Event Reporter 16.1

Table of Contents

1   Introduction

1.1   About EventReporter

[EventReporter](#) is an integrated, modular and distributed solution for system management.

Microsoft Windows 2003™, Windows XP™, Windows Vista™, Windows 2008™, Windows 7™, Windows 8™, Windows 2012™, Windows2016™ and Windows 10™ are highly capable operating systems (we will call all of them "Windows" 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 Windows system event log. Messages can be delivered via email and [syslog](#) protocol.

The initial product - called EvntSLog - was specifically written with mixed Windows 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 Windows event logs into their central syslog setup. Administrative duties and exception notification can easily be built via Unix-based scripting.

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

EventReporter can be teamed with other MonitorWare line of products. In this scenario, it provides a totally centralized and automated event log collection, monitoring and analysis solution. If you are looking for a solution that not only can forward event information but also monitor additional system settings, you might want to have a look at the [MonitorWare Agent](#).

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

The product is easy to install and configure, uses only minimal system resources and is proven to be reliable. Furthermore, it is extremely inexpensive with a per system licensing fee starting at US$ 59.

1.2   Features

Centralized Logging

This is the key feature. EventReporter allows consolidation of multiple Windows 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

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

SETP Support

SETP was originally developed for MonitorWare but now it is a key feature added in EventReporter 6.2 Professional Edition. Windows Event Messages can be forwarded using SETP protocol. Click here for more information on SETP.

Email Support

Windows 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 Windows event log type (e.g. "System" or "Application") as well as severity.

IPv6

Support for IPv6 is available in all network related facilities of the engine. All network related actions will automatically detect IPv6 and IPv4 target addresses if configured. You can also use DNS resolution to resolve valid IPv6 addresses. Network related Services can either use IPv4 or IPv6 as internet protocol. In order to support both protocols, you will need to create two services. The only exception is the RELP Listener, which uses IPv4 and IPv6 automatically if available.

Full Windows 2000*, 2003, XP, Vista, 2008, 7, 8, 10, 2012 and Windows 2016 Support

We had full Windows 2000*, 2003, XP, Vista, 2008, 7, 8, 10, 2012 and Windows 2016 support since these products were released! All extended Windows 2000 log information can be gathered, fully decoded and processed. Custom event logs can also be processed.

Robustness

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

Remote Administration

The client interface 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, its footprint is barely traceable. This ensures it can also be installed on heavily loaded servers.

Full Windows Event Log Decoding

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

Windows Service

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

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

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

Multi-Language Client

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

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

Friendly User Interface

New Cloning feature has been also added to the EventReporter Client. In short you can now clone a Ruleset, a Rule, an Action or a Service with one mouse click. Move up and Move down function has been added for Actions in the EventReporter Client. The EventReporter Client Wizards has been enhanced for creating Actions, Services and RuleSets. And other minute changes!

Handling for low-memory cases

MWAgent allocates some emergency memory on startup. If the system memory limit is reached, it releases the emergency memory and locks the queue. That means not more items can be queued,

this prevents a crash of the Agent and the queue is still being processed. Many other positions in the code have been hardened against out of memory sceneries.

* Support for Windows 2000 and other EOL operating systems is only partially available. Only a minimal service installation may be possible. More details: Information for a Mass Rollout

1.3   Components

EventReporter Client

The EventReporter Client is used to configure all components and features of EventReporter. The client can also be used to create a configuration profile on a base system. That profile can later be distributed to a large number of target systems.

EventReporter Service

The EventReporter Service - called "the service" runs as an Windows 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 the "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 Windows 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 Windows 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 using sleep periods of 500 milliseconds or less (although possible).

x64 Build

The installer inherits the 32bit as well as the 64bit edition. It determines directly, which version is suitable for your operating system and therefore installs the appropriate version. Major compatibility changes for the x64 platform have been made in the Service core. For details see the changes listed below:

- ODBC Database Action fully runs on x64 now. Please note that there are currently very few ODBC drivers for x64 available!
- Configuration Registry Access, a DWORD Value will now be saved as QWORD into the registry. However the Configuration Client and Win32 Service Build can handle these data type and convert

these values automatically into DWORD if needed. The Configuration Client will remain a win32 application. Only the Service has been ported to the x64 platform.

A note on cross updates from Win32 to x64 Edition of EventReporter!

It is not possible to update directly from Win32 to x64 Edition using setup upgrade method. The problem is that a minor upgrade will NOT install all the needed x64 components. Only a full install will be able to do this. Therefore, in order to perform a cross update, follow these instructions:

1. Create a backup of your configuration, save it as registry or xml file (See the Configuration Client Computer Menu)
2. Uninstall EventReporter.
3. Install EventReporter by using the x64 Edition of the setup.
4. Import your old settings from the registry or xml file.

1.4 System Requirements

EventReporter requires very limited resources of the machine to run optimally. The actual minimum requirements to run the application depend on the type of installation. If only the client is installed, they are a bit higher otherwise the service needs a few enabling it to run on a large variety of machines – even the highly utilized ones.

Client

- The client can be installed on Windows 2003 and above. This includes Windows XP, Windows 2008/2012/2016 servers, Windows Vista and Windows 7/8/10. The operating system variant (Workstation, Server …) is irrelevant.
- The client is suited for 32bit and 64bit operating systems. It runs automatically on each platform in 32Bit or 64Bit mode.
- The client uses Microsoft .Net Framework technology. The Installer will automatically install the necessary .Net Framework components before installation. A network connection maybe required in order to download additional components.
- The client requires roughly 8 MB RAM in addition to the operating system minimum requirements. It also needs around 5 MB of disk space.

Service

- The service has fewer requirements.
- It works under the same operating system versions.
- At runtime, the base service requires 5 MB of main memory and less than 5 MB of disk space. However, the actual resources used by the agent largely depend on the services configured.
- Please note that the 32Bit Service is limited to 2GB of usable memory. The 64Bit version does not have any limit. A typical Syslog message (including overhead) takes roughly 4-8 KB. With 1024 MB, you can buffer up to 100,000-200,000 messages in 1024 MB.
- Under Windows 2003*, the 3 additional event logs ("DNS Server", "File Replication Service" and "Directory Service") are automatically supported.

2 Getting Started

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

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

2.1 Installation

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

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

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

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

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

2.1.1 Information for a Mass Rollout

A mass rollout in the scope of this topic is any case where the product is rolled out to more than 5 to 10 machines and this rollout is to be automated. This is described first in this article. A special case may also be where remote offices shall receive exact same copies of the product (and configuration settings) but where some minimal operator intervention is acceptable. This is described in the second half of this article.

The common thing among mass 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 you can also export and re-import configuration settings, so a hybrid solution may be the best when a lower number of machines is to be installed (normal interactive setup plus import of pre-created configuration settings).

Before considering a mass rollout, be sure to read "The EventReporter Service". This covers necessary background information and most importantly the command line switches.

Automated Rollout

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 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 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, configure and run the service on "other" servers might be:

```
copy \\server\share\evntslog.exe c:\some-local-dir
copy \\server\share\evntslog.pem c:\some-local-dir
cd \some-local-dir
evntslog -i
regedit /s \\server\share\configParams.reg
net start "Adiscon EvntSLog"
```

Please note: These files are needed if you are using EventReporter 13.0 and above. If you are using a older version, you additionally need the files "libeay32.dll" and "ssleay32.dll".

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

Also, in this file, the service name can be changed to a different name if needed. When the configParams.reg is imported, then the service name will be set as specified in the Windows Services snap-in.

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 i.e. evntslog.exe and its helper dlls, which are 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.

Please also note that, in the example above, "c:\some-local-dir" **actually is the directory where the product is being installed**. The "evntslog -i" does not copy any files - it assumes they are already at their final location. All "evntslog -i" does is to create the necessary entries in the system registry so that the EventReporter is a registered system service.

Branch Office Rollout with consistent Configuration

You can use engine-only install also if you would like to distribute a standardized installation to branch office administrators. Here, the goal is not to have everything done fully automatic, but to ensure that each local administrator can set up a consistent environment with minimal effort.

You can use the following procedure to do this:

Do a complete install on one machine.

Configure that installation the way you want it.

Create a .reg file of this configuration (via the client program).

Copy the evntslog.exe, evntslog.pem, Microsoft.VC90.CRT.manifest, msvcm90.dll, msvcp90.dll, msvcr90.dll and .reg file that you created to a CD (for example). Take these executable files from the install directory of the complete install done in step 1 (there is no specfic engine-only download available).

Distribute the CD.

Have the users create a directory where they copy all files. This directory is where the product is installed in - it may be advisable to require a consistent name (form an admin point of view - the product does not require this).

Have the users run "evntslog -i" from that directory. It will create the necessary registry entries so that the product becomes a registered service.

Have the users double-click on the .reg file to install the pre-configured parameters (step 3).

Either reboot the machine (neither required nor recommend) or start the service (via the Windows "Servcies" manager or the "net start" command).

Important: The directory created in step 6 **actually is** the program directory. Do not delete this directory or the files contained in it once you are finished. If you would do, this would disable the product (no program files would be left on the system).

If you need to update an engine-only installation, you will probably only upgrade the master installation and then distribute the new exe files and configuration in the same way you distributed the original version. Please note that it is **not** necessary to uninstall the application first for an upgrade - at least not as long as the local install directory remains the same. It is, however, vital to **stop** the service, as otherwise the files can not be overwritten.

2.2   Obtaining a Printable Manual

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

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

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

2.3   Export Settings

When working on a support incident, it is often extremely helpful to re-create a customer environment in the Adiscon lab. To aid in this process, we have added functionality to export an exact snapshot of a configuration. There are multiple methods available. Adiscon Support prefers Adiscon Config Format. Please note that when we have received your file, we are also able to make adjustments (if needed) and provide those back to you. This is a very helpful support tool.
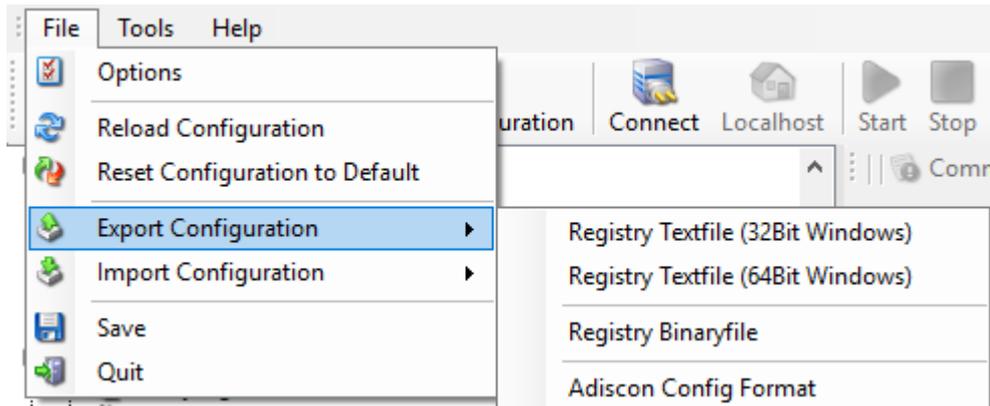
Figure1: Export Settings to a file

To use it, please do the following:

1.   Go to "File->Export Configuration"
2.   Choose "Adiscon Config Format".
3.   Save the configuration file.

You may be reluctant to send the registry file because of security reasons. We recommend you to review the contents of the registry file for security purposes with a notepad or any other text editor.

Please Note: We have a 10 MB limit on our mail account. Please zip the registry file and then send it to us. If the file size doesn't reduce after compressing it you should contact Adiscon Support for further instructions.

2.4   EventReporter Tutorial

This tutorial provides a rough overview of EventReporter as well as some of its typical uses. It is in no way complete, but helps in understanding EventReporter and how it can be configured to suit your needs.

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

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

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

2.4.1 Filter Conditions

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

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

If you are note so sure about the Boolean operators, you might find the following brush-up helpful:

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

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

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

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

TRUE – returns true.

FALSE – returns false.

2.4.2 Ignoring Events

There are some events which occur often and you do not want them to be stored in your log files or either take any action on those.

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

In this tutorial, we define a filter that discards such events. In our example, we assume that Events with the ID 105, 108 and 118 are not required. Please note that for simplicity reasons we only filter based on the event ID. In a production environment, you might want to add additional properties to the filter set.
In this sample, no service or rule set is yet defined. It is just a "plain" system right after install, as can be seen in the following screen shot:
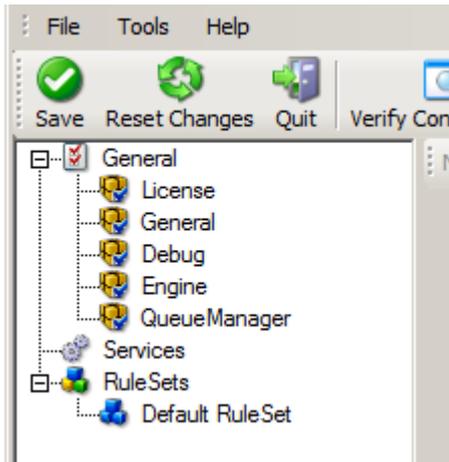
Ignoring Events - Figure 1

We begin by defining a rule set. Right-click on "RuleSets" and choose "Add RuleSet" from the context menu. Type in a name of your choice. In this tutorial, we use the name "Defaults". Click on "Next". Leave all as it is in the next dialog. Click "Next", then "Finish". As can be seen in following screen shot, the rule set "Defaults" has been created but is still empty.
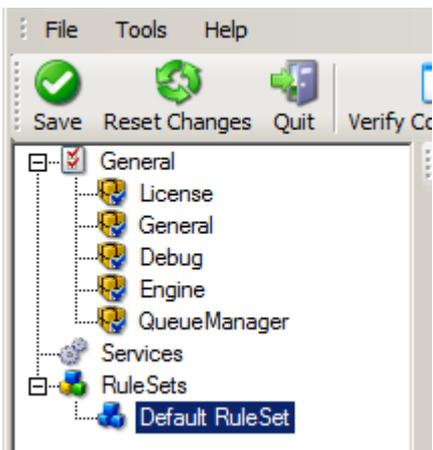
Ignoring Events - Figure 2

Of course we can only use a rule if we configure a corresponding service. To do so, right-click on "Running Services" and then select "Add Services". Choose the desired "Service" from the context menu i.e. "Event Log Monitor" in this sample. Provide a name of your choice. In our sample, we call the service "Event Log Monitor". Leave all defaults and click "Next", then "Finish". Now click on "Event Log Monitor" under "Running Services". Your screen should look as follows:
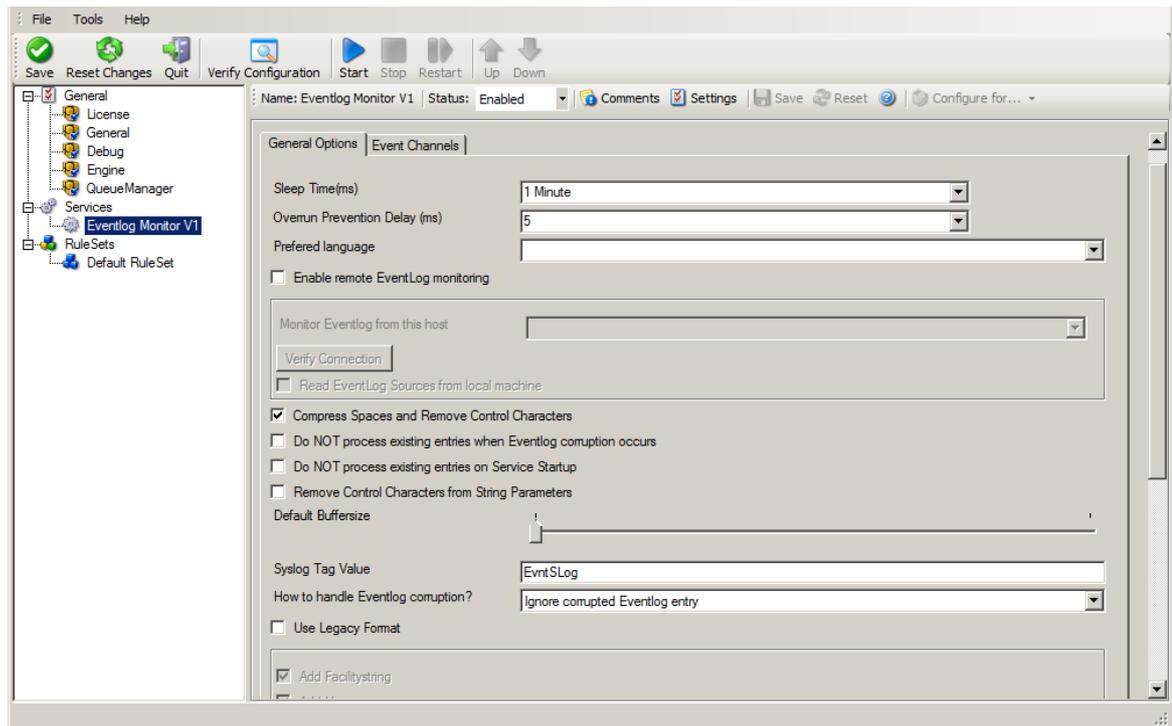
Ignoring Events - Figure 3

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

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

In the tree view, right-click "Defaults" below "RuleSets". Then, click "Rules", select "Add Rule". Choose any name you like. In our example, we call this rule "Discard". Then, expand the tree view until it looks like the following screen shot:
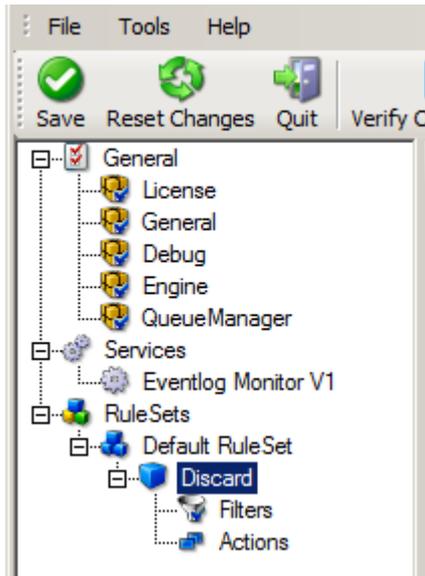
Ignoring Events - Figure 4

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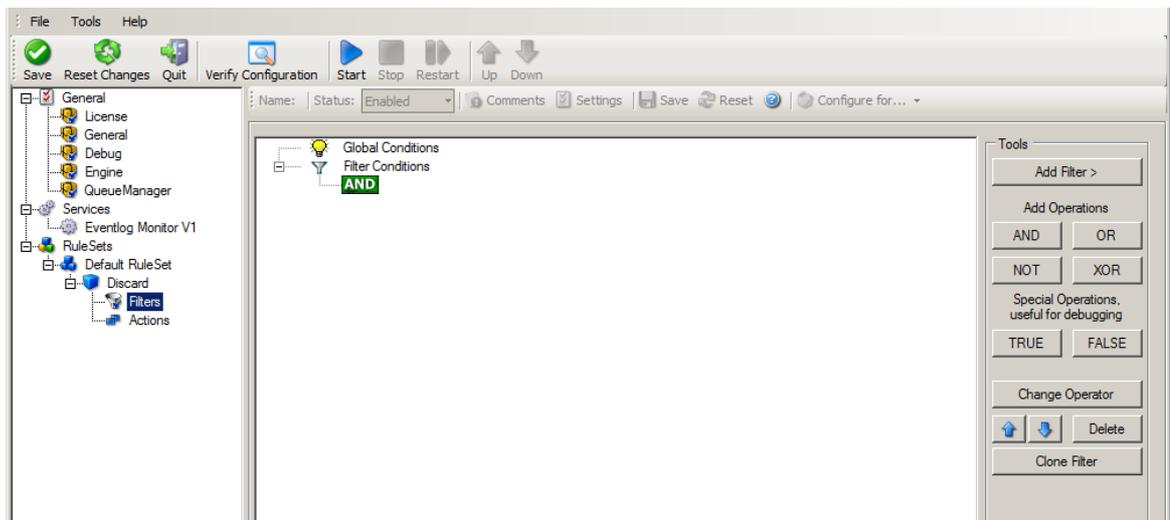
Ignoring Events - Figure 5

