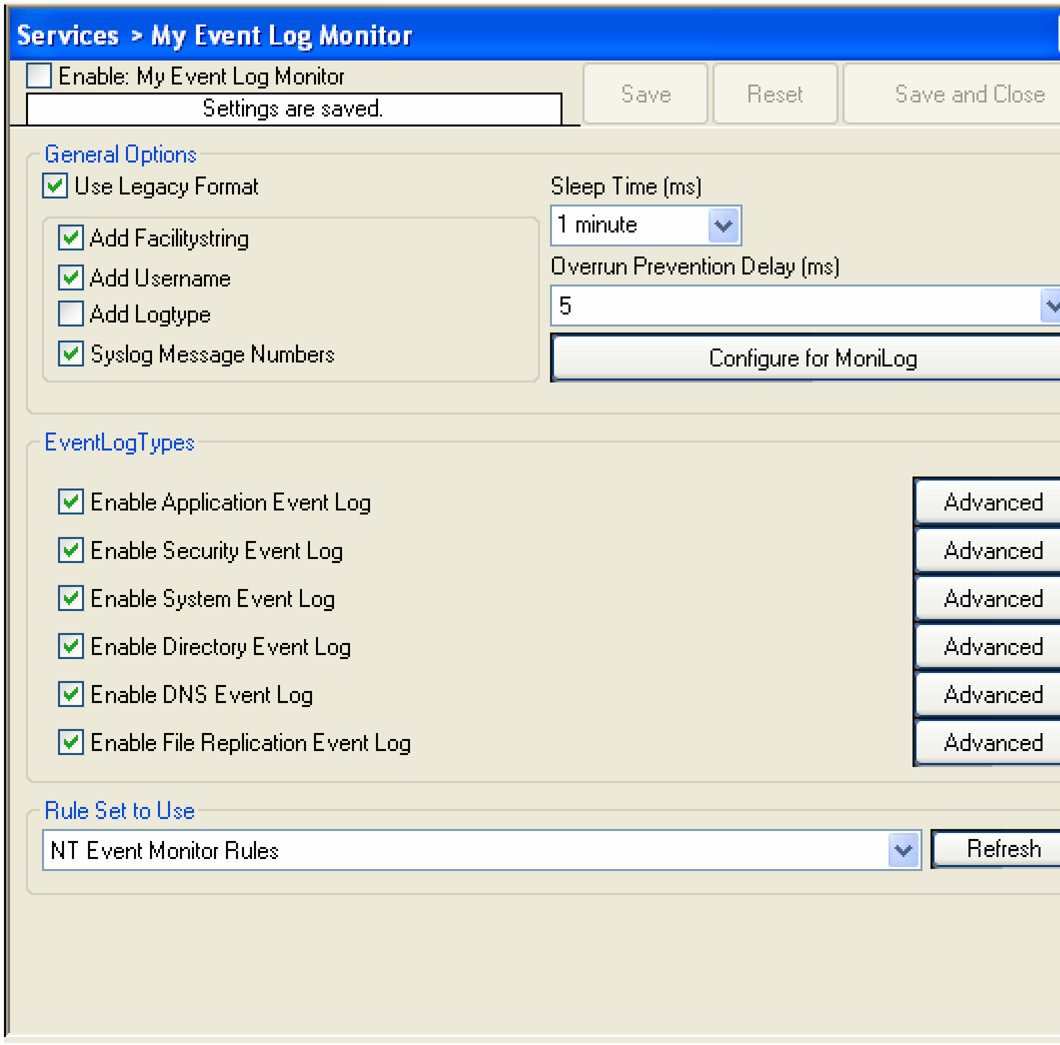
This controls the amount of debug information being written. We highly recommend only selecting "Minimum Debugoutput" unless otherwise instructed by Adiscon support.

Services

The EventReporter service gathers information from Windows event logs.

Event Log Monitor

This dialog configures event log monitor services. These services offer capabilities like Adiscon's EventReporter product. To allow previous EventReporter customers seamless upgrades to the EventReporter, it has a number of compatibility settings to support older message formats.


Services > My Event Log Monitor

☐ Enable: My Event Log Monitor

Settings are saved.

General Options

☑ Use Legacy Format

☑ Add Facilitystring

☑ Add Username

☐ Add Logtype

☑ Syslog Message Numbers

Sleep Time (ms): 1 minute

Overrun Prevention Delay (ms): 5

EventLogTypes

☑ Enable Application Event Log

☑ Enable Security Event Log

☑ Enable System Event Log

☑ Enable Directory Event Log

☑ Enable DNS Event Log

☑ Enable File Replication Event Log

Rule Set to Use: NT Event Monitor Rules

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).

Add Facility String

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

However, pre-version 3.2 EventReporter customers might want to turn it off to preserve compatibility with their existing parsing scripts. These versions did not support the "facility string.

This setting does only apply if the “Use Legacy Format” option is checked. Otherwise, it does not have any meaning and consequently cannot be configured in that case.

Syslog Message Numbers

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

This setting does only apply 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 scripts to parse EventReporter output. This option must also be unchecked if MoniLog is being used.

This setting does only apply if the “Use Legacy Format” option is checked. Otherwise, it does not have any meaning and consequently cannot be configured in that case.

Default Ruleset Name

Name of the rule set to be used for syslog server services. The rule set name must be valid.

Sleep Time

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

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

Very security-aware environments might use a shorter interval. The EventReporter 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

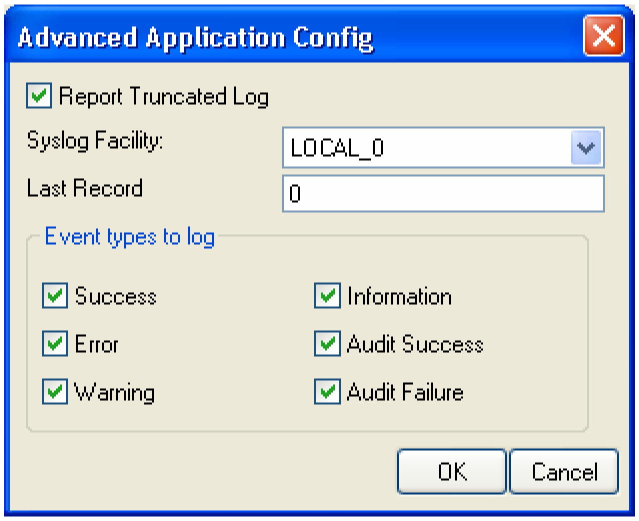
This property allows configuring a delay after generating an event. The time is the delay in milliseconds.

If run at a value of zero, the EventReporter 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 EventReporter 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, EventReporter 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.

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:


Advanced Application Config

[✓] Report Truncated Log

Syslog Facility: LOCAL_0

Last Record: 0

Event types to log

[✓] Success
[✓] Information
[✓] Error
[✓] Audit Success
[✓] Warning
[✓] Audit Failure

Report Truncated Log

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

The agent 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.

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

NT event log records are numbered serially, starting at one. The agent 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 NT 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 NT event type. There is a checkbox corresponding to each NT event type. Only checked event types will be processed. Unchecked ones will be ignored.

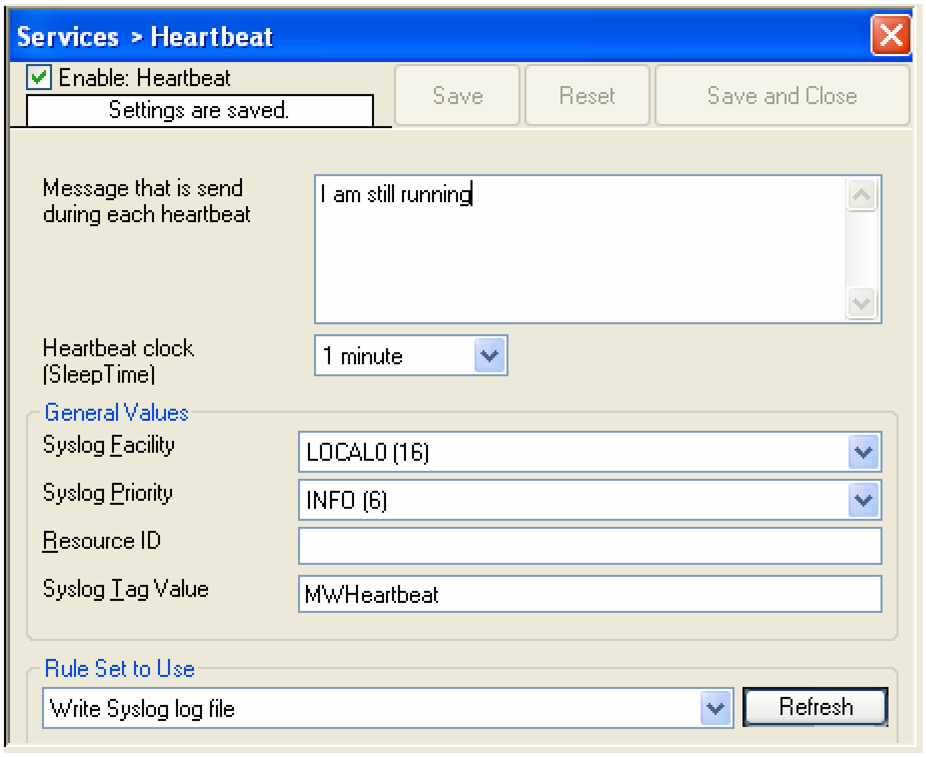
Filtering out unnecessary log types at this level enhances system performance because no information units will be generated and passed to the rule engine. Thus, Adiscon strongly recommends dropping unnecessary log types.