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

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

We filter out "uninteresting" events via their event id. Again, there are different ways to do this. In the sample, we do it via right clicking the "OR" node and selecting "Add Filter" from the pop up menu. Or you can use the Add Filter Button. This can be seen in the screen shot below:
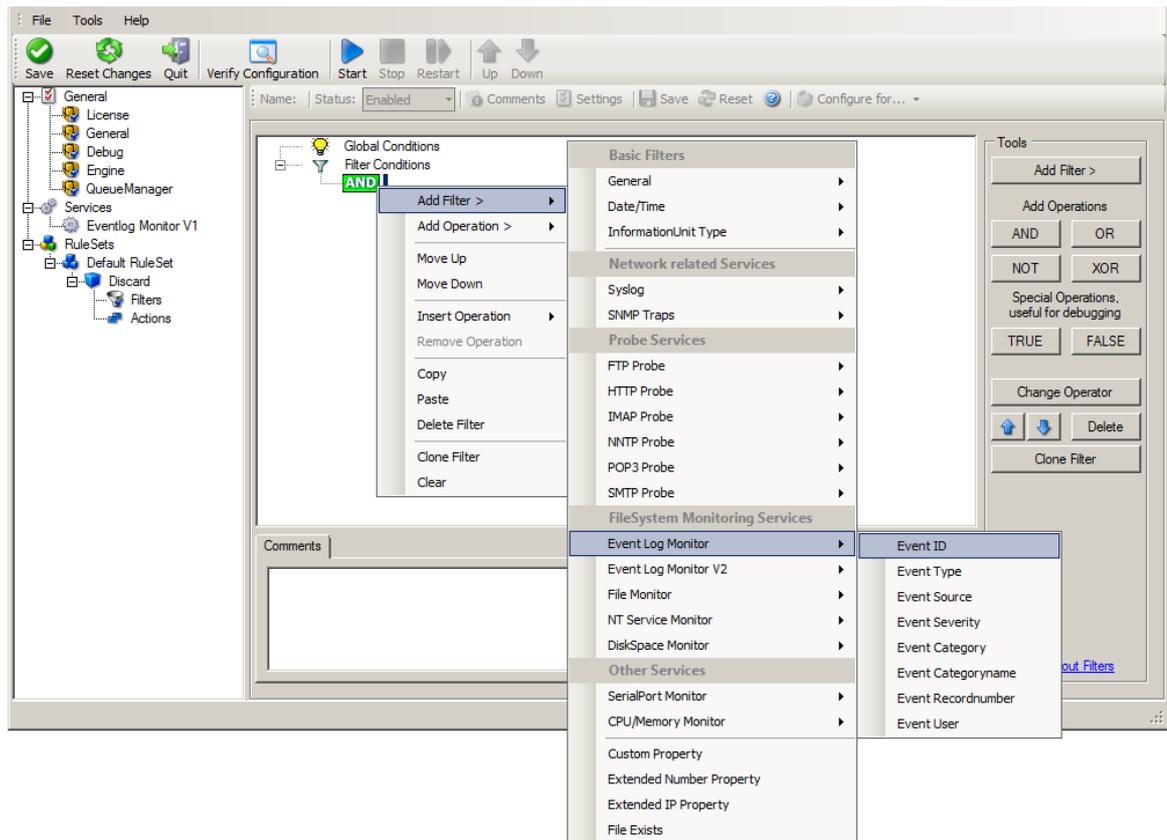
Ignoring Events - Figure 6

I prefer to add all three Event ID property filters first and later on change the Event ID to the actual value I am looking for. When you have added them, it should look as follows:
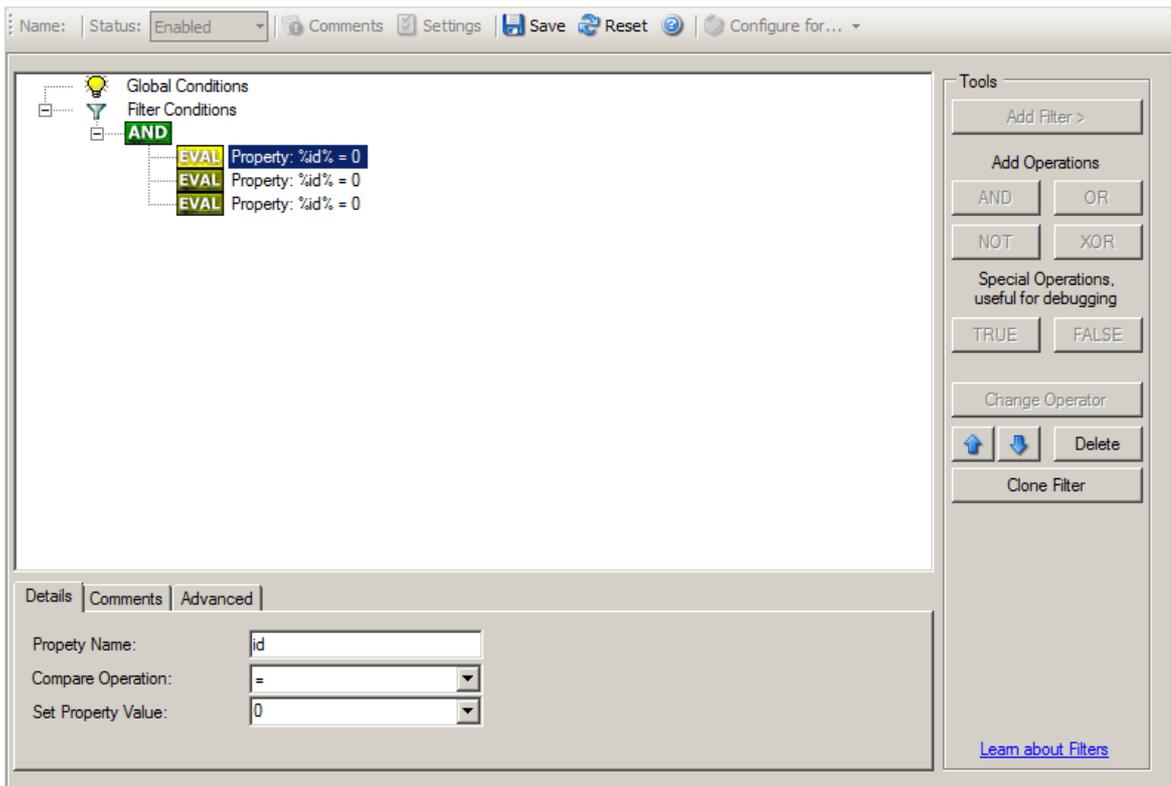
Ignoring Events - Figure 7

In order to enter the actual values, select each of the three filters. A small dialog opens at the bottom of the screen. There you enter the values you are interested in. In our sample, these are IDs 105, 108 and 118. As we are only interested in exactly these values, we do a comparison for equality, not one of the other supported comparison modes. When you have made the updates, you screen should look as follows:
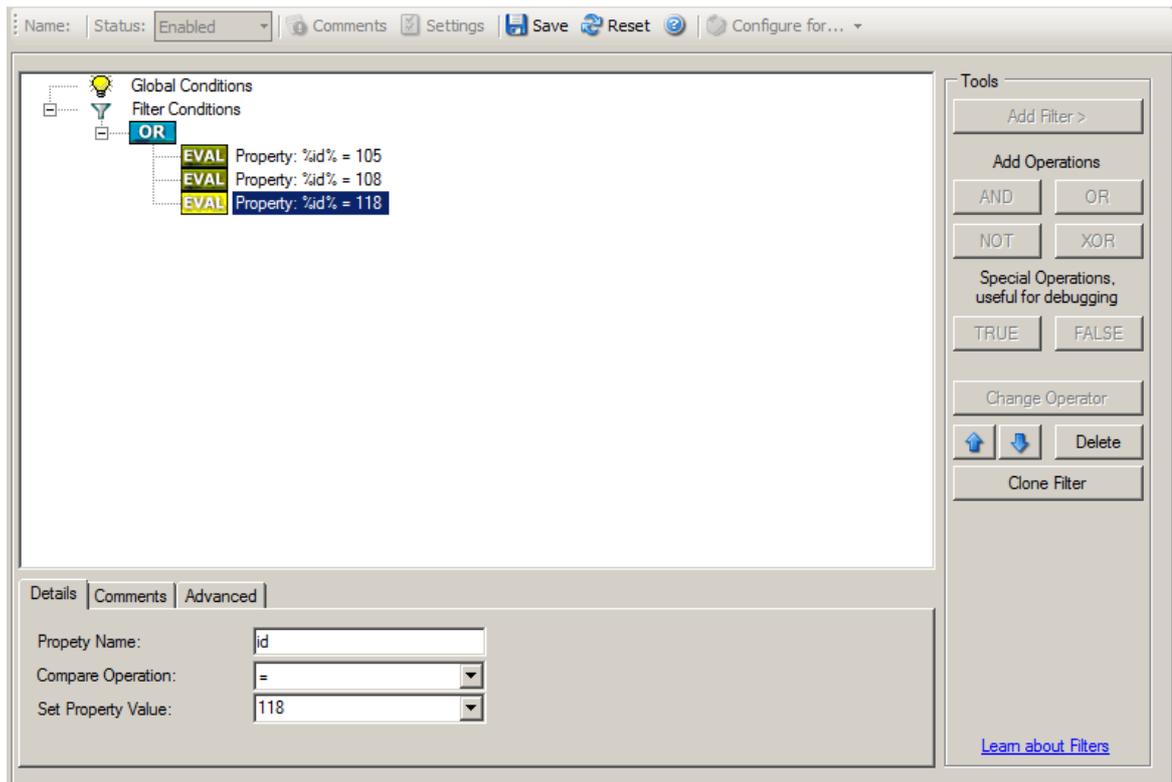
Ignoring Events - Figure 8

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

Now let us go ahead and actually discard these events. This is done via an action. To do so, right-click on "Actions" and select "Discard."
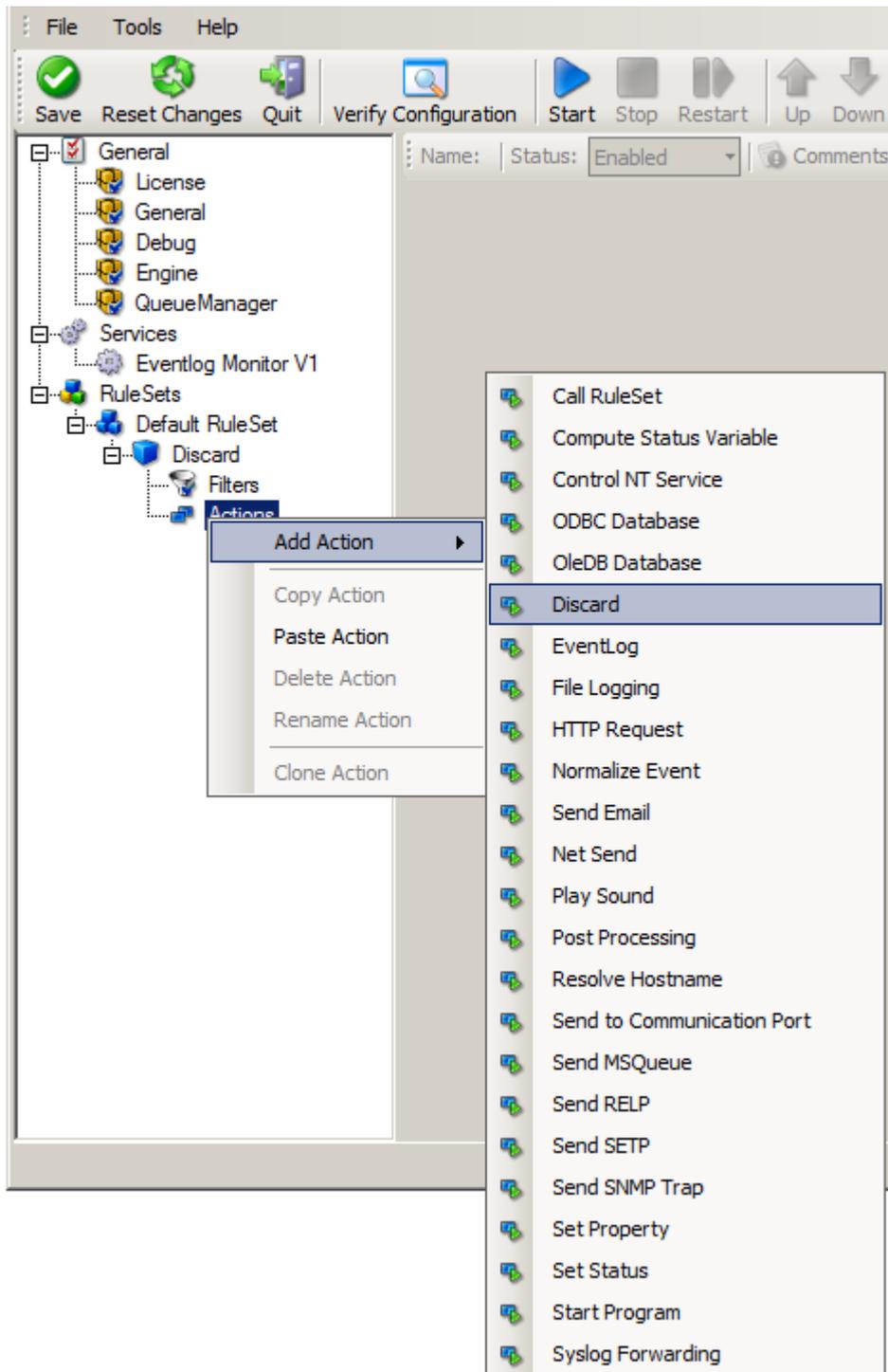
Ignoring Events - Figure 9

Again, name the action as you like in the following dialog. We use "Discard" as this is quite descriptive. Select "Next" and then "Finish" on the next page. Your screen should like follows:
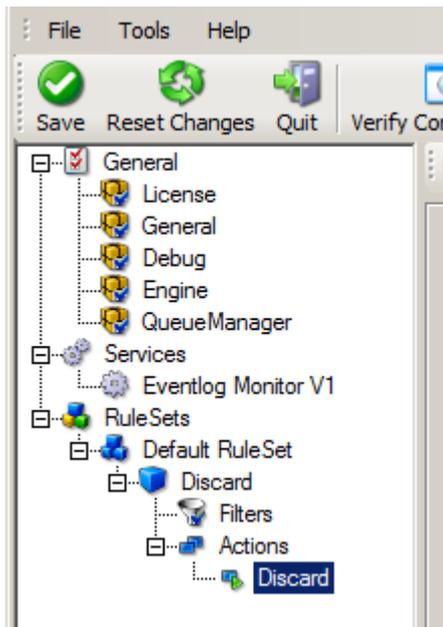
Ignoring Events - Figure 10

This concludes the definition of our first rule.

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

2.4.3   Logging Events

Range of events need to be stored persistently for later review and analysis. As such, we are in a need of a rule that persists the events. In our sample, we choose to work with a text log file (not a database, which we also could use). We will now create a rule to store all those events not discarded by the previous rule into a log file.

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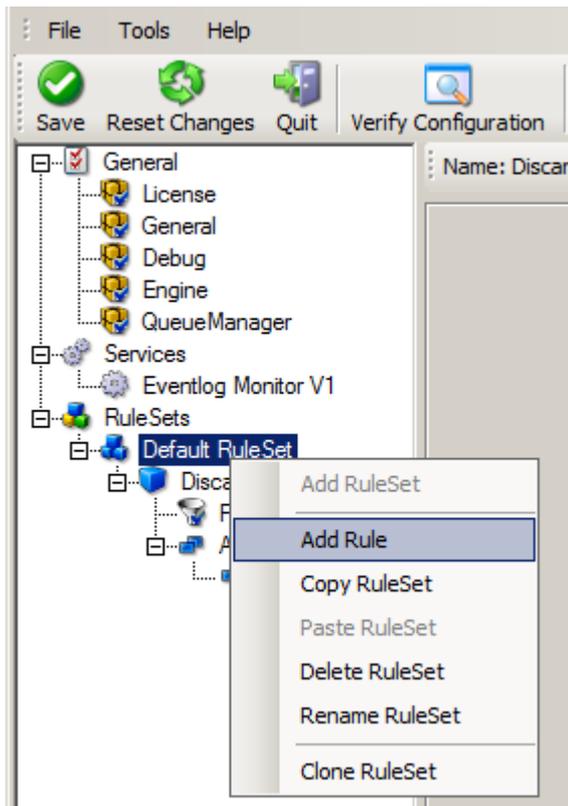
Logging Events - Figure 1

Use a name of your choice. 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 the help of this rule, we do not require any filter rules here. However, a corresponding action must be defined. Therefore, we just need to define the action:

To do so, expand "Write To file" and right-click "Actions". Select "Add Action", then "Write to File" as can be seen below:
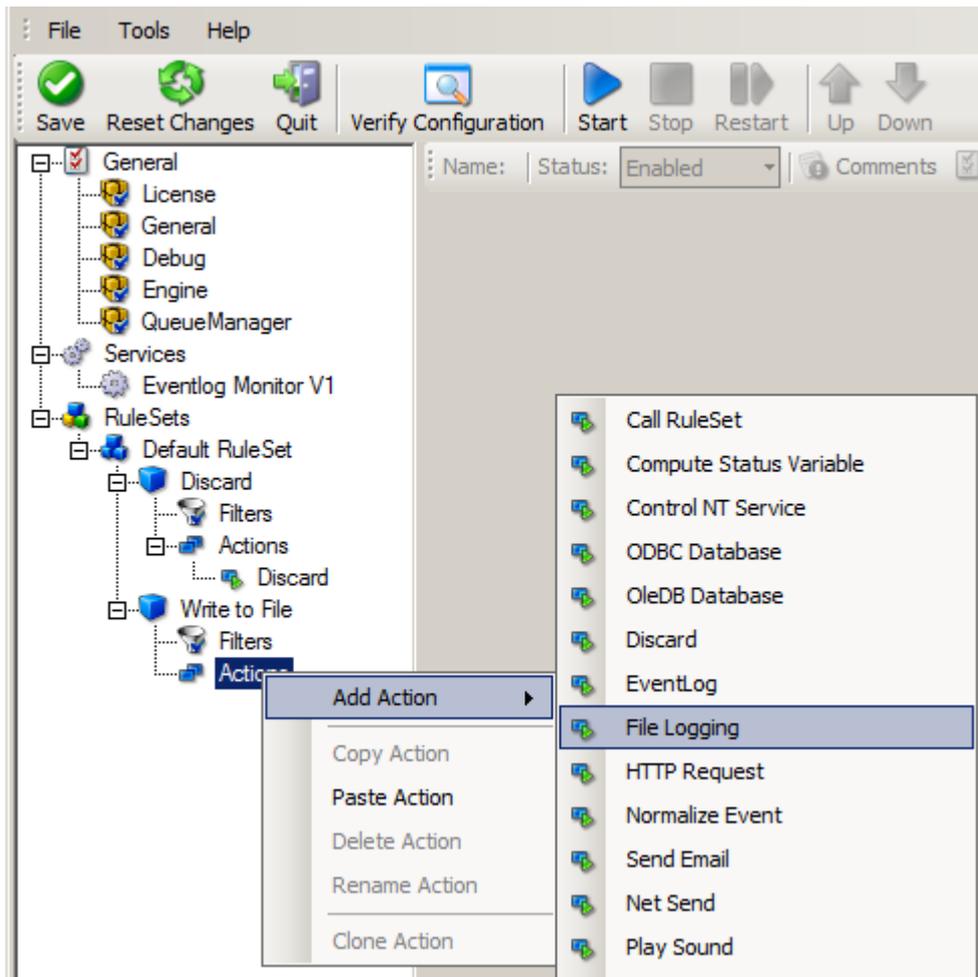
Logging Events - Figure 2

Again, choose a name. Do not modify the defaults. In our sample, we call this action "Records". Click "Next", then "Finish." Now the tree view contains a node "Records", which we select:
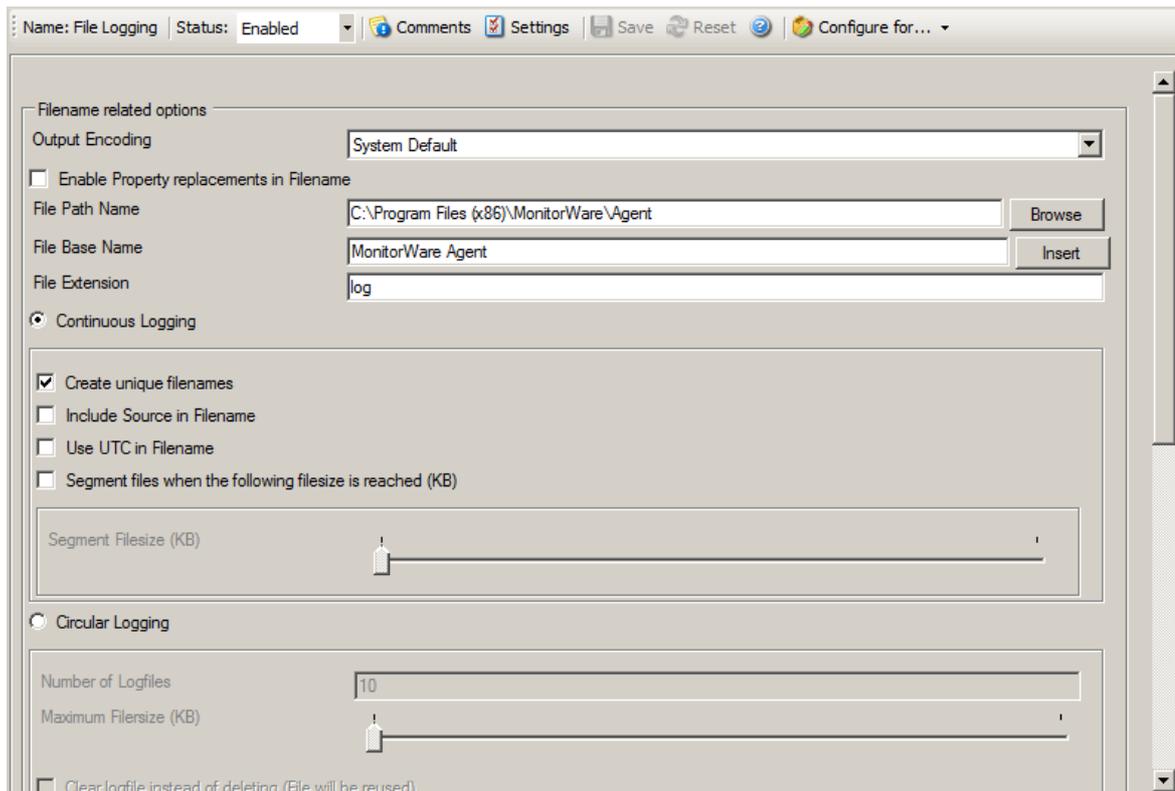
Logging Events - Figure 3

Important

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

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

Summary

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

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

Logging Events - Figure 4

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

Please note that the current date is appended to the log file. This facilitates file management in archiving. The format is "logdata-YYYY-MM-DD.log".

You have now learned to define rules and actions. The following chapters thus will not cover all details of this process. If in doubt, refer back to these chapters here.

2.4.4   Time-Based Filters

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

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

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

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

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

Please note: we use the 12-hour clock system.
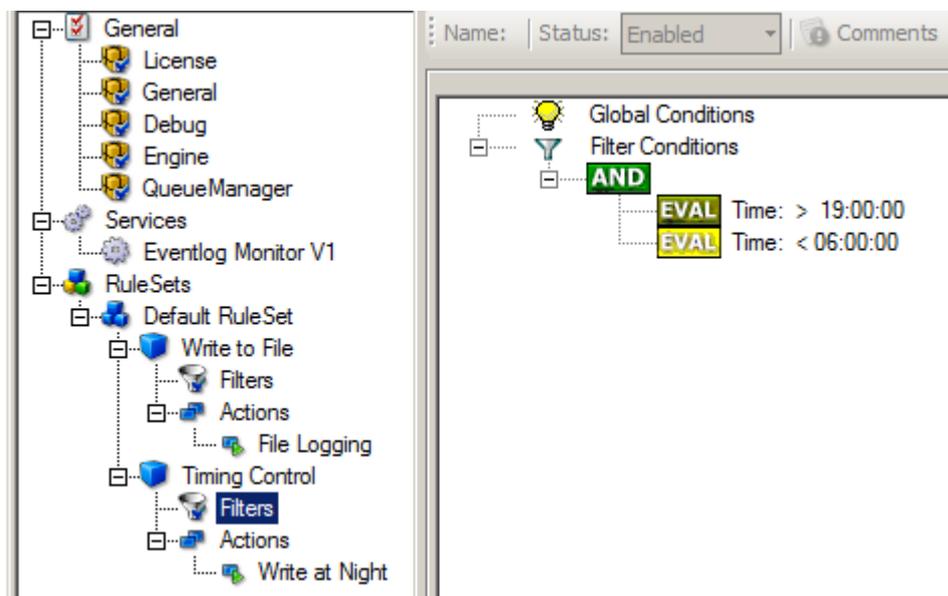
Time-Based Filters - Figure 1

All events generated by services binding to our rule set "Defaults" will now also be passed along the "Timing Control" rule set. If these events come in night times between 07:00:01 PM and 5:59:50 AM, the action "Write at Night" is executed.

Please note that the use of the "OR" operator is important because either one of the time frames specified does apply. This is due to the midnight break.

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

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

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

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

As stated earlier, time frames are most often used in combination with other filters. Here is a more complex example:
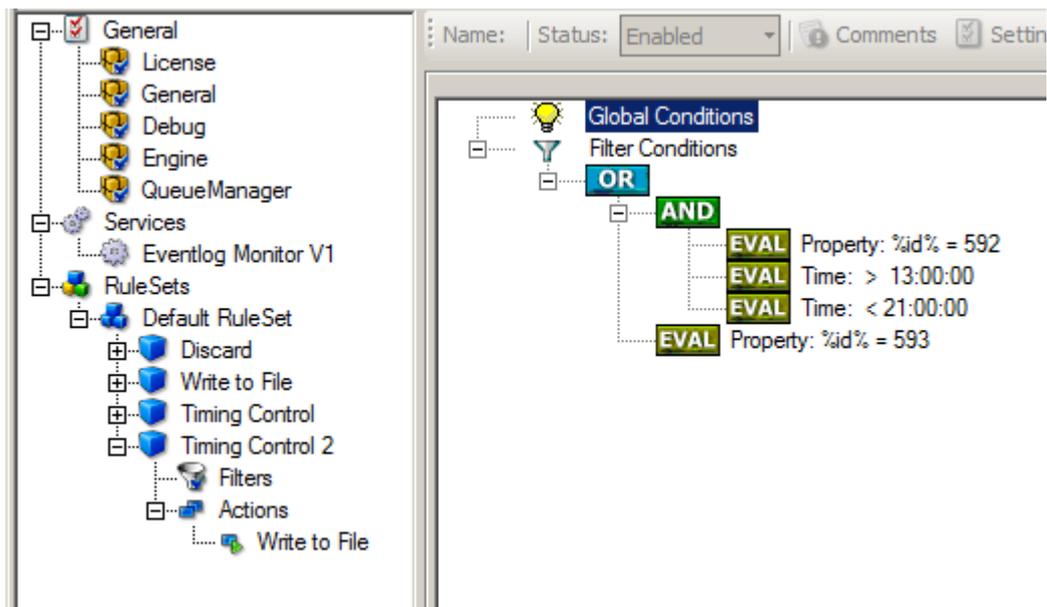
Time-Based Filters - Figure 2

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

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

Now let us look at some sample data:

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

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

In all, the filter condition is false.

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

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

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

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

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

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

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

2.4.5 Email Notifications

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

Here, we create an additional rule for that purpose: Right-click the "Defaults" rule set and select "Rule Sets", "Add Rule" from the pop up menu. Provide a name. We will call it "Email Reception" in this example. Then, add a "Forward via Email" action. In the action details, be sure to configure at least the mail server, recipient and subject properties.

Please note that many mail servers also need a valid sender mail address or otherwise deny delivery of the message.
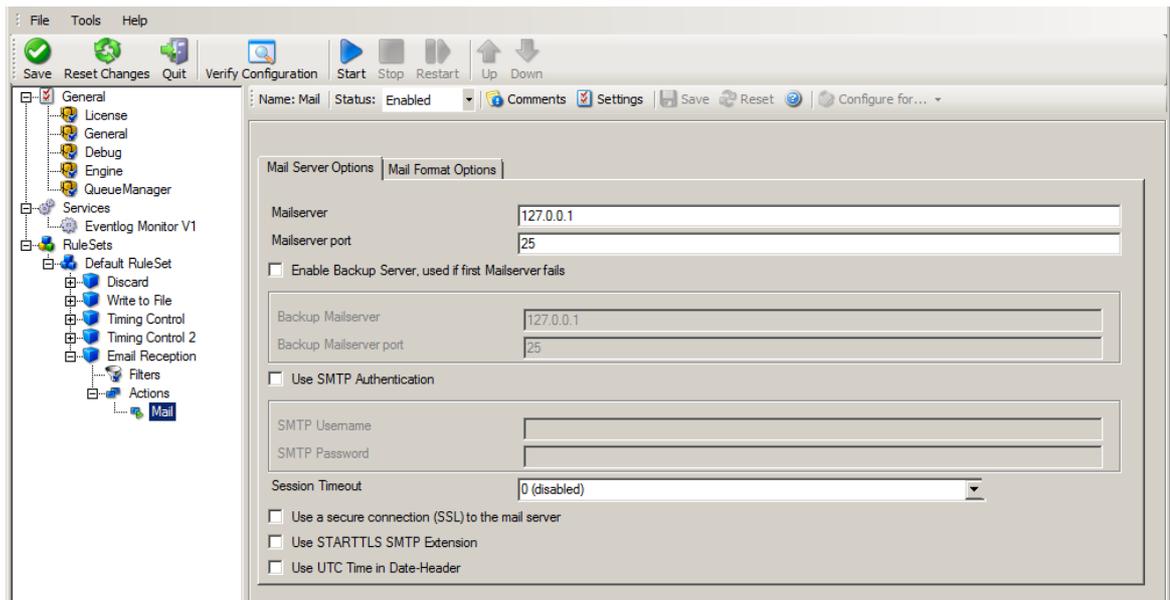
Email Notifications - Figure 1

Then, select the filter conditions. Let us assume we are just interested in events of ID 600. Then the filter conditions should look as can be seen below:
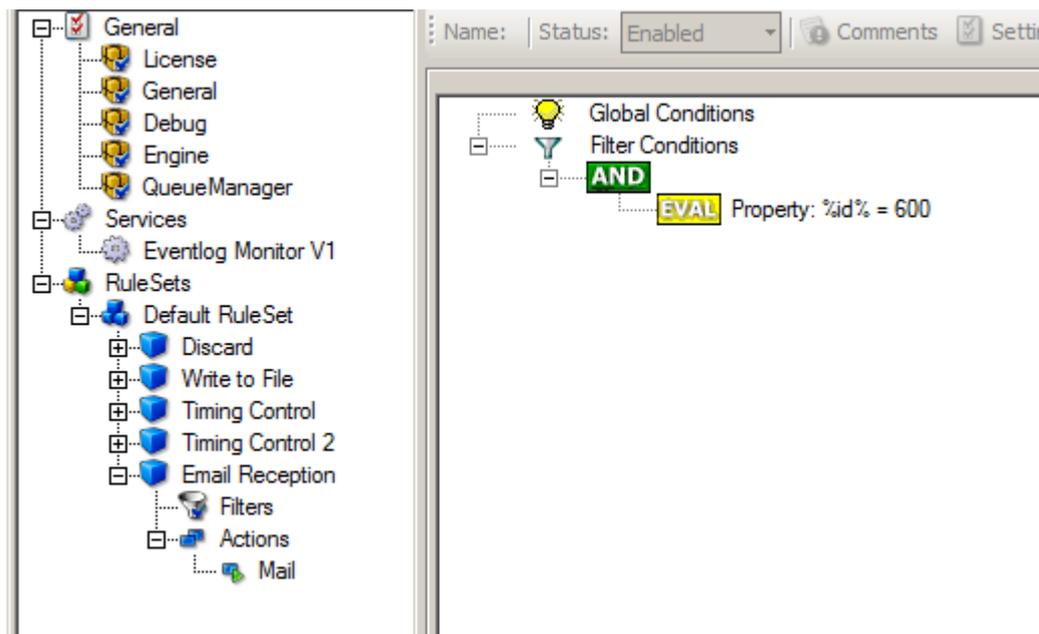
Email Notifications - Figure 2

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

2.4.6   Alarming via Net Send

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

Please bear in mind that the Windows messenger service is not the instant messaging service that many people nowadays associate with it. The messenger service is meant for administrator notifications. If a Windows workstation (or server) receives a message via messenger 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 else you will become overwhelmed with the messaging windows. A better example could be a filter that checks for a server running low on disk space (using the disk space monitor).
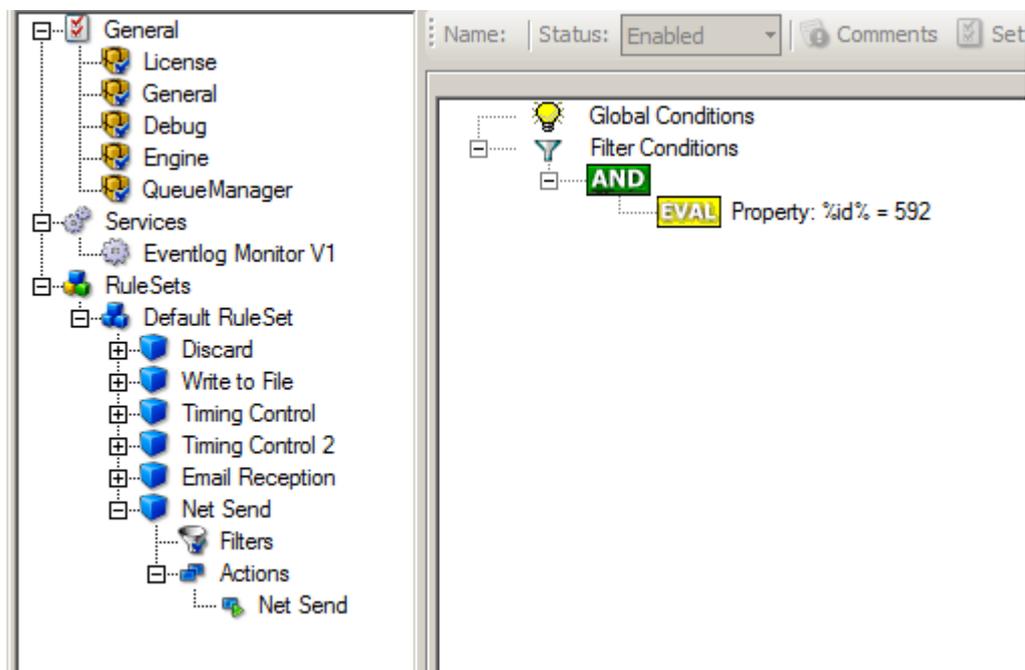
Alarming via Net Send - Figure 1

This time, we use the "Net Send" action as can be seen below. The target field holds either the name or IP-Address of the workstation this message should be going to. The message text itself goes into "Message to send" field.
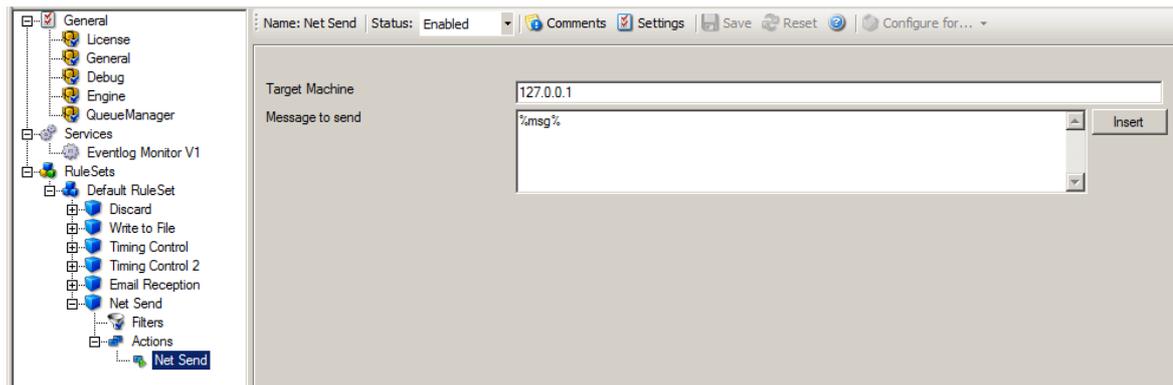
Alarming via Net Send - Figure 2

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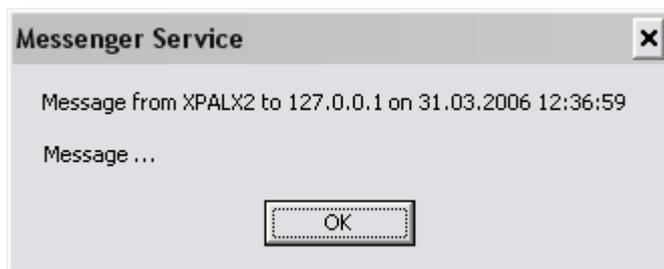
Alarming via Net Send - Figure 3

2.4.7   Starting Scripts and Applications in Response to an Event

We can start an application or a script on the occurence of certain events. This is done to start administrative scripts to perform corrective actions. For example, if a disk runs low on space, you could start a script that deletes temporary files, or if a service fails, a script could restart it.