Important notice to pre version 6 EventReporter Customers

EventReporter had advanced filtering options. These options could be used to filter event log records based on their type, source and other settings. This functionality has been superseded by the rule engine. Consequently, it is no longer available at the event log monitor level.

Heartbeat

The heartbeat process can be used to continuously check if the EventReporter is running. 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 Agent is either in trouble or already stopped running.



Enable: Heartbeat ✓
Settings are saved.

Message that is send during each heartbeat: I am still running

Heartbeat clock (SleepTime): 1 minute

General Values

Syslog Facility: LOCAL0 (16)

Syslog Priority: INFO (6)

Resource ID:

Syslog Tag Value: MWHeartbeat

Rule Set to Use: Write Syslog log file

Message to Send

This is the message that is used as text inside the information unit. Use whatever value is appropriate. There is no check inside EventReporter for a specific value.

Sleep Time

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

Default Ruleset Name

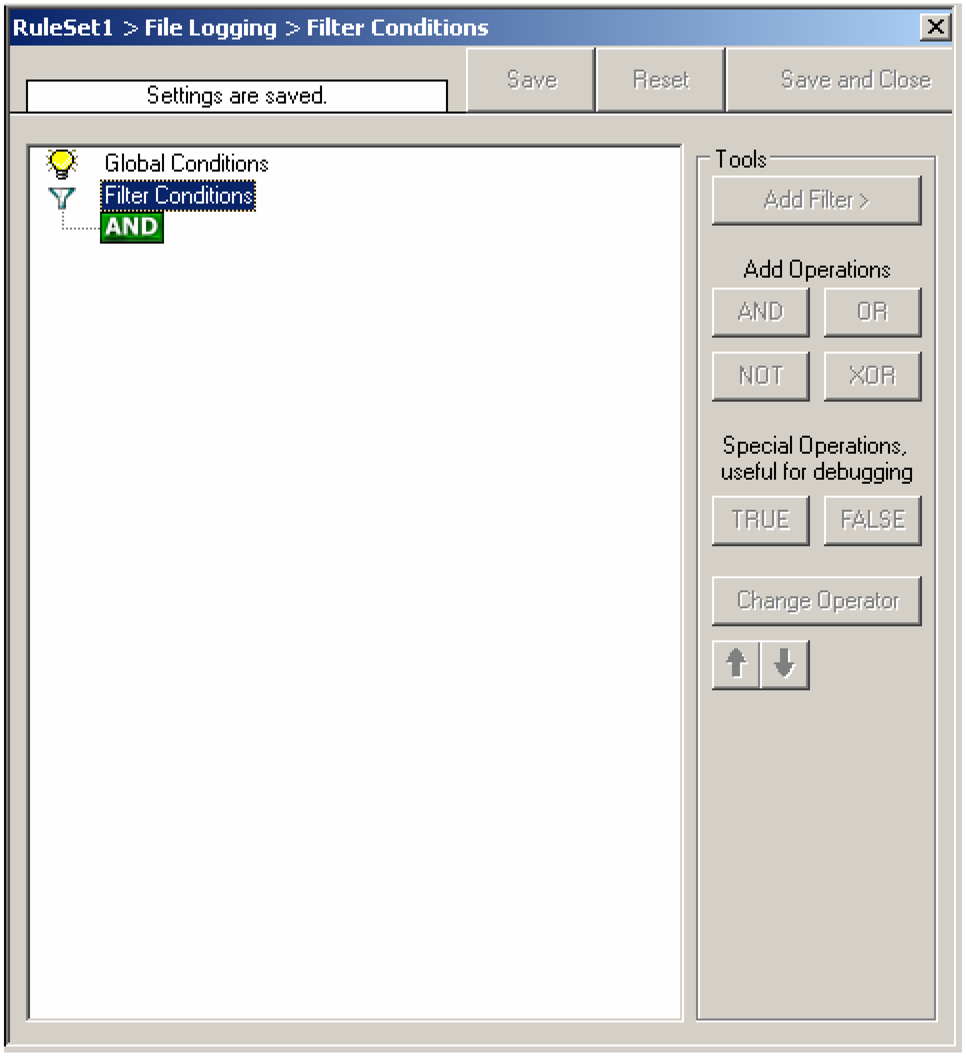
Name of the rule set to be used for this service. The rule set name must be valid.

Filter Conditions

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

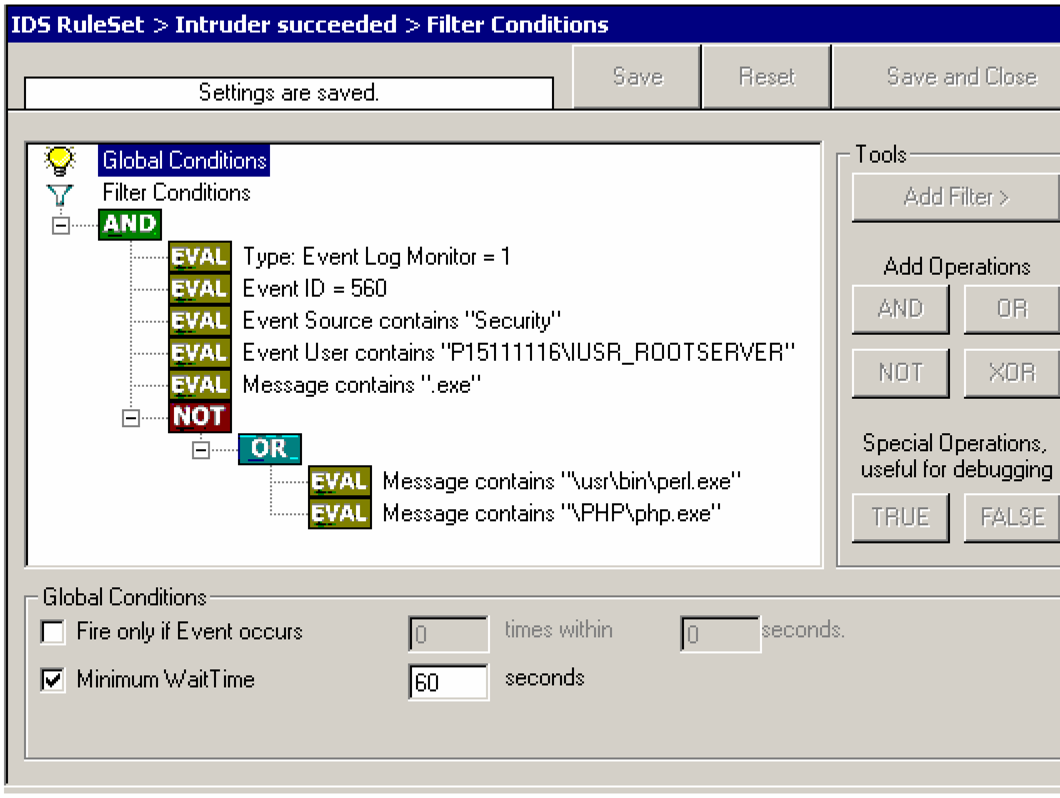
Filter conditions can be as complex as needed. Full support for boolean operations and nesting of conditions is supported.

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



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

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


IDS RuleSet > Intruder succeeded > Filter Conditions

Settings are saved.

Global Conditions
Filter Conditions
AND
- EVAL Type: Event Log Monitor = 1
- EVAL Event ID = 560
- EVAL Event Source contains "Security"
- EVAL Event User contains "P15111116\IUSR_ROOTSERVER"
- EVAL Message contains ".exe"
- NOT
  - OR
    - EVAL Message contains "\usr\bin\perl.exe"
    - EVAL Message contains "\PHP\php.exe"

Global Conditions
- [ ] Fire only if Event occurs 0 times within 0 seconds.
- [✓] Minimum WaitTime 60 seconds

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

Here is how the above sample works: first of all, the message contents is 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 of all, we check if the specific event really occurred. For this, we need to make sure we deal with an Event Log Monitor infounit. Then, these infounits 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.

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.

Fire only if Event occurs

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

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

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

Minimum Wait Time

This filter condition can be used to prevent rules from firing to 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.

Operations

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

AND

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

OR

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

NOT

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

XOR

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

TRUE

Useful for debugging, will just return TRUE.

FALSE

Useful for debugging as well, will return FALSE.

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 InfoUnit is evaluated. Note, if a filter is used that does not apply to the evaluated InfoUnit, it will be just ignored. This gives you the possibility to build one Filter set for several types of InfoUnits.

For details on how filter conditions are evaluated, please see "**Fehler! Verweisquelle konnte nicht gefunden werden.**" on page **Fehler! Textmarke nicht definiert.**.

There are different types of Filter, and so there a different ways in which you can compare them to a value. The following Types exist:

String

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

Number

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

Boolean

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

Time

Can be compared with another time but only with "=".

Below is a List of possible filters, which can be evaluated.

General

These are non-event log specific settings.

Source System

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

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

Message Content

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

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

This filter is of type string.