Our sample, on the other hand, is kept quite simple again. We just show how to generically start an exe file. To do so, we define a new rule, name "Application starts" below. Again, we use the imaginary event 592 as a filter condition. Therefore, the application will start whenever event 592 comes in.
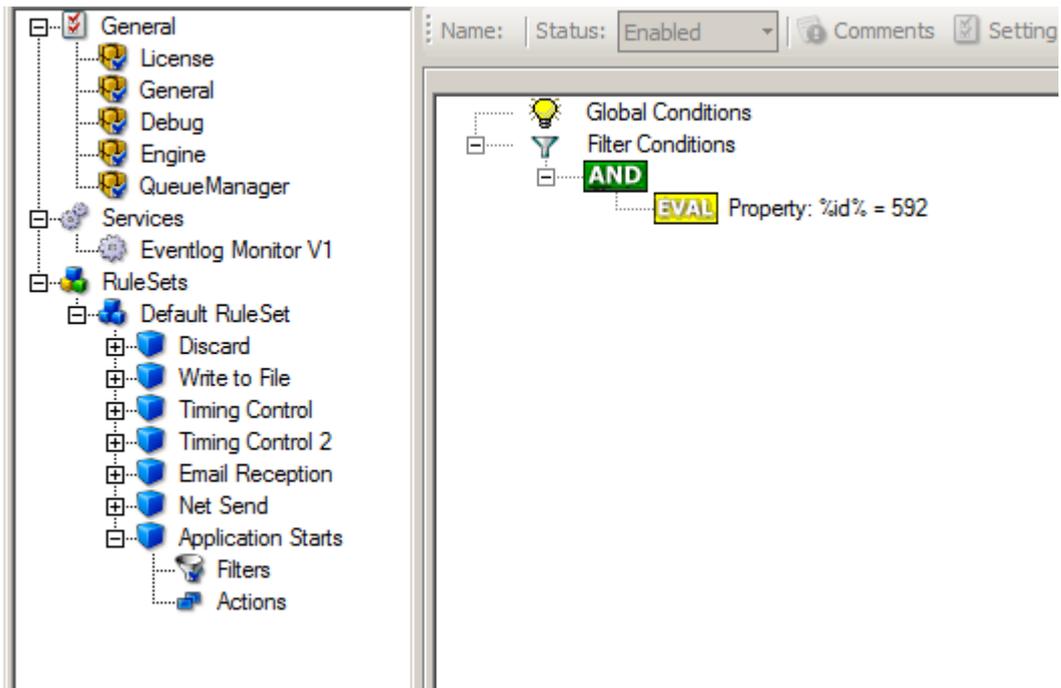
Starting Scripts and Applications in Response to an Event - Figure 1

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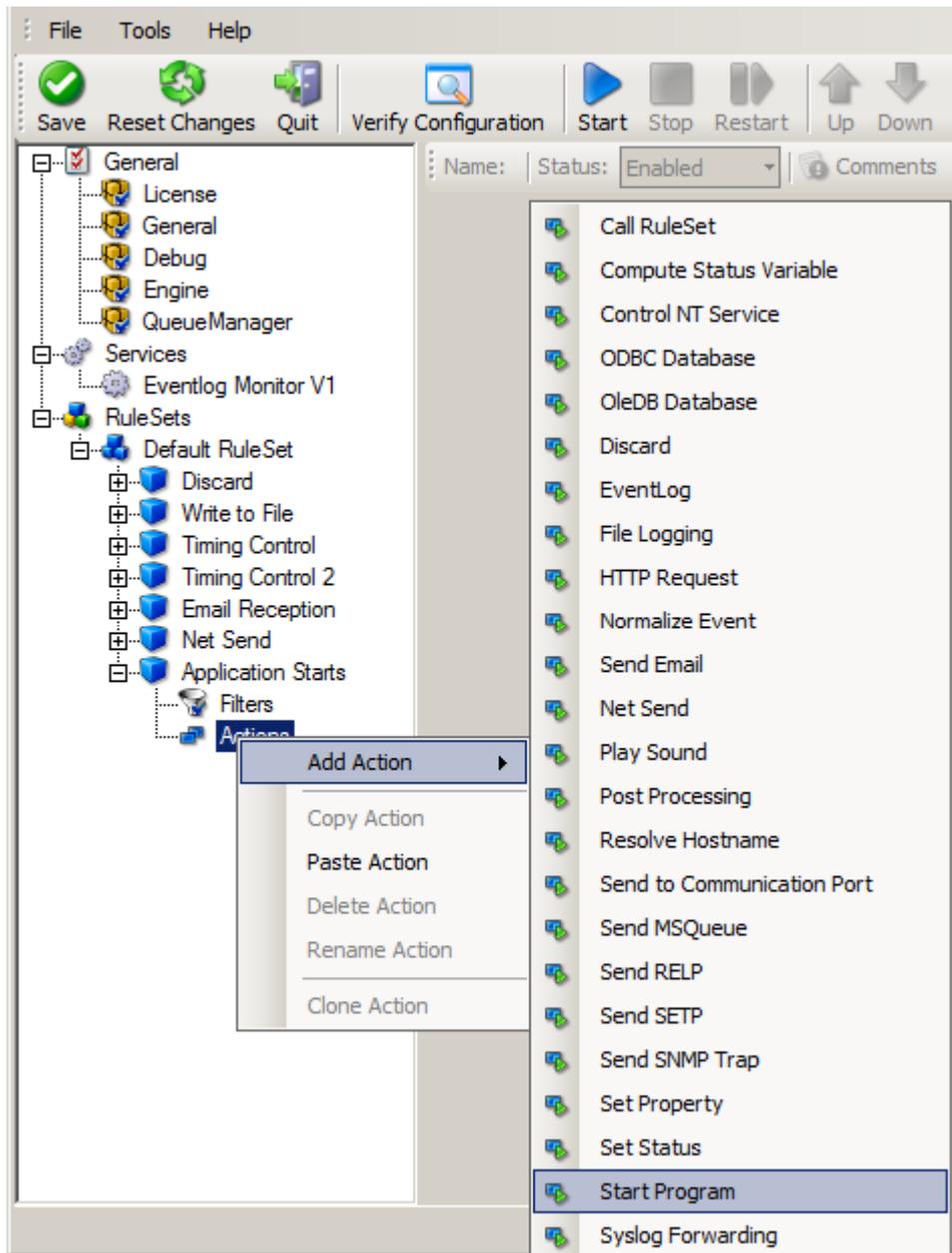
Starting Scripts and Applications in Response to an Event - Figure 2

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

| Name: Start Program | Status: | Enabled | ▼ | �

 Comments | ☒ Settings | 💾 Save | 🔄 Reset | ? | 🔘 Configure for... ▼ |

Command to execute

C:\Program Files\Alivemon\Alivemon.exe   Browse

☑ Use legacy parameter processing

Command Parameters

/p   Insert

◉ Synchronus Processing (Wait for Completion)

Sync Timeout

0 (disabled)   ▼   milliseconds
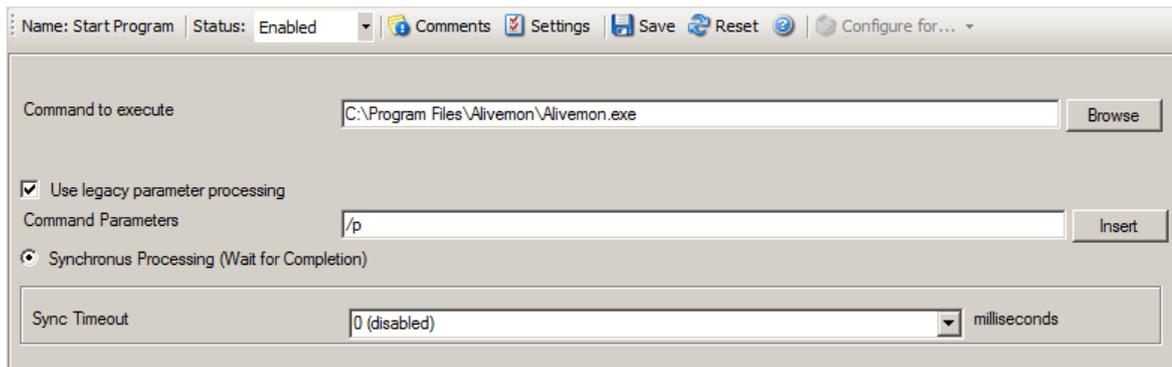
Starting Scripts and Applications in Response to an Event - Figure 3

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

3   Step-by-Step Guides

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

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

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

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

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

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

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

Installations and Configurations

- How do I export the configuration and create a debug file?
- How do I enter the license information from the product delivery email?
- Forwarding filtered IIS Logfiles
- Database Logging with MSSQL
- How to apply filters to only get interactive logon/logoff events?
- How do I apply filters in MonitorWare Agent, WinSyslog and EventReporter?
- How To Setup MonitorWare Agent/ WinSyslog/ EventReporter
- Configuring Windows for the Event Log Monitor

- Intrusion detection via the Windows event log

Services

- How To setup the EventLogMonitor Service
- How To setup the EventLogMonitor V2 Service
- Forwarding Windows event logs to a Syslog server
- Forwarding Windows event logs to an SETP server

Actions

- How To setup the Forward via Syslog Action
- How To setup an SETP Action
- How To setup a Write to File Action
- How To setup the Forward via EMail Action
- How To setup the Set Property Action
- How To setup the Set Status Action
- How To setup the Start Program Action
- How To setup the Control Windows Services Action
- How To Create a Rule Set for Database Logging
- How to store custom properties of a log message in a database

Centralized Monitoring / Reporting

- How To setup Windows centralized Monitoring (EventReporter 9.x & WinSyslog 8.x)
- How To setup a central log server for Windows machines and syslog sending devices (MonitorWare Agent 4.x & EventReporter 8.x)
- How to setup Windows centralized Monitoring (Common)
- How To setup Windows centralized Monitoring (EventReporter 8.x, WinSyslog 7.x and Monilog 2.x)
- How To Report Log Truncation

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

4 Configuring EventReporter

EventReporter is easy to use and is powerful.

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

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

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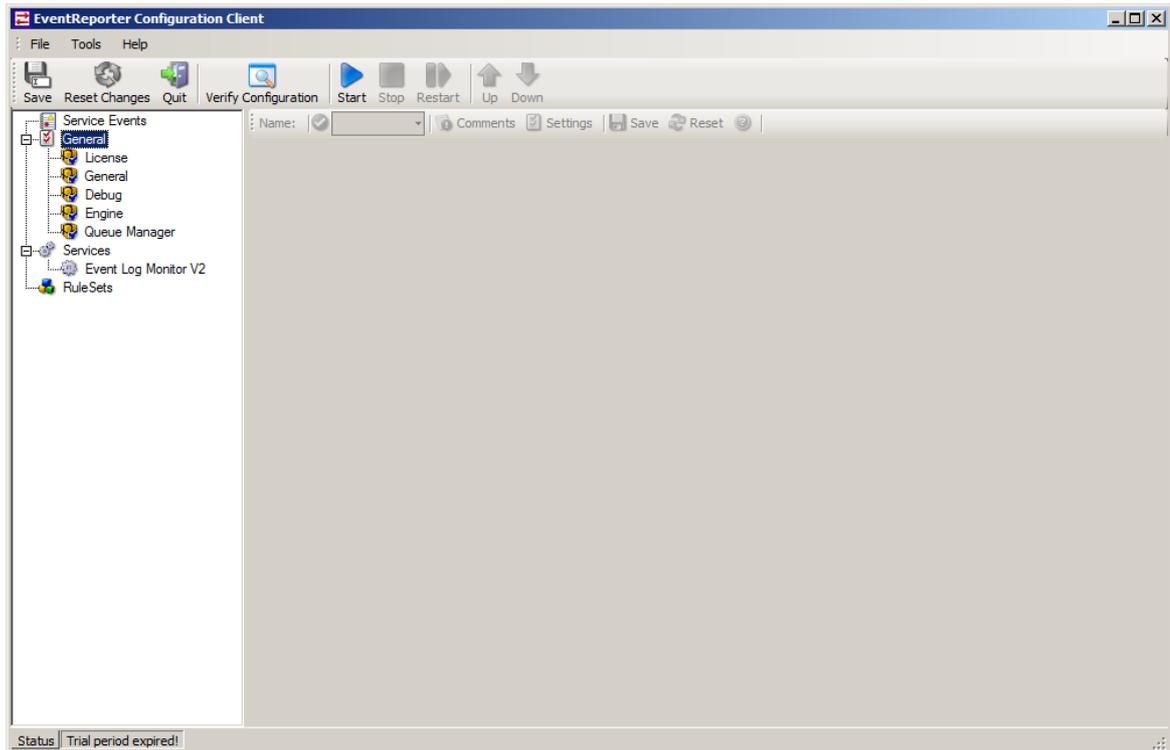
EventReporter Configuration Client

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

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

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

The tree view's **Running Services** area lists all configured services as well as their parameters. There is exactly one service entry for each service created. Please note that there can be as many instances of a specific service type as your application requires. In the above example, there are two instances of the Syslog Server, each one listening to a separate port. Theoretically, you can run a few hundred services in a single service instance. However, both from a usage scenario point of view as well as concerning operating system resources, we recommend limiting the services to a maximum of 20 to 30. Of course, there are some applications where more than this limit is useful. EventReporter does not restrict this number. If there is a need for a large number of services and the hardware is capable of managing all these tasks, there is nothing in the EventReporter that limits from doing so.

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

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

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

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

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

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

The following sections describe each element's properties.

4.1   Client Options

There are several options, that refer to the configuration client and not the service. These can be found under File -> Options
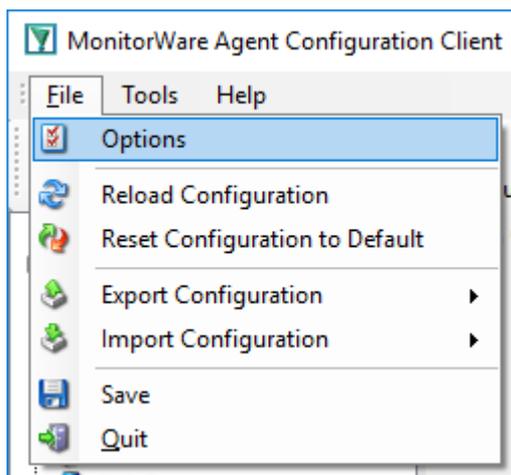
General Options
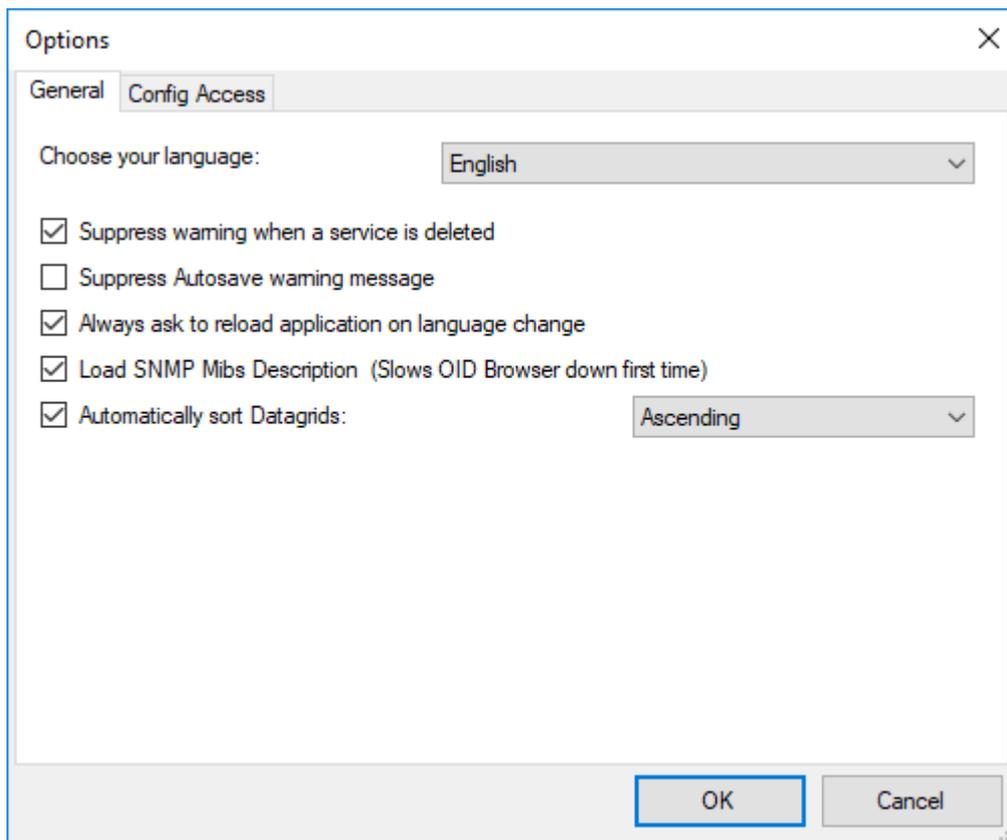
Choose your language

You can choose from various language packs, devlivered with the client. Please note, that some languages are not fully supported and "English" is the default and suggested language.

Suppress warning when a service is deleted

If this iption is checked, warnings when deleting a service will be suppressed. Such a warning can occur when you try deleting a service and there is no other service using the connected ruleset.

Show autosave warning message

If you make changes in the configuration and switch to another component, a warning will occur if you haven't saved the changes. This warning will also allow you to directly enable auto-saving the configuration.

Always ask to reload application after language change

When you change the language, a popup will ask you to reload the configuration client to properly apply the changes and load with the set language.

Automatically sort Datagrids

Datagrids are used in certain areas within the configuration objects. You can change the default sorting behavior from ascending to descending here.

Config Access

Load Configuration Registry Path

The Configuration Client can be switched to a different registry path for configuration. The registry path change can be made permanent here. The changed registry path is the saved within the Parameters key of the Service.

Load Configuration from File

Alternatively, you can configure the service to load the configuration from a file. You can set the paths with the two fields below.

When enabled, the configuration will always be backed up before applying the new configuration. The backup consists of the last iteration and will be placed in the same directory.

Create individual configuration files for Services

Can only be enabled when "Load Configuration from File" is enabled. When enabled, the Services section of the configuration will be put into a separate file.

Create individual configuration files for RuleSets

Can only be enabled when "Load Configuration from File" is enabled. When enabled, the RuleSet section of the configuration will be put into a separate file.

4.2   Client Tools

There are several tools within the configuration client that you can use to test certain services or debug the application in general. Some can be found under in the Tools menu in the.
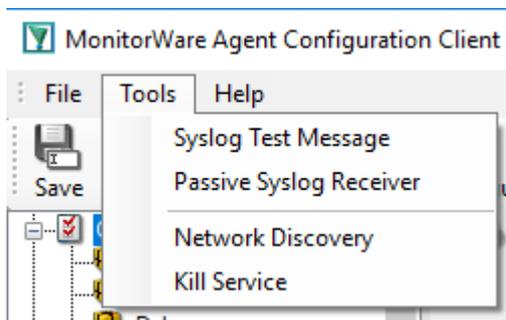
Syslog Test Message

Opens a new windows which can send syslog test messages to Syslog Servers. This can also be opened within the configuration window of a Syslog service.

Syslog Server
The hostname or ip address of the target syslog server.

Syslog Port
The port that should be used to connect to the target syslog server.

Network Protocol
Which network protocol should be used, either UDP or TCP can be selected.

Message Delimiter
When using TCP protocol, a message delimiter (separator) can be configured which is a simple linefeed by default.

Repeat Message
How often you want to repeat the test message. Can be configured from 1 to 1000.

Append Number to Syslog Message
If sending multiple messages, enable this option in order to add a syslog number at the end of the message.

Sleeptime between sending

When using TCP, you can use 0ms. For UDP we recommend 1-5ms as sleeptime between sending syslog messages. Otherwise package loss can happen.

Load RAW Syslogdata from File

You can choose to load raw syslogdata from file using this option. When loading UTF8 data make sure to set the Output encoding format from ASCII to **UTF8**. And if your file contains multiple syslog messages make sure that **Send One Message per LineFeed** is checked.

Configure Syslog message with these properties

Choose this if you want to configure all properties of the syslog message manually.

Send one Message per LineFeed

Check if your syslogdata contains multiple syslog messages divided by linefeeds

Output encoding

Select the Output encoding you wish to use. When using UTF8, the UTF8 BOM is automatically prepended.

Passive Syslog Receiver

Opens a new windows to test Passive Syslog Servers. This can also be opened within the configuration window of a Passive Syslog service.

Syslog Server

The hostname or ip address of the target passive syslog server.

Syslog Port

The port that should be used to connect to the target passive syslog server.

Message Delimiter

The message delimiter (separator) used to split syslog messages which is a simple linefeed by default.

Send this Message after Connect

If required, configure a custom message that is send to the server after connect.

Expect this Message after Connect

If required, configure a custom message that is expected by the sender when the server response to our custom message.