Status Name and Value (Type=String)

Date/Time

This filter condition is used to check the time frame (and/or day of week in which an event occurred. For example, a syslog message from a Cisco router saying that it dialed up is normal if it occurs during office hours. If it occurs at night, so, it is an alerting signal and an administrator might receive notification of this event (while he might otherwise decide to discard it). This can be done with the time setting.

The following filters are available in detail:

Start time (Type=Time)

End Time (Type=Time)

Run on Monday (Type=Boolean)

Run on Tuesday (Type=Boolean)

Run on Wednesday (Type=Boolean)

Run on Thursday (Type=Boolean)

Run on Friday (Type=Boolean)

Run on Saturday (Type=Boolean)

Run on Sunday (Type=Boolean)

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 InformationUnitType available (shown below).

Syslog (Type=Boolean)

Heartbeat (Type=Boolean)

Event Log Monitor (Type=Boolean)

File Monitor (Type=Boolean)

Ping Probe (Type=Boolean)

Port Probe (Type=Boolean)

NT Services Monitor (Type=Boolean)

Disk Space Monitor (Type=Boolean)

Event Log Monitor

Event log monitor specific filters are grouped here.

Event ID

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

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

This filter is of type number.

Event 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.

EventLog 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.

Actions

Actions are carried out when the filter conditions of a given rule match.

File Options

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

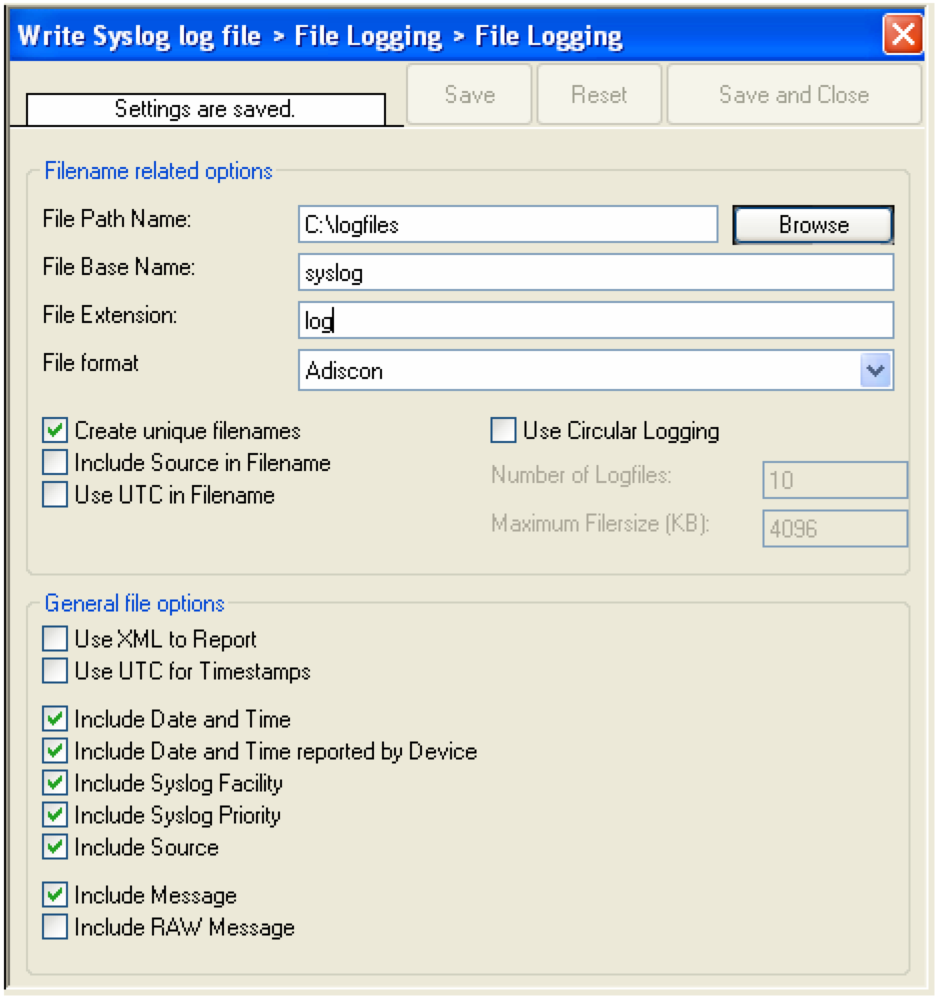
File logging is used to write text files of received messages. One file per day is written. New entries are appended to the end of the file.

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.


Write Syslog log file > File Logging > File Logging

Settings are saved.

Filename related options

File Path Name: C:\logfiles

File Base Name: syslog

File Extension: log

File format: Adiscon

[✓] Create unique filenames
[ ] Include Source in Filename
[ ] Use UTC in Filename
[ ] Use Circular Logging
Number of Logfiles: 10
Maximum Filersize (KB): 4096

General file options

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

File Logging Options

Create unique Filenames

If checked, EventReporter 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. This is used by some customers that have custom scripts to look at the file name.

Click here for a sample screen-shot.

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. The dates recorded inside the file are controlled by a different setting.

Click here for a sample screen-shot.

File Path Name

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

Click here for a sample screen-shot.

File Base Name

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

Click here for a sample screen-shot.

File Extension

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

Click here for a sample screen-shot.

File Format

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

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

The “WebTrends syslog compatible” mimics the format that WebTrends applications expect. Please note that we only mimic the log file format. It is still the job of the reporting device (most notable firewall) to generate the correct WebTrends WELF format. The “WebTrends” format is supported because many customers would like to use EventReporter enhanced features while still having the ability to work with WebTrends.

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

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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 the time the message was received by EventReporter. Therefore, it always is a consistent value.

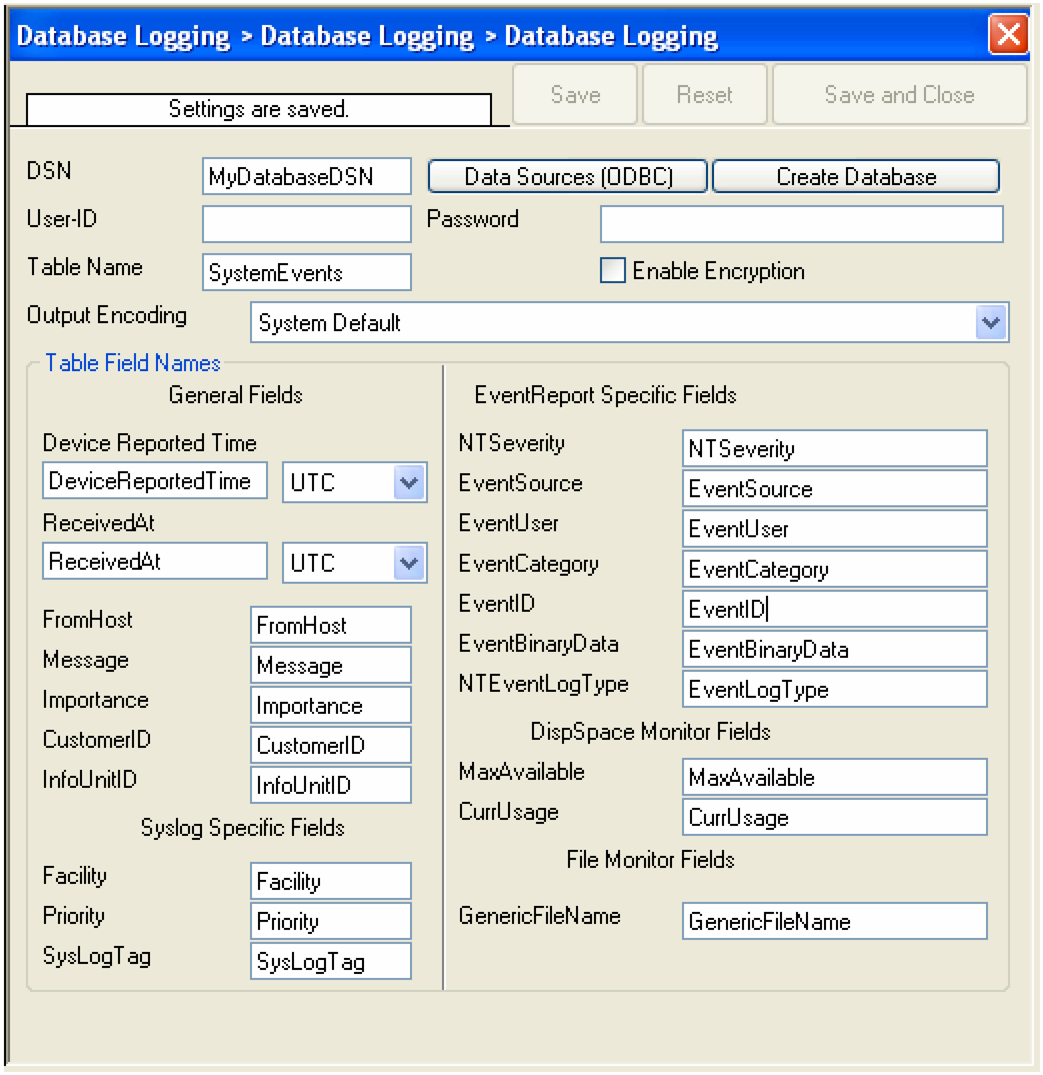
In contrast, the “Date and Time Reported by Device” is a timestamp taken from the actual message. As such, it is dependent on the reporting device clock, which might be off. In addition, in the case of syslog messages, there is no time zone information within the device reported timestamp. As such, if 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 to customize the message part that is being written. The raw message is the message as it was received by EventReporter – totally unmodified. 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 non is selected no message is written at all. Please note that we support these configurations, too – there might be a legitimate need for them.

Click here for a sample screen-shot.

Database Options

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



Settings are saved.

DSN	MyDatabaseDSN
User-ID	
Password	
Table Name	SystemEvents
Enable Encryption	☐
Output Encoding	System Default

Table Field Names

General Fields

Device Reported Time	DeviceReportedTime	UTC
ReceivedAt	ReceivedAt	UTC
FromHost	FromHost	
Message	Message	
Importance	Importance	
CustomerID	CustomerID	
InfoUnitID	InfoUnitID	

Syslog Specific Fields

Facility	Facility
Priority	Priority
SysLogTag	SysLogTag

EventReport Specific Fields

NTSeverity	NTSeverity
EventSource	EventSource
EventUser	EventUser
EventCategory	EventCategory
EventID	EventID
EventBinaryData	EventBinaryData
NTEventLogType	EventLogType

DispSpace Monitor Fields

MaxAvailable	MaxAvailable
CurrUsage	CurrUsage

File Monitor Fields

GenericFileName	GenericFileName

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

DSN

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

Important: The DSN must be a system DSN, not a user or file DSN. The DSN must be configured to have the correct connection parameters (for example database type and name, server name, authentication mode, etc.).

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

Important

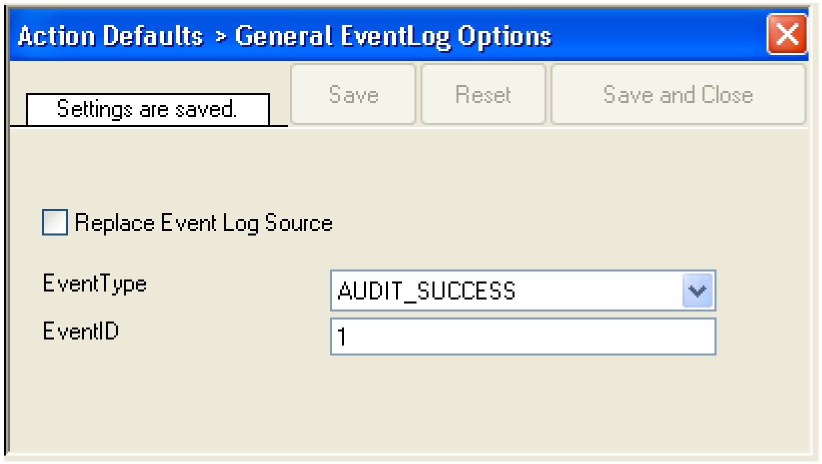
The default name for the message field - "Message" is a reserved name on Sybase database systems. If you would like to log to a Sybase database, you must change that field name. Otherwise, you will receive an ODBC error (visible in NT Event Viewer). We are unfortunately not able to change the default,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

The database conforms to the Common MonitorWare Database Format

For a specification of the database format and samples provided, please see "Database Format" on page 66.

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.



Settings are saved.

Replace Event Log Source: ☐

EventType: AUDIT_SUCCESS

EventID: 1

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.

Click here for a sample screen-shot.

EventType

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

Click here for a sample screen-shot.

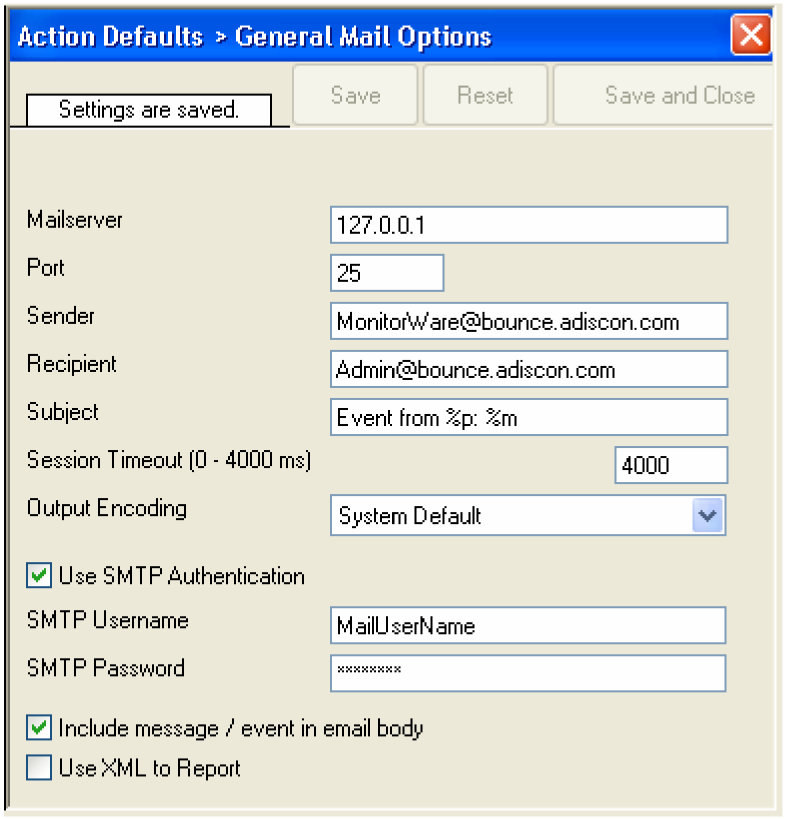
EventID

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 EventReporter itself.

Click here for a sample screen-shot.

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



Mailserver	127.0.0.1
Port	25
Sender	MonitorWare@bounce.adiscon.com
Recipient	Admin@bounce.adiscon.com
Subject	Event from %p: %m
Session Timeout (0 - 4000 ms)	4000
Output Encoding	System Default

[✓] Use SMTP Authentication

SMTP Username	MailUserName
SMTP Password	********

[✓] Include message / event in email body

[ ] Use XML to Report

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

Subject

Subject line to be used for outgoing emails. The subject line is used for each message sent. It can contain replacement characters 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 replacement characters – can hold a maximum of 255 characters. Characters beyond this will be truncated. Please note that some email systems do impose a stricter limit and truncation as such might occur before the 255-character limit.

The following replacement characters can be used 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.

In the example above, replacement characters are being used. 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

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

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

Click here for a sample screen-shot.

Session Timeout

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

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

This is most appropriate when large burst of messages are expected and these should be combined in few mail messages. Otherwise, multiple mail messages can easily overflow the administrator's mailbox.

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

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

Click here for a sample screen-shot.

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 userid 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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

Use XML to Report

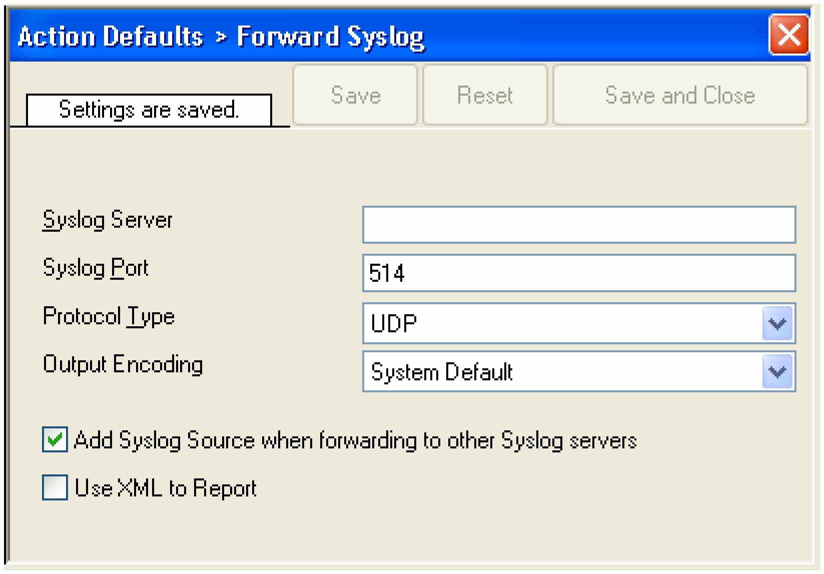
If checked, the received event will be included in XML format in the mail. If so, the event will include **all** information, like the original timestamp, the facility, priority etc. XML format is especially useful if the mail is sent to an automated system, which will then parse the message.

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

Click here for a sample screen-shot.

Forward Syslog Options

This dialog controls syslog forwarding options.



Syslog Server

Syslog Port: 514

Protocol Type: UDP

Output Encoding: System Default

[✓] Add Syslog Source when forwarding to other Syslog servers

[ ] Use XML to Report

Forward Syslog Properties

Syslog Server

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

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

Protocol Type

The Agent can forward messages via either UDP or TCP. The syslog standard allows UDP delivery only. This is also the default. Change it to TCP only if you have a very good reason to do so and you know the receiving server is capable of accepting syslog over TCP.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

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.