Network Discovery

Opens up a Wizard that will help you discover devices in your local network. Once the wizard has scanned your network, it will show Windows compatible devices it has found. Please note that this

will require Windows Management Instrumentation (WMI) access to the remote machines which may be disabled in Windows Firewalls by default.
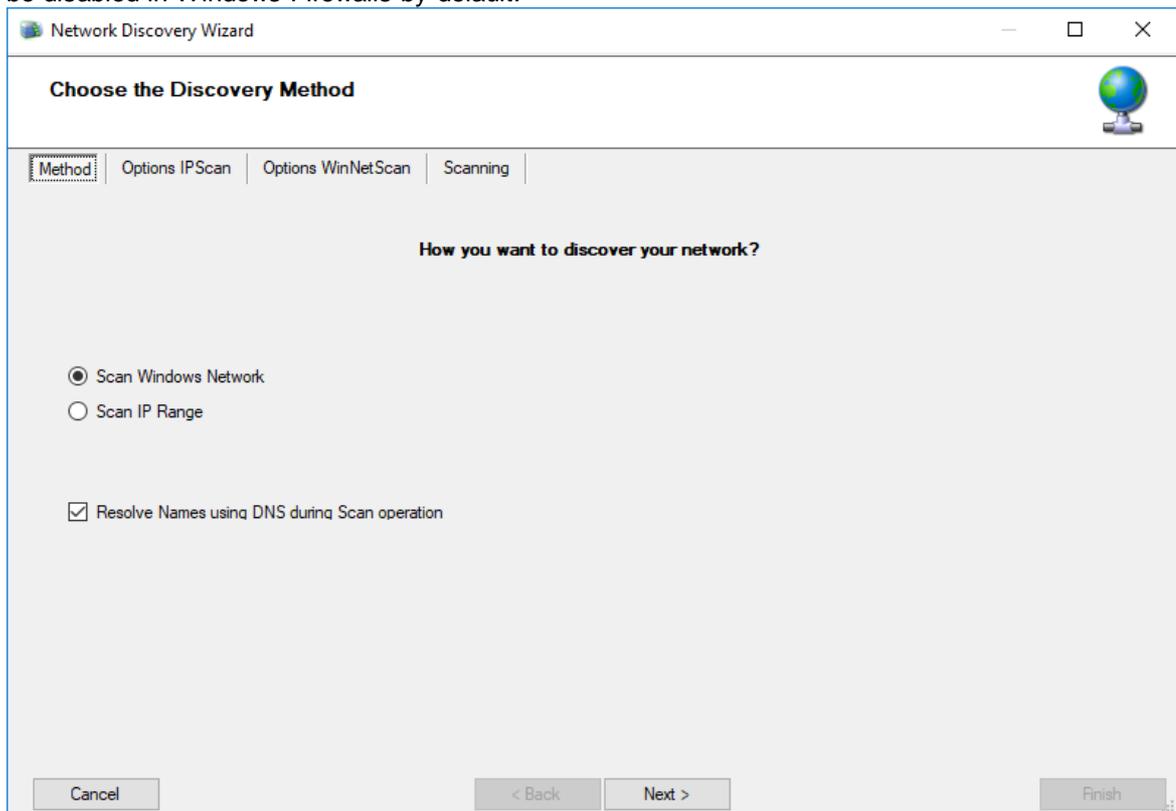
Kill Service

When stopping a service, and it doesn't shutdown in the time period, you can use this function to forcefully stop the service. The service process will be killed if possible.

DebugLog

The **DebugLog** Button will be available if Debug Logging is enabled in your Debug Options
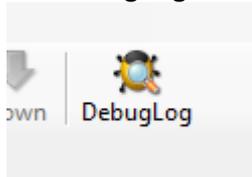
When clicked, a new Logviewer window will be opened. The Debug Logviewer can load, parse and analyze debug log-files from the service.
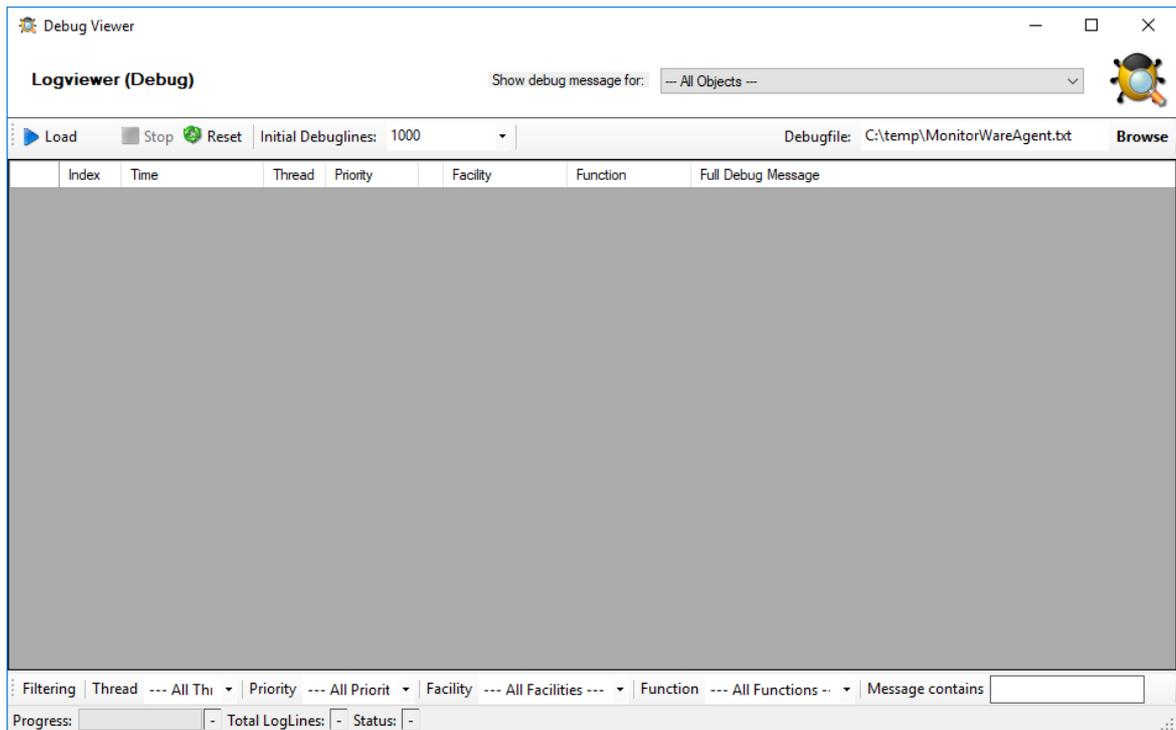
Debugfile

Will automatically be set to your configured debug file. You can also choose other saved debug-files for analysis.

Load

When Load is clicked, the Logviewer will load lines as configured in the initial debug-lines field. When loading all log-lines on a large debug log-file, this may take a while. While the Load button is grayed out, the Logviewer will continue to read data from the debug log as it is being written.

Stop

Stop continuous loading of the debug log.

Reset

Will reset all loaded log-lines from memory and clear the debug data-grid.

Init Debuglines

The amount of log-lines you want to read the first time.

Show debug messages for

Once the debug-log is processed, the Logviewer will automatically add filters for objects like services, rulesets, rules and actions. You can use this select box to filter by them.

Filtering (bottom bar)

At the bottom of the Logviewer window, you can filter the debug-log for Thread (ID), Priority, internal Facility and Functions. You can also filter for words or word sequences. The view will automatically be refreshed once you changed a filter.

4.3   Using File based configuration

Working with File based Configurations

Support for running the Service from file based configuration may be interesting for environments where you want to minimize registry access to a minimum or you want to manually edit the configuration without using the configuration client every time.

The Adiscon Configuration format is quiet simple. In the following description, all the configuration options will be explained in detail.

Adiscon Configuration format explained

Our configuration format is something between JSON and XML but hold at a very simple level.

Variables

All variables start with a dollar ($). Name and Value of a variable are separated by the FIRST space character. Everything else behind the first space will be considered as the Value. A linefeed terminates the value. If your configuration value contains has linefeeds, you have to replace them with "\\n" or "\\r\\n". A single backslash can be used to escape a brackets ({ and }).

Comments

All lines starting with a sharp (#) at the beginning will be ignored.

File Includes

Sample

includeconfig my-subconfigfiles-*.cfg

The includeconfig statement will include either a single file or many files based on a filename pattern. In this sample all Files starting with "my-subconfigfiles-" and ending with ".cfg" will be included into the configuration. It is possible to create your own custom file structure with includes. The configuration client will be able to load and show your custom file structure, however it will not be able to maintain (save) it. We support a maximum include depth of up to 10 levels when using the includeconfig statement.

General Options

Sample

general(name="[name]") {
        $nOption 1

```
        ...
}
```

All options between the brackets will be loaded as variables into the general configuration object. The name attribute field specifies the general configuration block name. The brackets start and end an object block.

Services

All possible configuration parameters are named within the detailed services documentation.

Sample Service configuration:

```
input(type="[ID]" name="[name]") {
        $var1 Value1
        $var2 Value2
        ...
}
```

The brackets start and end a service block. All variables between the brackets will be loaded into the service configuration. The name attribute specifies the service display name. The type attribute contains the service type ID. It can be one of the following types:

1	= Syslog
2	= Heartbeat
3	= EventLog Monitor V1 (Win 2000 / XP / 2003)
4	= SNMP Trap Listener
5	= File Monitor
8	= Ping Probe
9	= Port Probe
10	= NTService Monitor
11	= Diskspace Monitor
12	= Database Monitor
13	= Serialport Monitor
14	= CPU Monitor
16	= MonitorWare Echo Request
17	= SMTP Probe
18	= FTP Probe
19	= POP3 Probe
20	= IMAP Probe
21	= IMAP Probe
22	= NNTP Probe
23	= EventLog Monitor V2 (Win VISTA/7/2008 or higher)
24	= SMTP Listener
25	= SNMP Monitor
26	= RELP Listener
27	= Passive Syslog Listener

1999998= MonitorWare Echo Reply
1999999= SETP Listener

RuleSets

All possible configuration parameters are named within the detailed actions documentation.

Sample

```
ruleset(name="[name]" expanded="[on/off]") {
        rule(name="[name]" expanded="[on/off]" actionexpanded="[on/off]"
ThreatNotFoundFilters="[on/off]" GlobalCondProperty="[on/off]" GlobalCondPropertyString=""
ProcessRuleMode="[0/1/2]" ProcessRuleDate="[uxtimestamp]") {

                action(type="[ID]" name="[name]") {
                        $var1 Value1
                        $var2 Value2
                        ...
                }
                filter(nTabSelection="0") {
                        $nOperationType AND
                        $PropertyType NOTNEEDED
                        $PropertyValueType NOTNEEDED
                        $CompareOperation EQUAL
                        $nOptionalValue 0
                        $nSaveIntoProperty 0
                        $szSaveIntoPropertyName FilterMatch
                }
        }
}
```

The brackets start and end a ruleset block. The attributes of a Ruleset are self-explainable. Within a RuleSet, you can have Rules. The attributes of Rules are also self-explainable and partially Global Conditions that are equal to the options found in the Filter dialog. Within a Rule you can one Basefilter. This Basefilter again can have child filters it it, and these child filters can have child filters again. All "expanded" settings are optional and only important for the client treeview.

Within a Rule you can have Actions. The brackets start and end an action block. All variables in an action block between the brackets will be loaded into the action configuration. The name attribute specifies the service display name. The type attribute contains the action type ID. It can be one of the following types:

1000 = ODBC Database
1001 = Send Syslog
1008 = Net Send
1009 = Start Program
1011 = Send SETP
1012 = Set Property

1013 = Set Status
1014 = Call RuleSet
1015 = Post Process
1016 = Play Sound
1017 = Send to Communication Port
1021 = Send SNMP
1022 = Control NT Service
1023 = Compute Status Variable
1024 = HTTP Request
1025 = OleDB Database
1026 = Resolve Hostname
1027 = Send RELP
1028 = Send MS Queue
1029 = Normalize Event
1030 = Syslog Queue

How to enable file based configuration?

To switch from registry to file configuration mode, all you need to do is go the "Config Access" tab in the Configuration "Client Options" and switch from "Load Configuration from Registry" to "Load Configuration from File" mode. Once you accept the change, the Client will ask you if you want to export the current loaded configuration into the file. Hit **YES** if you want to do so, and **NO** if already have an existing configuration file. The configuration client will reload itself automatically after this.
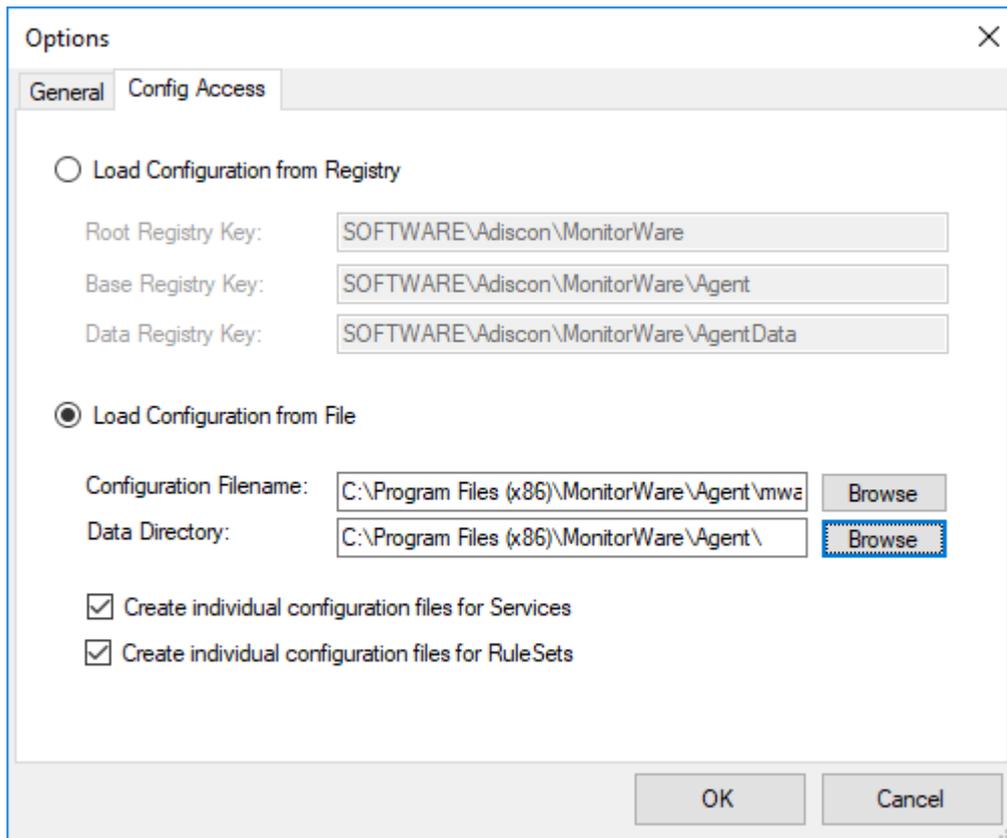
Screenshot from Client Options to configure file based configuration.

Create individual configuration files for Services

When enabled, the configuration client will create separated configuration files for each configured service. The main configuration file will then use the includeconfig statement to include all these configuration files by using a pattern. When deleting a service, its configuration file will be deleted as well.

Create individual configuration files for RuleSets

When enabled, the configuration client will create separated configuration files for each configured ruleset. The main configuration file will then use the includeconfig statement to include all these configuration files by using a pattern. When deleting a ruleset, its configuration file will be deleted as well.

4.4   General Options

4.4.1   License Options

License Options Tab

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

License Option Parameters

Registration Name

File Configuration fields:	szLicense
Description:	The user chooses the registration name. It should correspond to your organization name, e.g. a company called "AA Carpenters, Inc." should not choose "AA" as registration name. This can easily be mistaken and most probably 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

File Configuration fields:	nLicenseKey1, nLicenseKey2, nLicenseKey3, nLicenseKey4, nLicenseKey5
Description:	Adiscon provides this number. It is valid for a specific registration name. Be sure to enter the correct registration number. Each block of the license key must be filled into one of the key fields. Alternatively, you can use the "Import from Clipboard" button. The client detects invalid registration numbers and report the corresponding error.

4.4.2 General

General Options Tab

The General Options available on this form are explained below:

General Options

Process Priority

File Configuration fields:	nProcessPriority
Description:	Configurable Process Priority to fine-tune application behavior.

Queue Limit

File Configuration fields:	nQueueLimit
Description:	The applications keeps an in-memory buffer where events received but not yet processed are stored. This allows the product to handle large message bursts. During such burst, the event is received and placed in the in-memory queue. The processing of the queue (via rule sets) itself is de-coupled from the process of receiving. During traffic bursts, the queue size increases, causing additional memory to be allocated. At the end of the burst, the queue size decreases and the memory is freed again.

	Using the queue limit, you can limit that maximum number of events that can be in the queue at any given time. Once the limit is reached, no further enqueueing is possible. In this case, an old event must first be processed. In such situations, incoming events might be lost (depending on the rate they come in at). A high value for the queue size limit (e.g. 200,000) is recommended, because of the risk of message loss. It is also possible to place no limit on the queue. Use the value zero (0) for this case. In this case, the queue size is only limited by virtual memory available. However, we do not recommend this configuration as it might cause the product to use up all available system memory, which in turn could lead to a system failure.

SystemID

File Configuration fields:	nSystemID
Description:	SystemID is of type integer to be used by our customer. In addition, it is user configurable.

CustomerID

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

Location of your MIBS

File Configuration fields:	szMIBSPath
Description:	Click the Browse button to search for your MIBS location or enter the path manually. The Client and Service will read all files from this directory automatically on startup.

Automatically reload service on configuration changes

File Configuration fields:	nEnableAutoConfigReload
Description:	When enabled (default), the service will detect configuration changes and reload it's core automatically. This feature only works if the latest Client Application is used for configuration. It will also work if you are using the file based configuration method and update the configuration file. It will not work if you are using the service in console mode unless you send any input to the console.

Enable random wait time delay when checking for new configurations / Maximum random delay time

File Configuration fields:	bAutoReloadRandomDelay, nAutoReloadDelayTime
Description:	When enabled, a random delay (with the configured maximum) will be added between new configuration checks. The maximum for this random delay is 24 hours. The random delay has no affect on the service control anymore.

4.4.3 Debug

Debug Options Tab

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

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

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

Debug Options

Enable Debug output into file

File Configuration fields:	nEnableDebugOutput
Description:	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

File Configuration fields:	szDebugFileName
Description:	The full name of the log files to be written. Please be sure to specify a full path name including the driver letter. If just the file and/or path name is specified, that information is local to the service default directory. As this depends on a number of parameters, it might be hard to find the actual log file. So for consistency purposes, be sure to specify a fully qualified file name including the drive. Note: If the configured directories are missing, they are automatically created by application i.e. the folder specified in "File and Path Name".

Debug Levels

File Configuration fields:	nDebugErrors, nDebugInternal, nDebugMini, nDebugRuleEngine, nDebugUltra
Description:	These checkboxes control the amount of debug information being written. We highly recommend only selecting "Errors & Warnings" as well as "Minimum Debug Output" unless otherwise instructed by Adiscon support.

Circular Debug Logging

File Configuration fields:	nCircularLogging
Description:	Support for circular Debuglogging has been added as the debuglog can increase and increase over time. This will avoid an accidental overload of the harddisk. Of course you can also customize the amount of files used and their size or disable this feature.

Automatically send problem reports to Adiscon Support

File Configuration fields:	nReportCrash
Description:	If enabled, problem reports will automatically be uploaded to http://crashdump.adiscon.com. A problem report is generated if the service internally stops working for some unknown reason. The reports are small dumpfiles which do not contain any personal data and will help us find and fix the problem. Also the

	dumpfiles are very small and do not exceed 256 Kbyte. In most cases only 32Kbyte data is send.

4.4.4 Engine

Engine specific Options Tab

The Engine specific Options are explained below:

Engine specific Options

Action specific

Enable retry of Actions on failure

File Configuration fields:	nEnableRetry
Description:	If enabled, the Agent retries Actions on failure (until the retry counter is reached). Note that the Event error 114 will only be written if the last retry failed, previous error's will only be logged in the debug log (With the error facility). Note that you can customize the Retry Count and the Retry Period in ms as well.

Rule Engine specific

Abort Rule Execution when one Rule fails?

File Configuration	bAbortRuleOnFailure

fields:	
Description:	If checked, and an action fails, the execution will be aborted. If unchecked, and an action fails, simply the next action in this rule will be executed.

DNS Cache Options

Enable internal DNS Cache

File Configuration fields:	nEnableDNSCache
Description:	The DNS cache is used for reverse DNS lookups. A reverse lookup is used to translate an IP address into a computer name. This can be done via the resolve hostname action. For each lookup, DNS needs to be queried. This operation is somewhat costly (in terms of performance). Thus, lookup results are cached. Whenever a lookup needs to be performed, the system first checks if the result is already in the local cache. Only if not, the actual DNS query is performed and the result then stored to the cache. This greatly speeds up revers host name lookups. However, computer names and IP addresses can change. If they do, the owner updates DNS to reflect the change. If we would cache entries forever, the new name would never be known (because the entry would be in the cache and thus no DNS lookup would be done). To reduce this problem, cache records expire. Once expired, the record is considered to be non-exisiting in the cache and thus a new lookup is done. Also, cache records take up system memory. If you have a very large number of senders who you need to resolve, more memory than you would like could be allocated to the cache. To solve this issue, a limit on the maximum number of cache records can be set. If that limit is hit, no new cache record is allocated. Instead, the least recently used record is overwiritten with the newly requested one.

How long should DNS names be cached?

File Configuration fields:	nDNSCacheTime
Description:	This specifies the expiration time for cache records. Do not set it too high, as that could cause problems with changing names. A too low-limit results in more frequent DNS lookups. As a rule of thumb, the more static your IP-to-hostname configuration is, the higher the expiration timeout can be. We suggest, though, not to use a timeout of more than 24 to 48 hours.

How many DNS records can be cached?

File Configuration fields:	nDNSCacheLimit
Description:	This is the maximum number of DNS records that can be cached. The system allocates only as many memory, as there are records required. So if you have a high limit but only few sending host names to resolve, the cache will remain small.

However, if you have a very large number of host names to resolve, it might be useful to place an upper limit on the cache size. But this comes at the cost of more frequent DNS queries. You can calculate about 1 to 2 KBytes per cache record.

Prefered protocol for name resolution

File Configuration fields:	nDNSInetProtocol
Description:	Select if you wish to prefer IPv4 or IPv6 addresses for name resolution. Note that this only has an effect on names which return both, IPv4 and IPv6 addresses.

Ressource Library Cache Options

How long should libraries be cached?

File Configuration fields:	nLibCacheTimeOut
Description:	This feature will be mainly useful for EventLog Monitor. For events with the same reoccuring event sources, this will be a great performance enhancement. The cache will also work for remote system libraries (requires administrative default shares). All libraries will be cached for 30 minutes by default.

4.4.5   QueueManager

Queue Manager Tab

Queue Manager Options

Queue Manager DiskCache

This feature enables the Agent to cache items in its internal queue on disk using a fixed data file.

First of all a Warning. Only use this feature if you really need to!

Depending on the speed of your hard disks, it will slow down processing of the actions, in worst case if the machine can't handle the IO load, the Queue will become full sooner or later. The DiskCache is an additional feature for customers, who for example want to secure received Syslog messages which have not been processed yet.

The diskcache will not cache infounits from services like EventLog Monitor, as this kind of Service only continues if the actions were successfully. All other information sources like the Syslog Server will cache it's messages in this file. If the Service or Server crashes for some reason, the queue will be loaded automatically during next startup of the Agent. So messages which were in the queue will not be lost. Only the messages which was currently processed during the crash will be lost.

Enable Queue Manager Diskcache

File Configuration fields:	nEnableRingBuffer
Description:	Enable the disk based queue manager. Please read the description about the Queue Manager DiskCache first!

File and Pathname

File Configuration fields:	szRingBufferFile
Description:	As everywhere else, you can define here, where the queue file should be stored.

Queue File Size

File Configuration fields:	nRingBufferSize
Description:	With this slider, the queue size can be set from 1 MB to 2048 MB.

Processing pointer

File Configuration fields:	nProcessingLow
Description:	Points to the current processing position within the queue file.

Saving Pointer

File Configuration fields:	nSavingLow
Description:	Points to the last processed position within the queue file.

Queue Manager specific

Number of worker threads

File Configuration fields:	nWorkerThreads
Description:	Defines the number of worker background threads that the core engine uses to process it's queue.

4.5 Services

4.5.1 Understanding Services

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

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

4.5.2 Event Log Monitor

This dialog configures the Windows Event Log Monitor service.

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

Newer Windows versions come with a considerably changed event logging system. In theory, the Event Log Monitor works with them, too. However, **we know of some incompatibilities.** For best results, we recommend using the Event Log Monitor V2 service, which was specifically written for Windows Vista and newer. The Event Log Monitor described here is applicable for Windows 2000, 2003 and XP (where the new event logging system is not available). The Client will automatically detect and load available EventLog types during the first startup of the Event Log Monitor.

Event Log Monitor	Event Log Monitor V2
Windows 2000	Windows Vista
Windows XP	Windows 2008
Windows 2003	Windows 7
	Windows 8
	Windows 2012
	Windows 10
	Windows 2016

Event Log Monitor Properties

The most important part of this dialog is the table in the middle. It specifies which event logs are to be monitored. In the screenshot above, the monitor is set to all Windows-native event log types that can possibly occur. However, there might also be custom event logs. Such custom logs can be created by any application. For example, an application "MySuperApp" might create an event log "MySuperAppLog". Then, it might log its messages into this log and not the Windows application event log.

In order to support such custom event logs, the log monitor allows an unlimited number of additional logs to be added to it. In order to do so, press the "Insert" button. A new log is added to the bottom of the list. Then, you can edit its name in the "Eventlogtype Name" combox (yes, you can overwrite the provided values!).

Logs checked in the table are actually processed. Those unchecked are kept in the configuration, but are not processed.

General Options
The General Options available on this form are explained below:

Sleep Time

File Configuration fields:	nSleepTime
Description:	The event log monitor periodically checks for new event log entries. The "Sleep Time" parameter specifies how often this happens. This value is in miliseconds. We suggest a value of 60,000 milliseconds for the "Sleep Time". With that setting, the event log monitor checks for new events every 60 seconds. Larger periods can be specified for occasionally connected systems or if email delivery with few emails per day is intended. Very security-aware environments might use a shorter interval. The event log monitor service is specifically designed to limit the burden on the monitored system. As such, resource usage is typically low, even with frequently run event log checks. However, we recommend not running the event log monitor more often than once a second.

Overrun Prevention Delay

File Configuration fields:	nPreventOverrunDelay
Description:	This property allows configuring a delay after generating an event. The time is the delay in milliseconds. If run at a value of zero, the event log monitor service generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because the service runs at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the service can still generate 1000 events per second. The default setting is an overrun protection of five millisecond, which allows roughly 200 events per second. This should be sufficient for even very busy servers.

Prefered Language

File Configuration fields:	nLanguageLCID
Description:	You can select a prefered language, and the Eventlog Monitor will send the message in this language. It will only work if these languages are installed and message libs are available with the prefered language. If this fails, it will automatically fall back to the system default language.

Enable Remote EventLog Monitoring

File Configuration fields:	nEnabledRemote
Description:	If enabled, the EventLog Monitor will read and process the EventLog from a remote machine. Use the verify button to make sure that the network connection is working

> correctly
>
> Please make sure that the machine, which you are going to monitor, does have File and Print Services enabled and is accessible by this machine.
> This is important as the EventLog Service will read the message libraries on the remote machine by using the default administrative shares. For this reason, the Service must be configured to run with a user who has administrative privileges/permissions on the local and remote machine. If File and Print services remain disabled, the local message libraries will used automatically instead. Note that you may experience a lot of missing message libraries in this case.
>
> Additionally you have an option to **read the EventLog Sources** from the local machine. If enabled, the local message libraries will be used instead of the remote machines ones. Sometimes local Event Sources are more reliable or are required for thirdparty EventLog implementations.

Compress Control Characters

File Configuration fields:	nCompressControlChars
Description:	This option allows you to control the control character removal and space compression. If checked, control characters (e.g. CR, LF, TAB - non printable characters in general) are removed. Also multiple spaces are compressed to a single one. By default this is checked. We recommend keeping it checked for most cases as it provides better display. **Please note that it should be unchecked if events should be forwarded via email. And it MUST be turned off if double-byte character sets are being processed (e.g. Japanese).**

Do NOT process existing entries when Event log corruption occurs

File Configuration fields:	nDoNotProcessLastRecord
Description:	When this option is checked, it prevents from reprocessing of the whole Windows event log when it is corrupted or truncated. So EventReporter / MonitorWare Agent do not processes all entries again.

Do NOT process existing entries on Service Startup

File Configuration fields:	szSyslogTagValue
Description:	

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

Remove Control Characters from String Parameters

File	nRemoveControlCharsFromParams

Configuration fields:	
Description:	Enable this option to remove control characters like carriage return or line feeds from parameter strings and category names in Winows Events.

Default Buffersize

File Configuration fields:	nDefaultBuffer
Description:	The default Buffersize is 10k. This value will be increased or decreased dynamically if necessary. If you want to use thirdparty applications like Netapp you must increase the Buffersize manually (minimum 65k), because dynamic adjusting is not possible with them.

SyslogTag Value

File Configuration fields:	szSyslogTagValue
Description:	The SyslogTag Value determines the SyslogTag that is used when forwarding Events via syslog. This is useful, if you want to determine later, what kind of syslog message this is, perhaps because you log EventLogs and syslog into the same database.

How to handle Eventlog Corruption

File Configuration fields:	nEventLogCor 0 = Retry processing from beginning 1 = Ignore corrupted Eventlog entry 2 = Clear all events from Eventlog
Description:	Sometimes it can occur that Eventlog messages are corrupted and cannot be read correctly. This usually happens if someone tampered with the Eventlog or if you are processing the Eventlog for the first time. In cases like this, you can automatically handle the situation with this option. You have the following options: - **Retry processing Eventlog from the beginning**: in this case the complete Eventlog will be processed again. - **Ignore corrupted Eventlog entry (default):** the affected Eventlog entry will be ignored and processing will continue. - **Clear all Events from the Eventlog:** the Eventlog will be cleared completely and new Events hopefully don't get corrupted before they are processed.

Use Legacy Format

File Configuration fields:	bUseLegacyFormat
Description:	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 to database option for example always stores the fields separated, so there is no specific option to do so).

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

Add Facility String

File Configuration fields:	bAddFacilityString
Description:	If checked, facility identification is perpended to the message text generated. This parameter enhances compatibility with existing Syslog programs and greatly facilitates parsing the generated entries on the Syslog server. We strongly encourage users to use this enhancement. **This setting only applies if the "Use Legacy Format " option is checked.** Otherwise, it does not have any meaning and consequently cannot be configured in that case.

Add Username

File Configuration fields:	nAddUserName
Description:	If checked, the Windows user that generated the event log entry is transmitted. If unchecked, this information is not forwarded. This is a configurable option for customers who have written parsing scripts for a previous format which did not contain Usernames. This option must also be unchecked if MoniLog is being used. **This setting only applies if the "Use Legacy Format " option is checked.** Otherwise, it does not have any meaning and consequently cannot be configured in that case.

Add Log Type

File Configuration fields:	nAddLogType
Description:	If checked, then log types e.g. system, security etc. etc. is perpended to the generated message.

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

Syslog Message Numbers

File Configuration fields:	bShowSyslogMsgNbr
Description:	If checked, a continuously advancing message number is perpended to the generated message. This is useful for Syslog delivery to make sure that all messages have been received at the remote server. **This setting only applies if the "Use Legacy Format " option is checked.** Otherwise, it does not have any meaning and consequently cannot be configured in that case.

Delay writing LastRecord

File Configuration fields:	nEnableLastRecordDelay
Description:	Enables the LastRecord writeback delay to the configured properties below.

Save after amount of entries

File Configuration fields:	nLastRecordDelayCount
Description:	The LastRecord will be written after the amount of processed eventlog entries that are specified here.

Event Log Channels Tab

The "Event Log Channels" are basically a list of the different log types. The corresponding log is only be processed if the respective "Enable" checkbox is checked. The parameters are common to all logs and are explained only once.
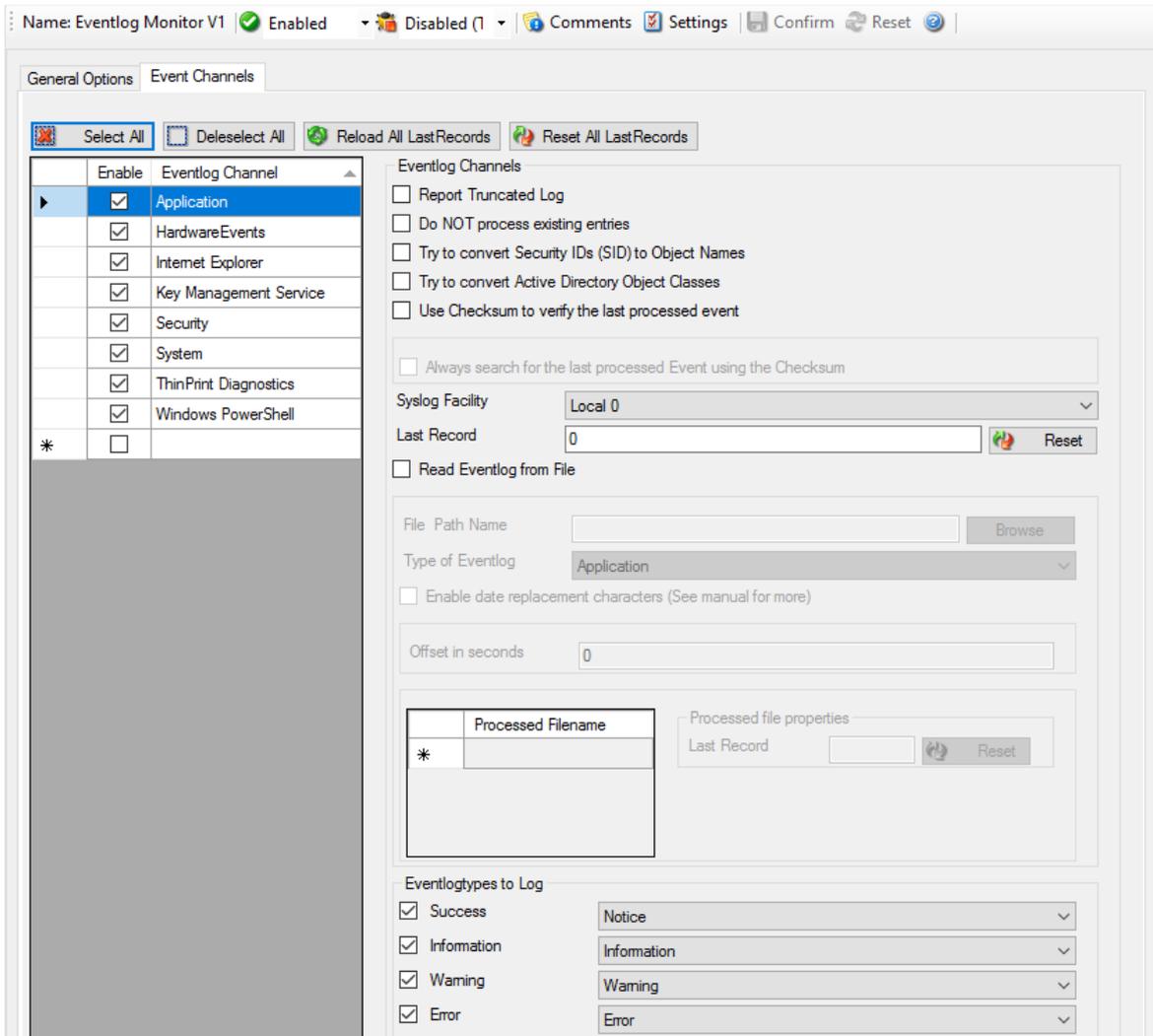
Event Log Monitor - EventLog Types General Options

Report Truncated Log

File Configuration fields:	bReportTruncatedLog
Description:	Windows event logs can be truncated programmatically or via the Windows Event Viewer program. When a log is truncated, all information is erased from it. Any entries not already processed by the service are lost. The service detects event log truncation. If "Report Truncated Log" is checked, it generates 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 Windows event logs as part of your day-to-day operation, we suggest you turn this option off. In this case, we also recommend using a short sleep period (for example 10,000 which is 10 seconds) to avoid losing log entries.

Do not Process Existing Entries

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

Try to convert Security IDs (SID) to Object Names

File Configuration fields:	nTryConvSIDtoObj
Description:	With this option you can convert Security ID's (SIDs) to object names. You can enable this feature in the advanced configuration of each event log type in the Event Log Monitor service. Simple check the "Try to convert Security IDs (SID) to Object Names" option. **Note that only the Security event log has this feature enabled by default. For all other event log types this feature is disabled by default.**

Try to convert Active Directory Object Classes

File Configuration fields:	nTryConvertDsClasses
Description:	With this option you can convert ActiveDirectory Schema GUID's from Security Events on Domain Controllers to object names. For Example Event 565, which usually has a lot of these Schema GUID's! The GUID's are internally cached to speed up EventLog processing operations. **Note that only the Security event log has this feature enabled by default. For all other event log types this feature is disabled by default.**

Use Checksum to verify the last processed Event

File Configuration fields:	nEventUseChecksum
Description:	A checksum of the last processed Event will be stored along with the LastRecord of an eventlog. This checksum is checked during each iteration. If the checksum does not match , we consider the EventLog has been altered, cleared or something else happened. In this case the EventLog is being reprocessed from the beginning. Please note: This option will prevent you from modifying the LastRecord value. If you do, the whole EventLog will be reprocessed! Please note that this behavior is by design and cannot be avoided. So we recommend to use this feature only if you intend to double check if the LastRecord value is valid.

Always search for the last processed Event using this Checksum

File Configuration fields:	nEventScanLastEventByChecksum
Description:	Usually, the last processed Event is determined by the LastRecord value. If the Checksum to verify the last processed Event is activated, then thie option to always search for the last processed Event using the Checksum is available. When activated, the last processed Event will also be always determined by the Checksum, not the LastRecord value.

Syslog Facility

File Configuration fields:	nFacility
Description:	The Syslog facility to map information units stemming from this log to. Most useful if the message is to forward to a Syslog daemon.

Last Record

File Configuration fields:	nLastRecord
Description:	Windows event log records are numbered serially, starting at one. The service records the last record processed. This textbox allows you to override this value. **Use it with caution!** If you would like a complete dump of a specific Windows event log, reset the "Last Record" to zero. If you missed some events, simply reset it to some lower value than currently set. It is possible to set "Last Record" to a higher value. This suspends event reporting until that record has been created. We strongly discourage to use this feature unless definitely needed.

Read Eventlog From File

File Configuration fields:	nReadFromFile
Description:	When enabled, the Eventlog is read from a file instead from the system.

File&Path Name

File Configuration fields:	szLogFileName
Description:	It defines that which file to be read, only available when "Read Eventlog From File" is enabled.

Type of Event Log

File Configuration fields:	szLogType

Description:	It defines as which type of Event log from file is handled. This is important to read the correct message libs from the system.

Enable date replacement characters

File Configuration fields:	nEnableDateReplacements
Description:	Allow the use of dynamic files/paths when using evt files. The same replacement characters as in the FileMonitor apply to this feature. A configured filename may look like this: C:\temp\evt_%Y%m%d.evt and would be replaced with C:\temp\evt_20130101.evt. To support changing log file names, there are replacement characters available within the file name. These are: Character Replacement Table

Character	Meaning
%y	Year with two digits (e.g. 2002 becomes "02")
%Y	Year with 4 digits
%m	Month with two digits (e.g. March becomes "03")
%M	Minute with two digits
%d	Day of month with two digits (e.g. March, 1st becomes "01")
%h	Hour as two digits
%S	Seconds as two digits. It is hardly believed that this ever be used in reality.
%w	Weekday as one digit. 0 means Sunday, 1 Monday and so on.
%W	Weekday as three-character string. Possible values are "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat". This replacement character is most useful for DHCP log files.
%generatedfilename%	It contains the fully generated filename (Can be useful for filtering).
%msgsep%	Only available if enable in the advanced option of the File Monitor. This value contains the current used messageseperator. This is usefull if you want to reconstruct messages where the seperator is part of the message.
%msgseplast%	Only available if enable in the advanced option of the File Monitor. This value contains the last used messageseperator. This is usefull if you want to reconstruct messages where the seperator is part of the message.

Offset in Seconds

File Configuration fields:	nEnableDateReplacementsOffset
Description:	When "Enable date replacement characters" is enabled, the current date will be used to generate the filenames. However in certain cases, there is a need to generate filenames with past or future dates. Negative values will generate past filenames, while positive values will generate filenames in the future. For example if

	you want to generate filenames from yesterday (24 hours back), use -84600 as offset.

Event Types to Log

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

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

General Values (Common settings for most services)

Default Ruleset Name

File Configuration fields:	szRuleSetName
Description:	Name of the rule set to be used for this service. The Rule Set name must be a valid Rule Set.