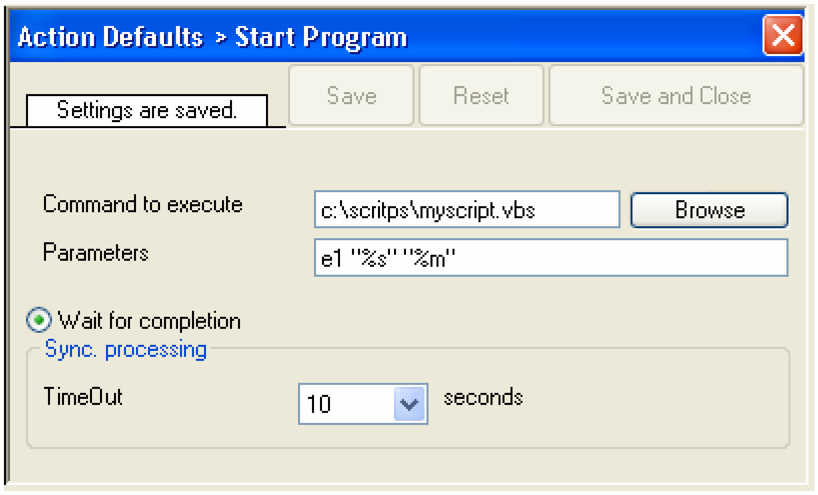
Click here for a sample screen-shot.

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.



Command to execute: c:\scritps\myscript.vbs

Parameters: e1 "%s" "%m"

Wait for completion

TimeOut: 10 seconds

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.

Click here for a sample screen-shot.

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!

Click here for a sample screen-shot.

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.

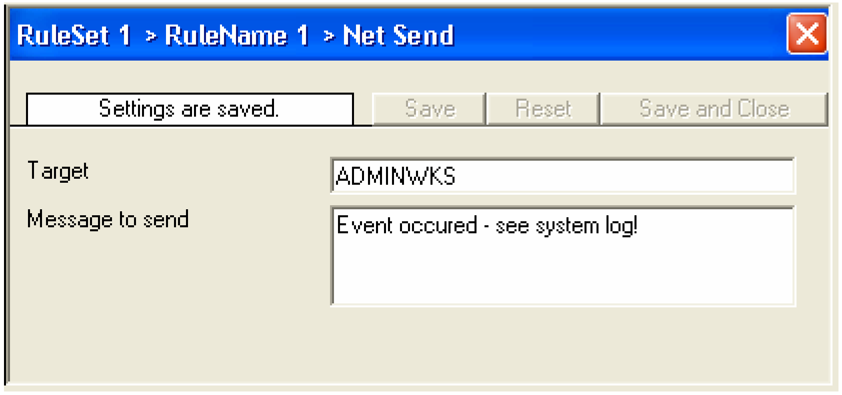
For performance reasons, we also strongly recommend to use the “Start Program” action only for rules that apply relatively seldom.

Click here for a sample screen-shot.

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”.



Target: ADMINWKS

Message to send: Event occured - see system log!

Net Send Dialog

Target

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

Click here for a sample screen-shot.

Message to send

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

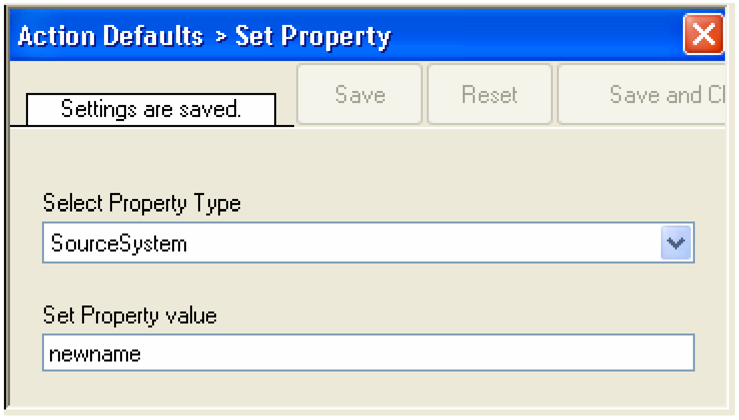
Click here for a sample screen-shot.

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!



Select Property Type: SourceSystem

Set Property value: newname

Set Property Dialog

Select Property Type

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

Click here for a sample screen-shot.

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.

Click here for a sample screen-shot.

Getting Help

TheEventReporter 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. Some of the most important FAQ entries are also included in this manual. However, we recommend using the web site as there might be updates even to the items included in this manual.

EventReporter Web Site

Visit the support area at

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.

Support Forum

Share questions and answers with your peers! The forum is also monitored by Adiscon support staff.

To access the forum, point your browser at

http://forum.adiscon.com/viewforum.php?f=3

Email

Please address all support requests to

support@adiscon.com

An appropriate subject line is highly appreciated.

Online Seminars

Adiscon offers a selection of online seminars. This selection is continuously being expanded. All available seminars can be found at:

http://www.adiscon.com/Common/SeminarsOnline/

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

Phone

+49-2235-985004 (with "+" being the international dialing prefix, for example 011 in the US).

Toll free from the US: 1-888-318-3395

Phone technical support is limited to UpgradeInsurance customers.

Please note that we are in the Central European Time zone (CET). That is 1 hour east of Greenwich Time. If it is 12pm in New York, it is 9pm at our office location. Our office hours are from 9am to 5pm. Therefore, we generally advise US customers to call in early mornings and Asian customers to call in late afternoon.

For best customer service, we highly recommend limiting phone calls to emergencies. We are checking our other support options regularly. Email support is available also during non-office hours, typically until 10pm CET.

Fax

Please direct your faxes to

+49-9349-928820

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

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

Software Maintenance

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

To learn more about UpgradeInsurance, please visit

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

Non-Technical Questions

Please address all non-technical questions to

info@adiscon.com

This email alias will answer all non-technical questions like pricing, licensing or volume orders.

Product Updates

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

Please visit

www.eventreporter.com

for information about new and updated products.

MonitorWare Concepts

Learn what MonitorWare is made for and made of.

The EventReporter offers advanced monitoring capabilities. To fully unleash MonitorWare's power, you need to learn a bit about its concepts. This chapter here has full details.

MonitorWare operates on a set of elements. These are

- Services
- Information Units
- Filter Conditions
- Actions
- Rules
- The SETP Protocol

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

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

Please see license.txt for full license information. This file can be found in the ZIP file and is displayed during installation.

Pricing

The license fee is US$ 129 per server and $59 per workstation. A workstation is a system running Windows NT Workstation, Windows 2000 Professional or Windows XP Professional or Home Edition.

For customers in the "Euro Zone" (European countries using the EURO as official currency), the license fee is EURO 169 per server and EURO $79 per workstation. These prices include 16% VAT, which can be waived if a proper VAT ID number is specified (almost all corporations and organizations inside the EURO zone do have a VAT ID. If in doubt, check with your financial department).

European Community residents with VAT identification number should state this number in order to receive tax exemption. If not stated, full VAT will be charged. All European Community orders will be processed in EURO. US$ payment is available for international customers, only.

Please email Adiscon at sales@adiscon.com if you are interested in a volume order.

How to order

The most convenient way is via our online order processing system found at

https://secure.adiscon.com/EventReporter/en/

If you do not like to order online, registration is still as simple as 1-2-3:

1. Print out the registration form on the order web site
2. Please fill it in. Remember to include number of licenses requested and payment information as well as your email id.
3. Mail or fax the registration form to Adiscon.

We accept all major credit cards. If you would like to place a purchase order, please see

http://www.adiscon.com/Common/en/OrderByPO.asp

for details.

If you need any additional payment options, please contact us at Info@Adiscon.com or the below given addresses.

Direct your orders to:

Adiscon GmbH
Franz-Marc-Strasse 144
50374 Erftstadt
Germany

Fax: +49-9349-928820
Phone +49-2235-985004

email: order@Adiscon.com

All credit card orders need to be processed in Euro. US$ payments will be converted to Euro according to current exchange rate. There might be a slight difference in the converted value due to exchange rate differences.

Order Form

Your order can be placed using the following form. The most current online order form is available at

https://secure.adiscon.com/EventReporter/en/

If you would like to order by mail or fax, please print out the order form and sign it.

Reference

The EventReporter Service

The service operates in the background while your computer is running.

The EventReporter is installed as a system service during setup. It typically runs on each machine being monitored. However, some machines can also be dedicated to run it for housekeeping functions (for example log consolidation).

The EventReporter can be "engine only" installed. In this case, only the service is installed onto a machine. It can be customized either by directly editing the registry or by copying a registry snapshot from a machine with installed client. Please note that "Engine Only" installs need a full EventReporter license.

The EventReporter service program is called "evntslog.exe". It is the sole executable that needs to be distributed for mass rollouts.

The Service Account

NT Services must utilize an NT logon account in order to perform their intended tasks. The EventReporter service is no different. The account initially used by the service is "local system". We recommend retaining this setting.

If for any reason you would like to change the service account, you can do so via the control panel "services" applet (or the "Computer Management" MMC under Windows 2000). However, you need to make sure that the new account has sufficient permissions.

Command Line Switches

The EventReporter supports a limited set of command line switches. These are primarily used for unattended installations or "engine only" installs. These are:

evntslog –h	Help, displays a short usage notice.
evntslog –i	Installs the service
evntslog –u	Removes (uninstalls) the service
evntslog –v	Displays version information as well as whether or not the service is installed.

Support for Mass Rollouts

A “mass rollout” in the scope of this chapter is any case where the product is rolled out to more than 5 to 10 machines.

The common thing among such rollouts is that the effort required to set up the files for unattended distribution of the configuration file and product executable is less than doing the tasks manually. For less than 5 systems, it is often more economical to repeat the configuration on each machine – but this depends on the number of rules and their complexity.

Please note that an automated configuration system is planned, which will enable fully automatic distribution of configuration settings after the initial setup. Please contact info@adiscon.com if you are interested in this system. We let you know when it is available.

Before considering a mass rollout, be sure to read “The EventReporter Service” on page 64. This covers necessary background information.

The basic idea behind a mass rollout is to create the intended configuration on a master (or baseline) system. This system holds the complete configuration that is later to be applied to all other systems. Once that is system is fully configured, the configuration will be transferred to all others.

The actual transfer is done with simple operating system tools. The complete configuration is stored in the registry. Thus, it can be exported to a file. This can be done with the client. In the menu, select “Computer”, then select “Export Settings to Registry File”. A new dialog comes up where the file name can be specified. Once this is done, the specified file contains an exact snapshot of that machine’s configuration.

This snapshot can then be copied to all other machines and put into their registries with the help of regedit.exe.

An example batch file to install the product and configuration on the “other” servers might be:

```
copy \\server\share\evntslog.exe c:\some-local-dir
cd \some-local-dir
evntslog –i
regedit \\server\share\configParms.reg
```

The file “configParams.reg” would be the registry file that had been exported with the configuration client.

Of course, the batch file could also operate off a CD – a good example for DMZ systems which might not have Windows networking connectivity to a home server.

Please note that the above batch file **fully** installs the product – there is no need to run the setup program at all. All that is needed to distribute the service is the evntslog.exe file, which is the core service. For a locked-down environment, this also means there is no need to allow incoming connections over Windows RPC or NETBIOS for an engine only install.

Formats

Database Format

EventReporter stores and expects data in the “MonitorWare Common Database Format”. This format is understood by all members of the MonitorWare line of products.

The database format is easy to implement and does not rely on database-specific features. All event data is stored in a single table.

There are some large textual elements inside that table, namely the message part and the Windows event log binary data part. These entities should be stored as a large text element whenever the database system supports it. For example, under Microsoft SQL Server this is the “text” data type.

Adiscon officially support Microsoft Jet and SQL Server databases. However, all MonitorWare products work with a large variety of databases, including for example Oracle or Sybase. As long as there is a standard ODBC driver available for a given database, it should be usable with MonitorWare.

The default table name as well as all field (column) names can be overwritten with the configuration client. This is most useful if the data is to be included into an already existing database or to solve reserved-name conflicts with not directly supported systems. For example, this needs to be done with Sybase as “message” is a reserved word there. For ease of use, we recommend not to change any of the default names if there is no definite need to do so.

There are samples available for Microsoft Jet (Access) and Microsoft SQL Server.

Database Samples

These sample here implement the MonitorWare Common Database Format in widely used database systems.

Attention Sybase users: the “Message” name is reserved in your database system and cannot be used as a field name. It needs to be changed, otherwise the table create will fail. Be sure to also change it in to client database field name configuration.

JET (MS Access) Sample

A sample JET (Microsoft Access) database file is included in the EventReporter install set. It conforms to the MonitorWare Common Database format.

It is in Microsoft Access 97 format to enhance compatibility. It can be converted to any more current format without any problems. In fact, we recommend using the most current format supported by your system because it offers the best performance. To convert it, please use Microsoft Access.

Microsoft SQL Server Sample

If you would like to create the default database on **Microsoft SQL server**, please use the following script:

```
CREATE TABLE.SystemEvents (
 ID int IDENTITY (1, 1) NOT NULL,
 ReceivedAt datetime NULL,
 DeviceReportedTime datetime NULL,
 Facility smallint NULL,
 Priority smallint NULL,
 FromHost nvarchar (60) NULL,
 Message text,
 NTSeverity int NULL,
 Importance int NULL,
 EventSource nvarchar (60),
 EventUser nvarchar (60) NULL,
 EventCategory int NULL,
 EventID int NULL,
 EventBinaryData text NULL,
 MaxAvailable int NULL,
 CurrUsage int NULL,
 MinUsage int NULL,
 MaxUsage int NULL,
 InfoUnitID int NULL ,
 SysLogTag varchar(60),
 EventLogType varchar(60),
 GenericFileName varchar(60)
)
```

This script should also be easily adaptable to other database systems like Oracle.

When porting the script to other database systems, please note that “nvarchar” is essentially “varchar”. The difference is that data is stored in Unicode which allows storage of non-ANSI characters. Typically, it can be replaced with “varchar” or an equivalent data type without any problems.

XML Format

The following XML tags are used by MonitorWare:

Tag Name	Content Description
iut	This is the InfoUnitType. This uniquely identifies the type of event. This is an integer value.
severity	The NT Event Log severity.
user	The user information that was logged with the event. The constant "N\A" denotes that there has **no** user information been logged with this record. This is most important with Windows event log events.
source	The computer the event originates from. It can be either an IP address or a computer name, depending on the reporting service and its configuration. If it is a name, it can similarly be either the name the system knows itself of, a name taken from a configuration database (like reverse DNS lookup) or an name overridden by an rule.
sourceproc	For Windows event log entries, this is the event source as reported in the event log.
id	The event ID as reported by the reporting service. For example, it is the Windows event log ID for the event log monitor.
msg	This is the message text that comes with the event. For

	example, with Windows event log reports it is the message logged in the event log while with syslog messages it is the actual message text. All events provide a “msg” part – but its format and meaning is largely dependent on the reporting service. We recommend not to parse the msg part, as this can change. All well-known values are available via separate XML tags.
category	A numerical category description. It is a sub-id for the current event and depending on the event source. Currently, it is most useful with Windows event log entries where it represent the numerical category description from the event log.
bdata	Used with few event sources. So far, only Windows event logs generate this tag (if configured to do so). The bdata tag includes large binary data that is associated with the event. For Windows event logs, it is the binary data from the log file. Other event source might use it for similar purposes. It is more a dump-like field and can become very large – so use it wisely.

Version History

Interested how EventReporter evolved and which features are new to this build? Read it here!

This short history shall provide you with some background information about the versions available as well as their pros and cons.

This is user driven software.

Please provide us with your feedback. Many features have become reality with the help of envisioning users!

6.1

Release Date: 2003-09-15

- **Event Log Filter Conditions, new functionality** - Added new compare method "is equal" and "is not equal" for String type Filters, changed Filter EventLog Severity from INT to STRING and added predefined selection for filtering EventLog Severity and EventLog Type. This makes advanced filter conditions much more user friendly and more powerful.
- **Service bugfix** - When logging Event data into a file in legacy format, the trailing “ got lost. This has been corrected now.
- **Usability issues fixed** - Adding Services, Rulesets, Rules and Actions is now also possible from one level lower than before.
- **New Manual** - The new version of EventReporter comes with a streamlined version of the manual that uses online resources.

6.0 Final Release

EventReporter 6.0 Final is available immediately. After months of testing, EventReporter 6 has become very stable. At least because it shares it's core engine from our product MonitorWare Agent. The new version comes with a newly designed enhanced multi-threading background service. It has a new powerful Filter-Engine which allows you to build very complex filters like known from Microsoft Network Monitor and it has many new actions.

UPDATE INSTRUCTIONS: Existing EventReporter 5.x users, please read the following lines.

Because nearly everything changed from basic application design, a special upgrade process has been added into the EventReporter Client. That means after you installed EventReporter 6.0, you will have to start the EventReporter Client first! It will ask you to update your configuration then. Proceed these steps, and you will have all your configuration updated to the new version. Please read the Upgrade Wizard carefully, it will ask you if you want to delete the old configuration after update.

All changes since EventReporter 6.0 RC 1:

- Stability - Minor stability changes were made since 6.0 RC1.
- EventReporter Client - Added more ContentMenu Option to add Services, Actions, RuleSets or Rules.
- EventReporter Client - If you deleted a RuleSet, all Filters of RuleSets below the deleted one were lost. This bug has been corrected now.

Changes since EventReporter 5.x:

- New Powerful Rule based engine - You can know process as many actions in rules as you want.
- New Actions - Flat File Logging, ODBC Database Logging, Start external Programs, Net Send a message and other new actions.
- New Scaleable Filter engine -The new filter engine as very powerful, you can build complex filter conditions like known from Microsoft Network Monitor. If you are new to this kind of filtering, I recommend that you read the Filter Conditions part of the manual before you start to play with the filters.
- Multiple EventlogMonitors - They are possible, but usually one EventlogMonitor is enough.
- Multiple Rule Bases - each EventlogMonitor instance can have a specific rule base assigned. And each Rule base can have multiple Rules with multiple actions.
- Large Burst Support - EventReporter has a unique high performance engine that guarantees message buffering during even the most extreme burst traffic.
- Scaleable Filter engine - The new filter engine as very powerful, you can build complex filter conditions like known from Microsoft Network Monitor.
- Add Comments - You can Add Comments under Services, RuleSets, Rules and Actions now. This is useful if you want to write down some notes.

- New Import / Export functions - It is now possible to Import or Export the registry settings by using a binary format.
- Import / Export RuleSets - You can Import / Export complete RuleSets into a XML Based format (Right click a RuleSet). This can be very useful if you want to duplicate RuleSets for example. The Client uses its own file extension here (.erx = EventReporter XML) which is also bound to the Client. That means double-clicking such a File will automatically invoke the Client to import the RuleSet.
- New Manual - A new manual is included in this version which explains how the new EventReporter works. It is not final yet, so it will be updated soon.