Please Note if you intend to make the Event ID part of the actual Syslog message while forwarding to a Syslog Server then you have to make some changes in the Event Log Monitor Settings. Click here to know the settings.

The event log monitor caches messages libraries. This greatly speeds up processing, but causes memory consumption for the cached libraries. By default, libraries are chached for 30 minutes. If memory consumption is too high, you may consider to lower the cache period. The cache is global to all event log monitors. As such, its size must be changed in the general settings.

4.5.3   Event Log Monitor V2

This dialog configures the Windows Event Log Monitor V2 service for Windows Vista, Windows 2008, Windows 7, Windows 8, Windows 2012 and Windows 10. For Windows 2000, 2003 and XP use the classical event log monitor.

Event Log Monitor	**Event Log Monitor V2**
Windows 2000	**Windows Vista**
Windows XP	**Windows 2008**
Windows 2003	**Windows 7**
	Windows 8
	Windows 2012
	Windows 10
	Windows 2016

The V2 event log monitor provides the ability to monitor so-called "log channels". Each channel can work either in polling of subscription mode. In subscription mode, we are automatically notified by the operating system whenever a new event is logged. In traditional polling mode, we periodically check the channel. In both cases, it is possible for a user to re-set the channel reporting to an older event (parameter "Last Record").

Both of these functionalities are implemented by periodically iterating over the configured channels. The frequency is controlled by the "Sleep Time" parameter.

Event Log Monitor Properties

Overrun Prevention Delay

File Configuration fields:	nPreventOverrunDelay
Description:	This property allows configuring a delay after generating an event. The time is the delay in milliseconds. If run at a value of zero, the event log monitor service generates events as fast as the machine permits. We have seen scenarios where routers and receivers are not able to keep up with this rate, resulting in packet loss. In addition, the CPU of the reporting machine is run at 100% - which is not a problem because the service runs

at a low priority. However, with even a 1-millisecond delay, there is no noticeable CPU activity even when large bursts of events are forwarded. At one millisecond, the service can still generate 1000 events per second.

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

Select Message Format

File Configuration fields:	nFormatType 0 = XML Format 1 = Predefined EventFormat
Description:	With this option you can choose whether the Events will be extracted in "Raw XML Format" or in the "Predefined Event Format". The XML format is the exact representation of the XML Stream returned by the EventLog System. **Please note that it only contains EventLog data and not a formatted message.** The "Predefined Event Format" is what the Event in the event viewer looks like.

SyslogTag Value

File Configuration fields:	szSyslogTagValue
Description:	The SyslogTag Value determines the SyslogTag that is used when forwarding Events via syslog. This is useful, if you want to determine later, what kind of syslog message this is, perhaps because you log EventLogs and syslog into the same database.

Sleep Time

File Configuration fields:	nSleepTime
Description:	As said in the overview, this controls iteration over the configured channels. The value is specified in milliseconds. For channels configured to use Polling Mode, the "Sleep Time" parameter specifies how often they are processed. Note that when multiple channels are set to polling mode, they are processed one after another. So there is a somewhat larger delay in processing than given by the "Sleep Time" parameter. The total frequency depends on how busy all polling channels are. For channels configured to subscription mode, the "Sleep Time" interval will only influence how often a potential reset of "Last Record" is checked. Other than that, it has no effect on the event delivery rate. We suggest a value of 60,000 milliseconds for the "Sleep Time". With that setting, the event log monitor checks for new events every 60 seconds. Larger periods can be

specified for occasionally connected systems or if email delivery with few emails per day is intended.

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

Note that it is almost always an error to use a "Sleep Time" value of 0. The main processing loop of the EventLog Monitor V2 would re-run without any delay and would cause a very high CPU usage, close to 100%. For these reasons, future versions of the product will no longer permit to use a "Sleep Time" of zero and automatically change it to one.

Wait time after action failure

File Configuration fields:	nSubscriptionSleepTime
Description:	Adds some extra wait time (Delay) if an action failed to process. Without the delay, the subscription would immediately process the last event again. In some cases a reasonable delay before a retry is needed.

Emulate %Param% properties from old EventLog Monitor

File Configuration fields:	nEmulateParameters
Description:	This option emulates the %Param% properties, which were often used in the old EventLog Monitor. The new EventLog implementation (e.g Windows 7, Windows Server 2008 Windows 8, Windows Server 2012) does not support them in the same way anymore. The Event Log Monitor V2 is still able to provide parameters in the "old style" format, what means that log analysis scripts can receive a consistent stream of data for both new style and old style Windows events.

Include optional Event Parameters as properties

File Configuration fields:	nIncludeEventParameters
Description:	If enabled, the < EventData> node from the raw XML stream (Eventlog entry) will be searched for variables. If variables with names are found, they will be set as Properties with their variable name automatically. If the variable does not have a name, it will be set to a common name like "Param1, Param2 …. ParamX".

Convert to EventLog Monitor V1

File Configuration fields:	nConvertToEventLogMonitorV1
Description:	This option maps the EventID's from the Security EventLog back to V1 (Windows 2000/2003). The internal InforUnitID is also changed to V1. This option helps

| | postprocessing EventLog V1 and V2 events equally. |

Delay writing LastRecord

File Configuration fields:	nEnableLastRecordDelay
Description:	Enables the LastRecord writeback delay to the configured properties below.

Save after waittime

File Configuration fields:	nLastRecordDelayTime
Description:	Regardless of the amount of processed eventlog entries, the lastrecord value will delayed for this waittime period.

Save after amount of entries

File Configuration fields:	nLastRecordDelayCount
Description:	Regardless of the configured waittime period, the LastRecord will be written after the amount of processed eventlog entries that are specified here.

Event Channels Tab
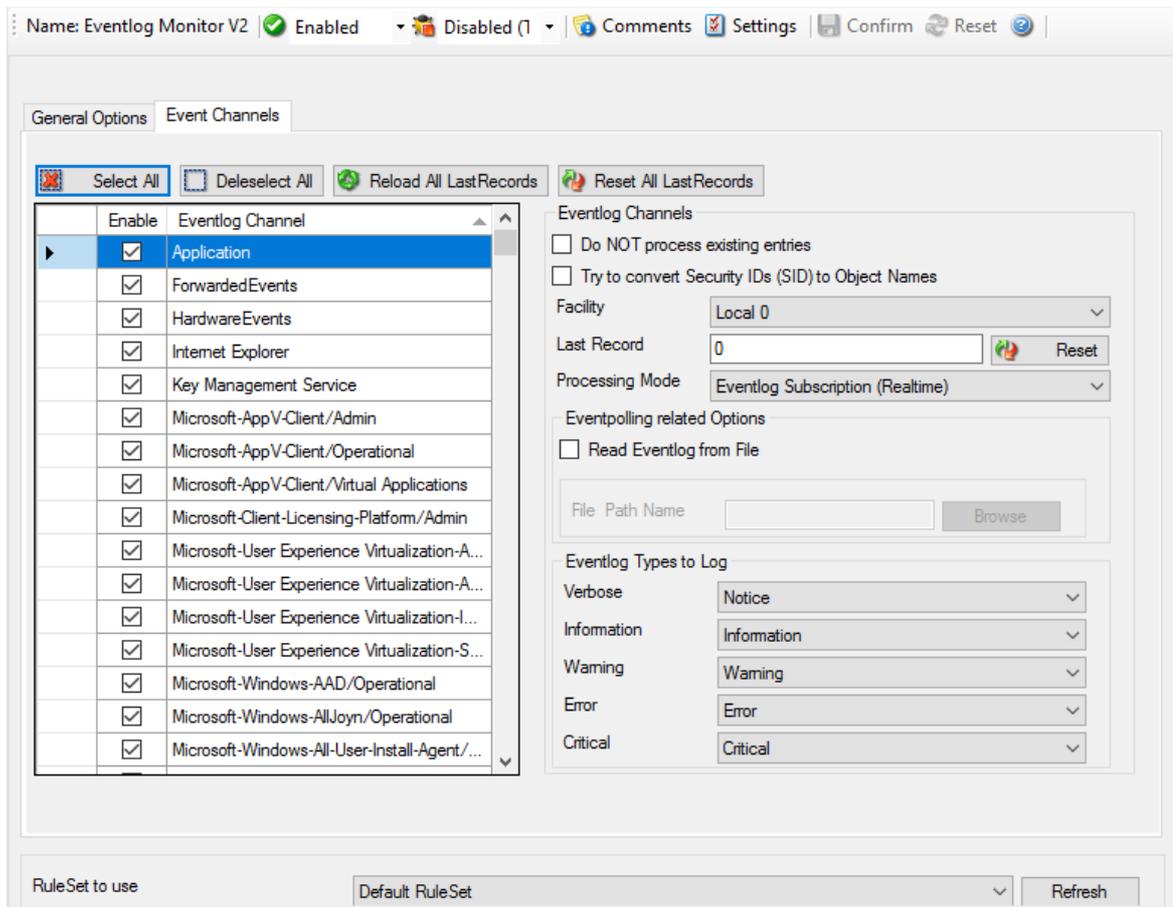
The most important part of this dialog is the treeview of available Channels. It specifies which event logs are to be monitored. In the screenshot above, the monitor is set to all Channels that are currently available. There happen to be custom Channels, too, due to Applications creating them on their own. They will be added to the treeview automatically every time you re-open this configuration window.

Here you can adjust the syslog facility and the event log types. You are also able to overwrite all existing custom advanced channel configurations with your new ones.

Channels

Channels which are checked in the table will be processed. Channels which are unchecked are kept in the configuration, but are not processed.

Facility

File Configuration fields:	nFacility
Description:	The Syslog facility to map information units stemming from this log to. Most useful if the message is to forward to a Syslog daemon.

Last Record

File Configuration fields:	szXMLBookmark
Description:	Windows event log records are numbered serially, starting at one. The service records the last record processed. This textbox allows you to override this value. **Use it with caution!** If you would like a complete dump of a specific Windows event log, reset the "Last Record" to zero. If you missed some events, simply reset it to some lower value than currently set. It is possible to set "Last Record" to a higher value. This suspends event reporting until that record has been created. We strongly discourage to use this feature unless definitely needed.

Processing Mode

File Configuration fields:	nApiReadMode 0 = Subscription Readmode (Realtime) 1 = Polling Readmode (Sleeptime)
Description:	There are two processing modes available, first the default processing mode is "EventLog Subscription" which processes Events in realtime. This means events are send to MonitorWare Agent by the OS as they happen, there is no delay at all. The other processing mode called "Eventlog Polling" and is similar to the method used in the old EventLog Monitor. The EventLog is checked and processed periodically controlled by the sleeptime. However using the polling method, you enable the "Read EventLog From File" option.

General Values (Common settings for most services)

Default Ruleset Name

File Configuration fields:	szRuleSetName
Description:	Name of the rule set to be used for this service. The Rule Set name must be a valid Rule Set.

4.5.4   Heartbeat

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

Heartbeat Properties

Message to Send

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

Sleep Time

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

General Values (Common settings for most services)

Syslog Facility

File Configuration fields:	nSyslogFacility
Description:	The Syslog facility to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Syslog Priority

File Configuration fields:	nSyslogPriority
Description:	The Syslog priority to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Resource ID

File Configuration fields:	szResource
Description:	The Resource ID to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Syslog Tag Value

File Configuration fields:	szSyslogTagValue
Description:	The Syslog tag value to be assigned to events created by this service. Most useful if the message is to forward to a Syslog server.

Rule Set to Use

File Configuration fields:	szRuleSetName
Description:	Name of the rule set to be used for this service. The Rule Set name must be a valid Rule Set.

4.5.5   MonitorWare Echo Reply

The Echo Reply service is used on each of the installed EventReporter/MonitorWare Agent. A central agent running the MonitorWare Agent is using the echo request and instructs to poll each of the other EventReporter/MonitorWare Agent services. When the request is not carried out successfully, an alert is generated. The MonitorWare echo protocol ensures that always a fresh probe of the remote EventReporter/MonitorWare Agent Service is done.

MonitorWare Echo Reply Properties

Internet Protocoltype

File Configuration	nInetType

fields:	
Description:	Select the desired protocol type. IPv4 and IPv6 are available. The IPv6 protocol needs to be properly installed in order to be used. Note that one Service can only handle IPv4 or IPv6, so if you want to use both protocols, you will need to create two separate services.

Listener Port

File Configuration fields:	nListenPort
Description:	Specify the listener port here.

IP Adress

File Configuration fields:	szMyIPAddress
Description:	The MonitorWare Echo Reply service can be bound to a specific IP Address. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address. This feature is useful for multihome environments where you want to run different Syslog Servers on different IP Addresses. Please note that the default IP Address 0.0.0.0 means ANY IP Address.

4.6 Filter Conditions

4.6.1 Filter Conditions

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

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

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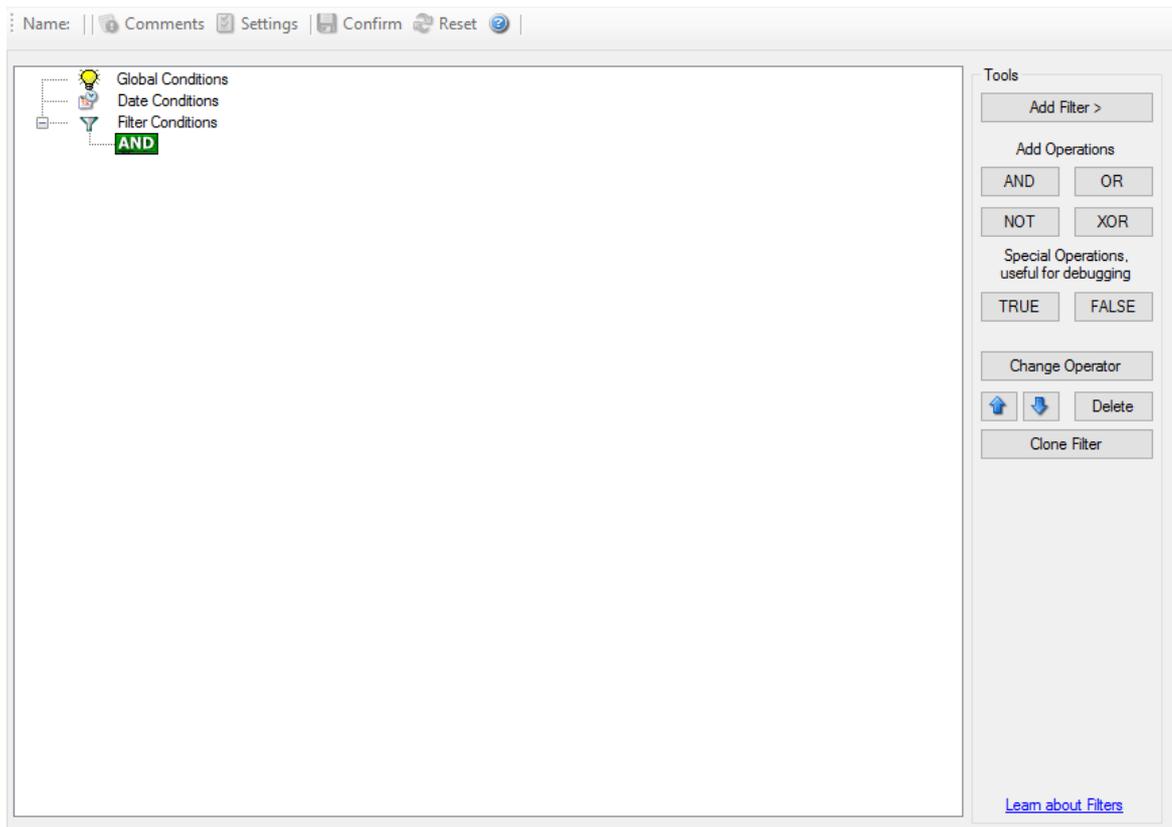
Filter Conditions - Display form

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

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

Filter Conditions - Complex Filter

This filter condition is part of an intrusion detection rule set. Here, Windows file system auditing is used to detect a potentially successful intrusion via Internet Information Server (IIS). This is done by enabling auditing on all executable files. Internet Information Server accesses 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 is triggered.

Here is how the above sample works: first, the message contents are checked if it contains either the full path name to perl.exe or php.exe. This is done in the "OR" branch at the bottom. We now need to keep in mind that when a filter condition evaluates to "true", the actions are executed. In case of perl.exe and php.exe, this is just the opposite of what we want. We need it to be executed, when other files are executed. Consequently, we negate (Boolean "NOT") the result of the OR. The end result of the "NOT" operation is then combined via a "AND" with some other properties describing the event we need.

First, we check if the specific event really occurred. For this, we need to make sure we deal with an Event Log Monitor information unit. Then, these information units are identified by the event source as well as the Event ID. We also check for the Event User to identify only IIS generated requests. Lastly, we check if the message contains the string ".exe".

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

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

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

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

4.6.2 Global Conditions

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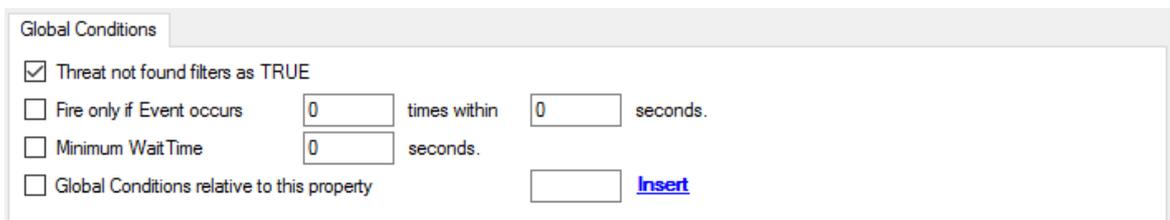
Filter Form - Global Conditions

Treat not found Filters as TRUE

If a property queried in a filter condition is not present in the event, the respective condition normally returns "FALSE". However, there might be situations where you would prefer if the rule engine would evaluate this to "TRUE" instead. With this option, you can select the intended behaviour. If you check it, conditions with properties not found in the event evaluates to "TRUE.

Fire only if Event occurs

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

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

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

Minimum Wait Time

This filter condition can be used to prevent rules from firing too often. For example, a rule might be created to check the status of a port probe event. The port probe probes an SMTP server. If the event is fired and the rule detects it, it spawns a process that tries to restart the service. This process takes 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 generates an additional event.

Setting a minimum wait time prevents 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 is not match. If, however, the same event is generated 5 hours later (with the mail gateway failing again), the rule once again fired and corrective action taken.

Global Conditions relative to this property

This feature enables you to control the Global Conditions based on a property.
For example take the source of a message as property. In this case, the Minimum WaitTime for example would be applied individual on each message source.

4.6.3   Date Conditions

Rule processing can be bound the a specific or installation date. By default a Rule will always be processed.

Filter Form - Date Conditions

Always process Rule

No date filter will be applied

Process only after Installation Date

Rule will only be processed if message was generated / received after the application installation date.

Process only after custom date

Rule will only be processed if message was generated / received after the custom specified date.

4.6.4   Operators

In general, operators describes how filter conditions are linked together. The following operators can be used.

AND

All filters placed below must be true. Only then AND returns true.

OR

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

NOT

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

XOR

Only one of the two filters are possible in the XOR Operator.

TRUE

Useful for debugging, just returns TRUE.

FALSE

Useful for debugging as well, returns FALSE.

4.6.5   Filters

Filters can be added under each Operation node. There are a few common filters which can be used for all services, and there are special filters which only apply if a special kind of Information Unit is evaluated.

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

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

An example, you have one ruleset, rule and action. In the filters you have one EventID filter. Then you have two services, one Eventlog Monitor and the other is Heartbeat monitor both pointing to this ruleset. The Information Units from the Eventlog Monitor

would be filtered correctly, but those from the Heartbeat monitor would not be filtered as they don't have an EventID property. The EventID filter would be ignored and the actions would be executed every time.

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

There are different types of filters, 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 =", "Range Match" or through REGEX.

Number

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

Boolean

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

Time

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

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

4.6.5.1 REGEX Compare Operation

REGEX Compare Operation

The property will be evaluated against a regular expression. Everything known in the regular expression syntax can be used to define a matching pattern.