6.0 Release Candidate 1

EventReporter 6.0 RC 1 is available for testing immediately. The new version comes with a newly designed enhanced multi-threading background service (The same engine as Adiscon's known MonitorWare Agent). It has a new powerful Filter-Engine which allows you to build very complex filters like known from Microsoft Network Monitor.

UPDATE INSTRUCTIONS: Existing EventReporter 5.x users, please read the following lines.

Because nearly everything changed from basic application design, a special upgrade process has been added into the EventReporter Client. That means after you installed EventReporter 6.0 RC1, you will have to start the EventReporter Client first! It will ask you to update your configuration then. Proceed these steps, and you will have all your configuration updated to the new version.

More details about the changes below.

- New Powerful Rulebased engine - You can know process as many actions in rules as you want.
- New Actions - Flat File Logging, ODBC Database Logging, Start external Programs, Net Send a message and other new actions.
- New Scaleable Filterengine -The new filter engine as very powerful, you can build complex filter conditions like known from Microsoft Network Monitor. If you are new to this kind of filtering, I recommend that you read the Filter Conditions part of the manual before you start to play with the filters.
- Multiple EventlogMonitors - They are possible, but usually one EventlogMonitor is enough.
- Multiple Rule Bases - each EventlogMonitor instance can have a specific rule base assigned. And each Rulebase can have multiple Rules with multiple actions.
- Large Burst Support - EventReporter has a unique high performance engine that guarantees message buffering during even the most extreme burst traffic.
- Scaleable Filterengine - The new filter engine as very powerful, you can build complex filter conditions like known from Microsoft Network Monitor.

- Add Comments - You can Add Comments under Services, RuleSets, Rules and Actions now. This is useful if you want to write down some notes.
- New Import / Export functions - It is now possible to Import or Export the registry settings by using a binary format.
- Import / Export RuleSets - You can Import / Export complete RuleSets into a XML Based format (Right click a RuleSet). This can be very useful if you want to duplicate RuleSets for example. The Client uses its own file extension here (.erx = Eventreporter XML) which is also bound to the Client. That means double-clicking such a File will automatically invoke the Client to import the RuleSet.
- New Manual - A new manual is included in this version which explains how the new EventReporter works. It is not final yet, so it will be updated soon.

5.4 Final Release

EventReporter 5.4 (Build 163) has been released into public on 2002-10-11. The new version offers the same as well as new enhancements (As bug fixes) as the 5.4 Release Candidate 1

- Added Options for LogType. The event logtype (Like "Application") can now added into a message. There will also a new xml tag for the logtype. See the documentation for more.
- There is a new setup for Windows NT available (witch does not need ServicePack level 6).

Please note that there was a minor update to service build 158 on 2002-01-28 fixing a bug in the advanced filter rules. As it was a minor fix, we did not increase the version number. If you experience problems with the advanced filter rules, make sure you have build 158. This can be seen in Help/About.

5.4 Release Candidate 1

This version (build 160) has been released to the public on 2002-08-02. It contains some fixes and some minor enhancements.

- Add Support for Syslog over TCP. You can now use TCP as transmission protocol for Syslog delivery.
- Replace Computer name - Allows you to change the computer name of Eventlog messages.
- Added new Filter - You can now also filter by EventUser in the advanced filter configuration.

5.3 final release

This version (build 157) has been released to the public on 2002-01-21. It contains some fixes and major enhancements. It is part of all EventReporter download sets beginning at the build date.

- Binary data contained in the event log records can now be (optionally) forwarded to the message receiver.
- Support for sending both email AND syslog messages for the same event. This is done via the advanced filters configuration.

- fixed a bug in Japanese language encoding. Half size kana characters were improperly translated in any mode besides "System Default".
- customizable email subject line - the subject can now contain actual data fields. Great for notifying pagers.
- Events can now be reported in XML format. This allows easy extension with new data fields as they become available. It can also easily parsed with most script languages.
- With the XML format, the category field of the event log can be forwarded
- French language support in the EventReporter client
- This release fully supports all feature of Windows XP and has received the "Designed for Windows XP" logo from Microsoft.
- A number of client improvements - for example, the existing event sources can now be read out by the advanced filter dialog and need not to be manually entered.
- French language support for the EventReporter client
- Added a feature to prevent syslog server and network overload by allowing to throttle the rate by which EventReporter sends out event reports (default is now around 200 events per second maximum).

5.2 SP 2 (service pack 2)

This version (build 152) has been released to the public on 2001-07-16. It contains some fixes and enhancements. It is part of all EventReporter download sets beginning at the build date.

- Removed bug in EventReporter Service. In some cases the event log message from an unregistered event source was not correctly read.
- Enhanced SMTP-Sending support.
- Removed a bug from EventReporter Client. When a Rule with no event source was added using the filter rules editor, the registry-key was also written. So this rule could never apply. Now the key is not written if the event source is empty and the rule applies to all event sources.

5.2 SP 1 (service pack 1)

This version (build 151) has been released to the public on 2001-06-01. It contains some fixes and enhancements. It is part of all EventReporter download sets beginning at the build date.

- Spanish language support added to the EventReporter client
- client can now copy email settings from the system tab to all others
- Fixed a bug that could cause the EventReporter service to abort when it processed an invalidly formatted third party message library

5.2 final release

This version (build 150) has been released to the public on 2001-04-20. It is the official final release of the 5.2 product.

- Fixed a bug in SMTP-Delivery. Emails sent by EventReporter had the wrong timestamp.
- Added a new Setting "Add Username". You can append the Username to all outgoing messages.
- Enhanced the EventReporter Client there are now more possibilities to control the Service.
- New install package based on InstallShield - service is now automatically installed at setup time

5.1 final release

This version (build 148) has been released to the public on 2001-03-01. It is the official final release of the 5.1 product.

It contains all build 147 features as well as one important fix and a new feature:

- Fixed a bug that could cause EventReporter to fail if DNS resolution was not available for the SMTP server specified. If that happened, EventReporter will ungracefully shutdown via a general protection fault. This has been fixed in build 148 - a matching error will now be reported to the event log and EventReporter will re-try in its next iteration.
- The client can now automatically check for product updates. To do so, it connects to Adiscon's real time product status system on the Internet.

5.1 Beta 1

This version (build 147) has been released to the public on 2001-02-07. It is a first publicly available 5.1 release. It is a beta build.

It contains the following fixes and enhancements:

- Fully based on Unicode. This enables the product to process all international characters in the Windows NT event logs. Also further speeds up the product, as Unicode-based processes can be executed faster by Windows NT (NT itself is based on Unicode - non-Unicode applications need to use a translation layer which takes it's time toll).
- Selectable output character encoding supporting system standard, EUC-JP and JIS. Specifically build in for the Japanese market.
- Syslog messages can now be consecutively numbered. So it is easy to detect if someone has deleted some messages from the syslog files.
- Remote Administration - client can now manage remote instances of EventReporter as long as there is network connectivity between the client and the machine the service is running on.
- Time savers in client: "Reload" and "Restore to Default" buttons. Customer feedback told us these were really missing!
- some minor bug fixes

5.0 Final Release

This version (build 145) has been released to the public on November, 29th 2000. It is a first publicly available 5.0 release. It is a production build without any notable differences from Beta 2.

5.0 Beta 2

This version (build 142) has been released to the public on November, 29th 2000. It is a first publicly available 5.0 release. It is a beta build.

It contains the following fixes and enhancements:

- Local pre-filtering has been added both to the service and the client. The product can now filter based on the event id, source, severity and category. Very complex filter rules can be created in order to care for the most demanding environments.
- localized client - immediately available in English, German and Japanese
- XML based system for localization - new client languages can be added very easily. Customers interested in a new language and willing to provide the translation of a few strings are very welcome to contact us.
- Fixed a problem that could cause the service to abort when using SMTP delivery and the mail server was offline.
- New, improved setup program based on Microsoft Windows Installer - the new standard for both manual and automated software installation.

4.2

This version (build 140) has been released to the public on November, 6th 2000. It is a final release, mainly for maintenance reasons.

It contains a the following fixes:

- daemon process runs at below normal thread priority, further lowering the performance impact on the monitored server
- fixed a bug that caused incorrect formatting of event messages in some scenarios under Windows 2000
- added support for misbehaving message libraries (If an external product message library failed, the EventReporter process could be aborted. We have added structured exception handling to catch this exception. This error occurred very rarely and only in conjunction with failing third-party products.)
- support for double byte character sets added

All users are strongly encouraged to upgrade to this release.

4.1.1

This version (build 139) has been released to the public on September, 29th 2000. It is a final release fixing a single problem.

It contains a the following fixes:

- fixed a bug that caused a debugging message to be reported in syslog delivery mode

4.1

This version (build 138) has been released to the public on September, 25th 2000. It is a final release, mainly for maintenance reasons.

It contains a the following fixes:

- fixed a bug that caused the installation to abort in some environments
- fixed a bug that caused the email delivery mode to fail (problem reported was "Fatal error (0) initializing transport", occurred only with a very limited number of mail servers)

4.0

This version (build 137) has been released to the public on July, 1st 2000. It is a final release.

It contains a the following enhancements:

- support for message delivery via email (complete delivery system and abstraction layer added)
- EventReporter Client added
- filtering of events based on severity code (e. g. error, warning, informational)
- greatly enhanced documentation
- greatly enhanced web site - especially support area

3.2

This version (build 136) has been released to the public on February, 14th 2000, just in the timeframe for Windows 2000 (release February, 17th 2000 by Microsoft). It is a final release.

It contains a single enhancement:

- automatic support for the 3 additional Windows 2000 event logs ("DNS Server", "File Replication Service", "Directory Service". This support is automatically enabled if Windows 2000 is detected.

3.1

This version (build 134) has been released to the public on November, 24th 1999. It is a final release.

This version is much improved. It contains the following enhancements:

- support for Windows 2000 (NT 4 type event logs, only)
- all buffers increased for much larger message sizes
- fixed a bug that caused event message IDs to become erratically negative
- allows to control whether or not log truncations cause notifications to be sent

- enhanced output format (higher compatibility layer to syslog) as well as added severity information in a standard header
- fixed a bug with message library handling that could cause truncated messages in some uncommon situations

3.0 release version

This version (Build 121) has been released to the public on November, 14th 1998.

There are no functional upgrades when compared to 3.0 beta 3 except that the build ID is displayed when installing the service.

This is the fully tested and officially released version.

3.0 beta 3

This version has been released to the public on November, 1st 1998.

Beta 2 was a private build. Beta 3 contains some fixes. This version is scheduled to become the final release.

- Beta 1 lost a couple of log messages if there was a high logging demand. Beta 3 fixes that.
- Performance has greatly been improved by avoiding unnecessary writes during each iteration
- Fixed a bug which caused EvntSLog to stop logging when the event log was reset. Please note: this was not a beta bug, but a bug contained in all previous releases of EvntSLog.
- Fixed a bug which caused EvntSLog to become unresponsive during longer sleep periods (> 15 seconds). EvntSLog is now a multithreaded service.
- Location of registry parameters was incorrectly printed during service installation (evntslog -i). This was mere a cosmetic problem, but caused user confusion.
- Release history in documentation improved.

3.0 beta 1

This version has been released to the public on September, 22nd 1998.

This version is much improved. It contains the following enhancements:

- multiple spaces inside a log record are compressed to a single one
- control characters (like CR/LF) are filtered out of the eventlog messages. These characters caused truncated log entries on some syslogds.
- Syslog facilities are now supported. A separate facility can be set for the system, security and application event log. By default, all logs are reported to LOCAL0. Facility setting can be modified by 3 new registry settings: System\nFacility, Application\nFacility, Security\nFacility. These settings are the decimal number of the facility code.
- Syslog priorities are now supported. Windows NT event types are mapped to syslog priorities based on the following mapping table: EVENTLOG_AUDIT_SUCCESS, EVENTLOG_INFORMATION_TYPE

==> LOG_NOTICE;
EVENTLOG_AUDIT_FAILURE, EVENTLOG_WARNING_TYPE ==> LOG_WARNING;
EVENTLOG_ERROR_TYPE ==> LOG_ERR;

- Licensing via licensee name and license key. There is only a single executable for both the trial and the licensed version. This way, a trial installation can become a fully licensed one with even less effort.
- support for the Compaq/DEC ALPHA platform

2.1

Fixed a bug when using NT SP 4 or some post SP3 hotfixes. These caused EvntSLog to crash when trying to access the security log.

2.0 release version

There are no functional upgrades when compared to 2.0 beta 2b. However, this is the fully tested and officially released version.

2.0 beta 2b

- Parameter szTarget2 added for enhanced robustness of the logging process
- The EventID is now written to the syslog host, too (in parentheses after the source name)

Please note that this version is beta software, too.

2.0 beta 2a

Contains one quick enhancement of the installation process. Released shortly after the initial 2.0 beta 2. Didn't change any core functionality.

- Service installations writes default registry parameters

Please note that this version is beta software, too.

2.0 beta 2

The next version publicly offered. It provides these enhancements:

- Runs as Windows NT system service.
- All three parts (system, security and application) of the event log can now be processed
- fixes some minor and one serious version 1.0 bug. With 1.0, event logging could stop if a very large log record (> 2 K of data) should be processed. This has been removed in 2.0 beta 2.
- Documentation has been changed to HTML only

Please note that this version is beta software.

1.0

This is the initial release. Made available on March, 23rd 1997. It provides all the basic functionality, but has some restrictions:

- There is no configuration program
- Only the system event log can be read
- Only a single syslog daemon is supported
- It does not run as a service

ICMP Codes

ICMP codes are often used when doing firewall and/or router diagnostics. For convenience, find an excerpt from RFC1700 below. The full RFC can be obtained from several places, for example at

http://www.ietf.org/rfc/rfc1700.txt

ICMP TYPE NUMBERS

The Internet Control Message Protocol (ICMP) has many messages that are identified by a "type" field.

Many of t

These ICMP types have a "code" field. Here we list the types again with their assigned code fields.

Type	Name	Reference
0	Echo Reply	[RFC792]
1	Unassigned	[JBP]
2	Unassigned	[JBP]
3	Destination Unreachable	[RFC792]
4	Source Quench	[RFC792]
5	Redirect	[RFC792]
6	Alternate Host Address	[JBP]
7	Unassigned	[JBP]
8	Echo	[RFC792]
9	Router Advertisement	[RFC1256]
10	Router Selection	[RFC1256]
11	Time Exceeded	[RFC792]
12	Parameter Problem	[RFC792]
13	Timestamp	[RFC792]
14	Timestamp Reply	[RFC792]
15	Information Request	[RFC792]
16	Information Reply	[RFC792]
17	Address Mask Request	[RFC950]
18	Address Mask Reply	[RFC950]
19	Reserved (for Security)	[Solo]
20-29	Reserved (for Robustness Experiment)	[ZSu]
30	Traceroute	[RFC1393]
31	Datagram Conversion Error	[RFC1475]
32	Mobile Host Redirect	[David Johnson]
33	IPv6 Where-Are-You	[Bill Simpson]
34	IPv6 I-Am-Here	[Bill Simpson]
35	Mobile Registration Request	[Bill Simpson]
36	Mobile Registration Reply	[Bill Simpson]
37-255	Reserved	[JBP]

```
Type    Name                                    Reference
----    -------------------------               ---------
  0     Echo Reply                               [RFC792]

        Codes
            0  No Code

  1     Unassigned                                  [JBP]

  2     Unassigned                                  [JBP]

  3     Destination Unreachable                  [RFC792]

        Codes
            0  Net Unreachable
            1  Host Unreachable
            2  Protocol Unreachable
            3  Port Unreachable
            4  Fragmentation Needed and Don't Fragment was Set
            5  Source Route Failed
            6  Destination Network Unknown
            7  Destination Host Unknown
            8  Source Host Isolated
            9  Communication with Destination Network is
               Administratively Prohibited
           10  Communication with Destination Host is
               Administratively Prohibited
           11  Destination Network Unreachable for Type of Service
           12  Destination Host Unreachable for Type of Service

  4     Source Quench                           [RFC792]
        Codes
            0  No Code

  5     Redirect                                [RFC792]

        Codes
            0  Redirect Datagram for the Network (or subnet)
            1  Redirect Datagram for the Host
            2  Redirect Datagram for the Type of Service and Network
            3  Redirect Datagram for the Type of Service and Host

  6     Alternate Host Address                     [JBP]

        Codes
            0  Alternate Address for Host

  7     Unassigned                                 [JBP]

  8     Echo                                    [RFC792]

        Codes
            0  No Code

  9     Router Advertisement                   [RFC1256]

        Codes

            0  No Code

 10     Router Selection                       [RFC1256]

        Codes
            0  No Code

 11     Time Exceeded                           [RFC792]

        Codes
            0  Time to Live exceeded in Transit
            1  Fragment Reassembly Time Exceeded

 12     Parameter Problem                       [RFC792]

        Codes
            0  Pointer indicates the error
            1  Missing a Required Option       [RFC1108]
            2  Bad Length

 13     Timestamp                               [RFC792]

        Codes
            0  No Code

 14     Timestamp Reply                         [RFC792]

        Codes
            0  No Code

 15     Information Request                     [RFC792]

        Codes
            0  No Code

 16     Information Reply                       [RFC792]

        Codes
            0  No Code
```

```
17      Address Mask Request                      [RFC950]

        Codes
            0  No Code

18      Address Mask Reply                        [RFC950]

        Codes
            0  No Code

19      Reserved (for Security)                     [Solo]

20-29   Reserved (for Robustness Experiment)         [ZSu]

30      Traceroute                               [RFC1393]

31      Datagram Conversion Error                [RFC1475]

32      Mobile Host Redirect               [David Johnson]

33      IPv6 Where-Are-You                  [Bill Simpson]

34      IPv6 I-Am-Here                      [Bill Simpson]

35      Mobile Registration Request         [Bill Simpson]

36      Mobile Registration Reply           [Bill Simpson]
```

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 www.adiscon.com/en/products/. To obtain information on the complete MonitorWare line of products, please visit www.MonitorWare.com.

Please note that EventReporter is part of the MonitorWare line of products. Please visit the MonitorWare site (www.MonitorWare.com) to receive updates and information on all members of the family. The site also does have information on combining the individual components to build a complex distributed configuration.

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.

Glossary of Terms

Index