Here are some regular expressions samples:
Regular Expression:
[0-9]{4,4}-[0-9]{1,2}-[0-9]{1,2}[0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}
Matches typical Date like 2015-11-20 12:11:01

Regular Expression: \n[0-9]{4,4}
Matches Linefeed and 4-digit number.

Regular Expression: (;|:) Matches semicolon or a colon.

More samples and details about the Regular Expression Syntax can be found here:
https://msdn.microsoft.com/en-us/library/bb982727(v=vs.90).aspx

4.6.6 General

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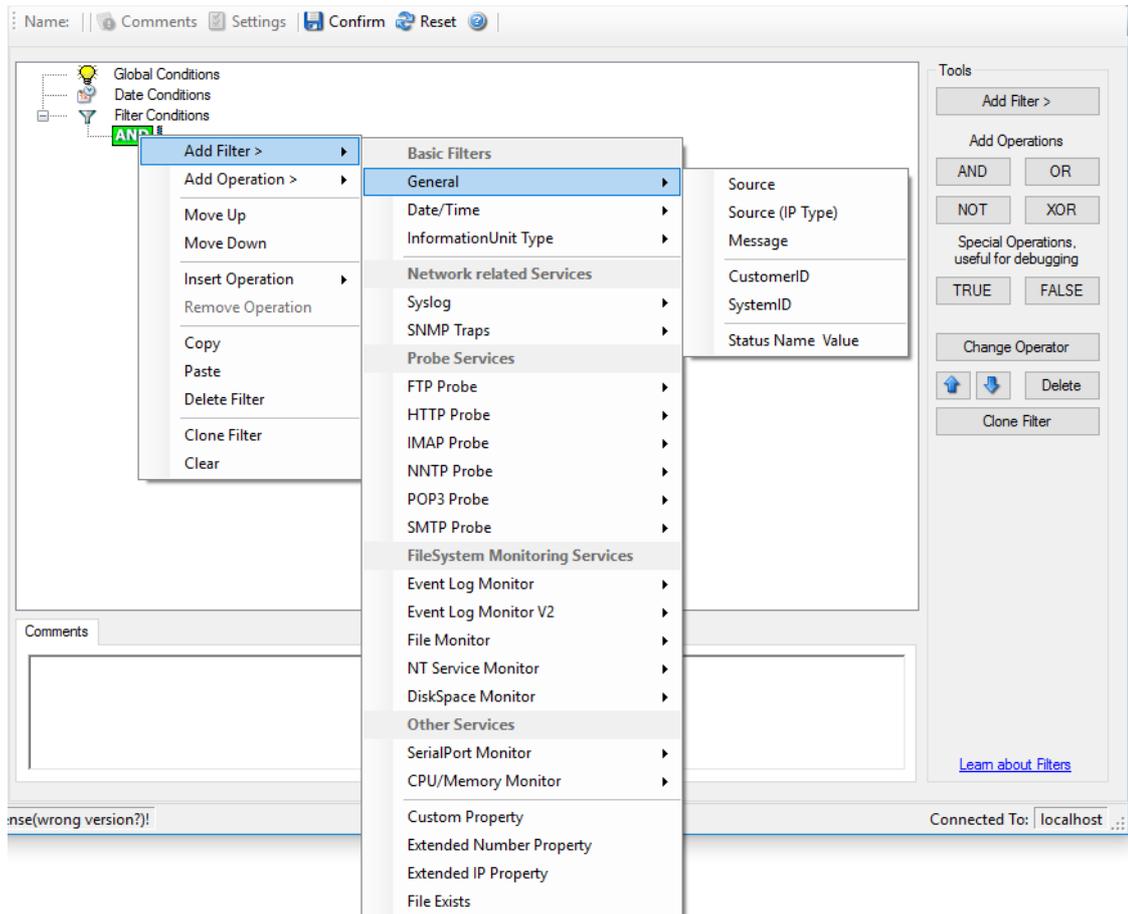
Filter Conditions - General

Source System

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

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

Source System (IP)

The IP Filter can basically work on any property, but we recommend to only use it on the %source% property, as we usually can be sure that this is contains a valid IP Address or hostname. The IP Filter can filter against hostnames and IP Addresses, hostnames are automatically resolved using the internal DNSCache (for obvious performance reasons).

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

Please see the description for "Extended IP Property" for more information on how to use this property.

Message Content

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

The content search can be limited to a region within the message. To do so, select a starting and ending position within the string by choosing the "**contains within range**" compare opertaion. This can be done by specifying the start range and end range into the respective boxes.

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

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

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

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

This filter is of type string.

CustomerID

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

CustomerID (Type=Number).

SystemID

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

SystemID (Type=Number).

Status Name and Value

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

4.6.7 Date/Time

This filter condition is used to check the time frame and / or day of week in which an event occurred.
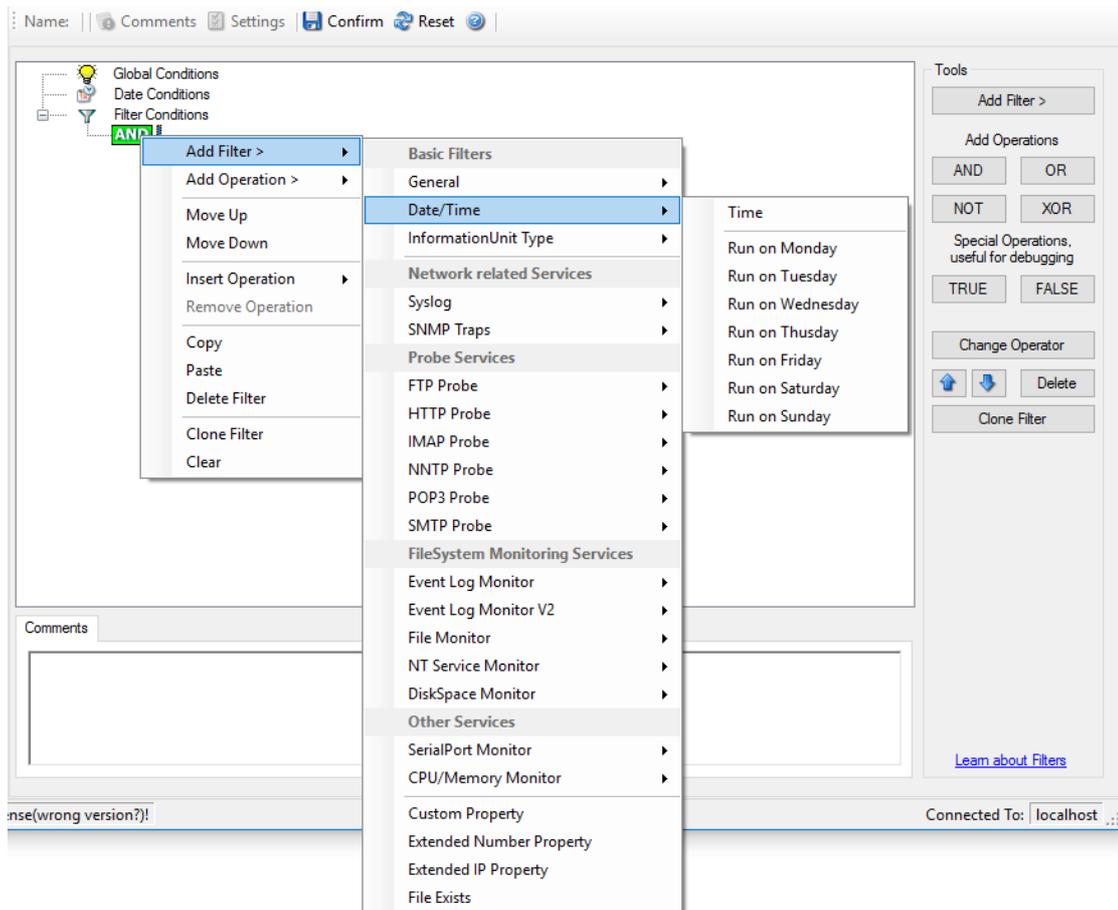
Filter Conditions - Date / Time

Time

This filter condition is used to check the period 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.

You can also set the timezone setting (DefaultTimemode, UTC or Localtime) for the TimeMode's (DeviceReportedTime/ReceivedTime).

Weekdays

This is closely equivalent to the time filter condition, except that it is applied on a per-day basis. So it can be used to detect for example events occurring on weekends and act differently on them. The following filters are available:

1.   Run on Monday (Type=Boolean)
2.   Run on Tuesday (Type=Boolean)
3.   Run on Wednesday (Type=Boolean)
4.   Run on Thursday (Type=Boolean)
5.   Run on Friday (Type=Boolean)
6.   Run on Saturday (Type=Boolean)
7.   Run on Sunday (Type=Boolean)

4.6.8   InformationUnit Type

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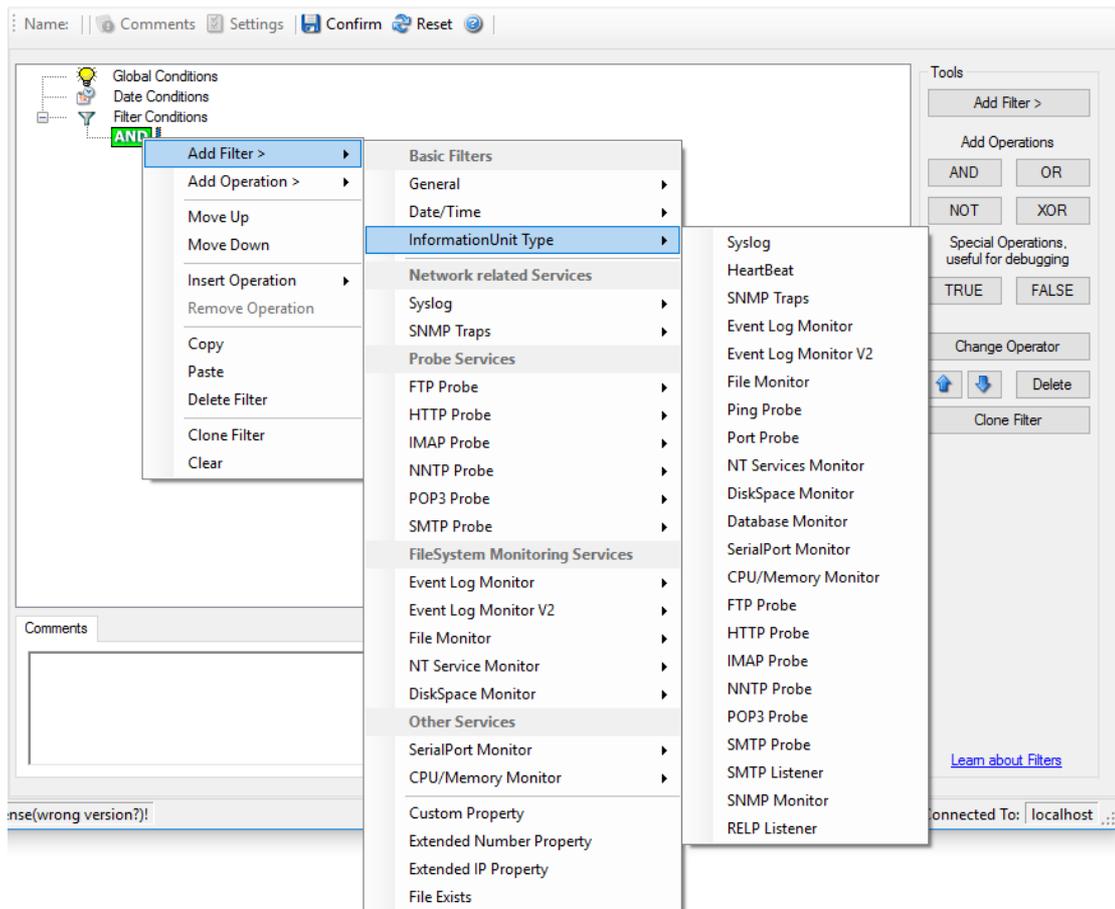
Filter Conditions - InformationUnit Type

The following filters are available:

1.   Syslog (Type=Boolean)
2.   Heartbeat (Type=Boolean)
3.   SNMP Traps (Type=Boolean)
4.   Event Log Monitor (Type=Boolean)
5.   File Monitor (Type=Boolean)
6.   Ping Probe (Type=Boolean)
7.   Port Probe (Type=Boolean)
8.   NT Services Monitor (Type=Boolean)
9.   Disk Space Monitor (Type=Boolean)
10.   Database Monitor (Type=Boolean)
11.   Serial Port Monitor (Type=Boolean)
12.   CPU/Memory Monitor (Type=Boolean)
13.   FTP Probe (Type=Boolean)
14.   HTTP Probe (Type=Boolean)
15.   IMAP Probe (Type=Boolean)
16.   NNTP Probe (Type=Boolean)
17.   POP3 Probe (Type=Boolean)
18.   SMTP Probe (Type=Boolean)

4.6.9 Event Log Monitor

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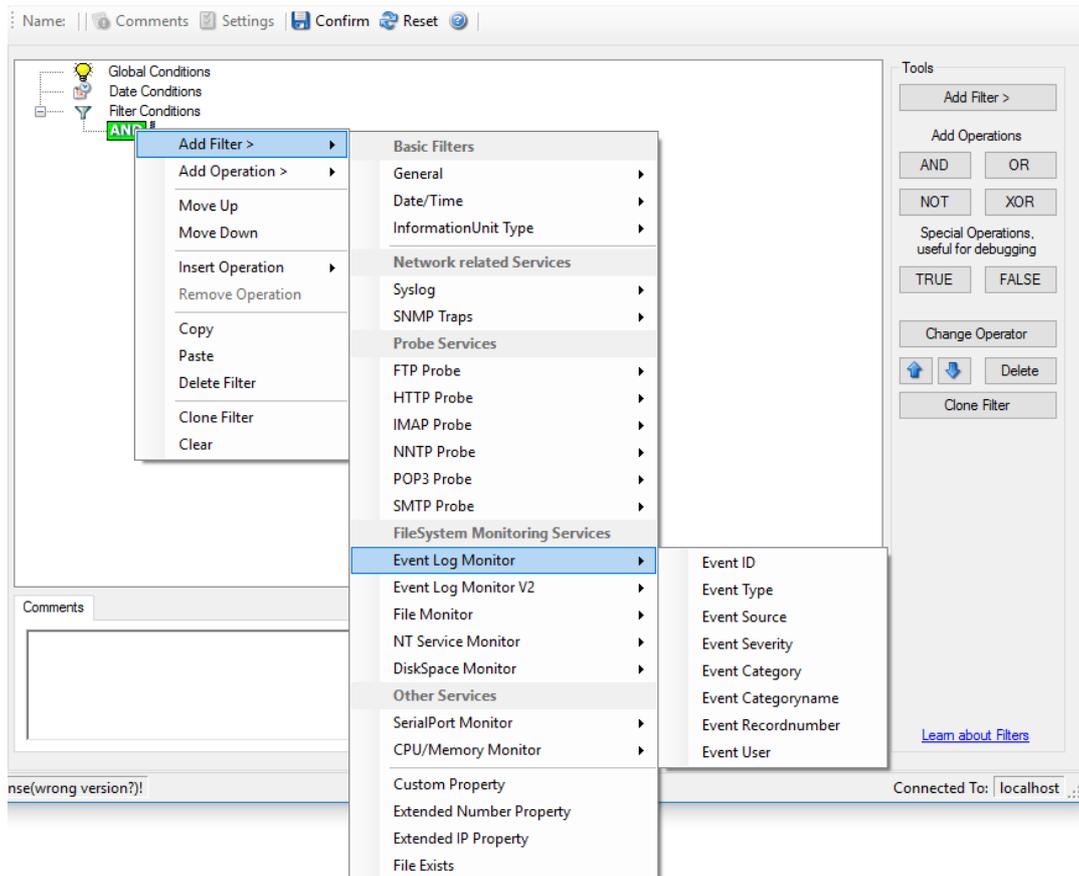
Filter Conditions - Event Log Monitor

Event ID

This is the event log ID as specified in the Windows 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 is to be used which might not properly reflect the actual value.

This filter is of type number.

Event Type

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

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

This filter is of type string.

Event Source

This is the event log source as specified in the Windows 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

Event Severity

This is the event log severity as specified in the Windows 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 is to be used which might not properly reflect the actual value.

This filter is of type number.

Event Category

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

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

This filter is of type number.

Event Categoryname

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

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

This filter is of type string.

Event Recordnumber

This value contains the internal event record number. Please note that if the event log has been truncated before, it may not start with 0 or 1 but a higher number.

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

This filter is of type number.

Event User

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

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

This filter is of type string.

4.6.10 Event Log Monitor V2

Event log monitor V2 specific filters are grouped here.
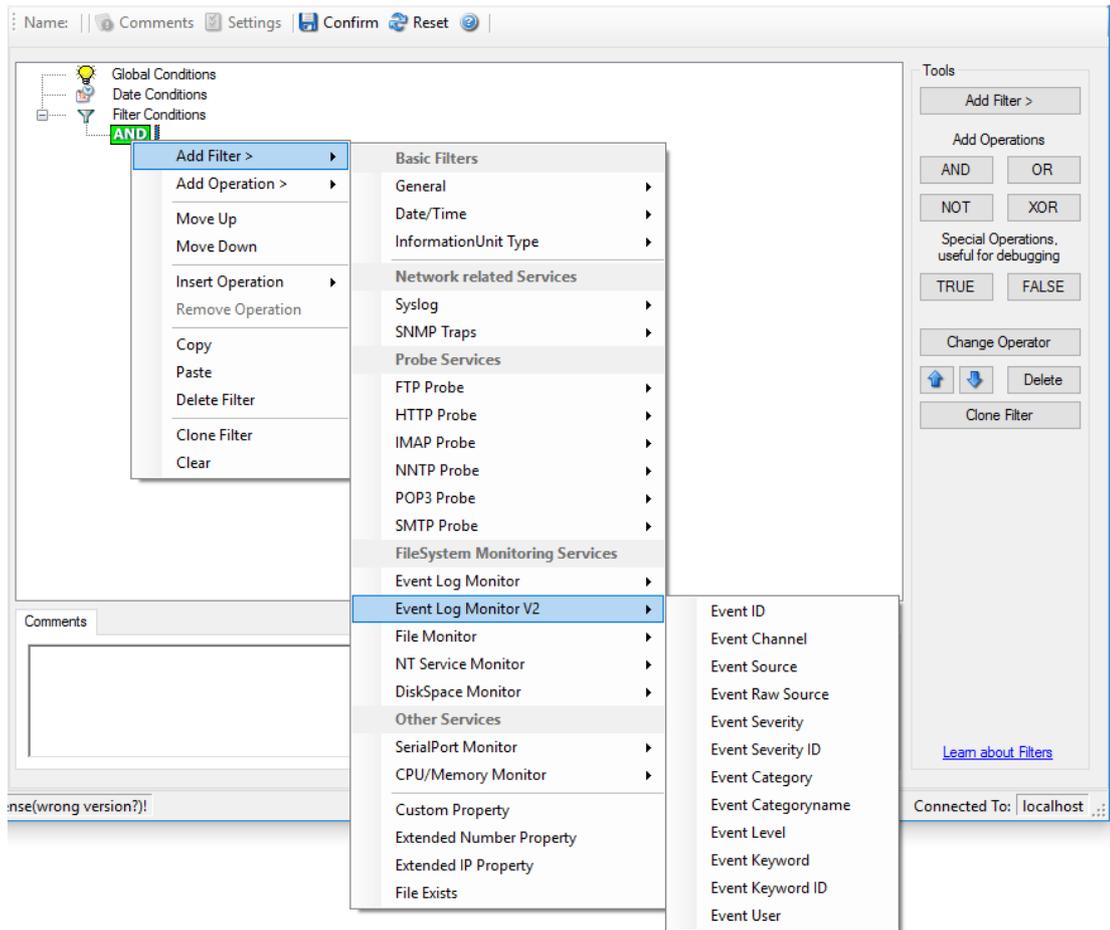
Filter Conditions - Event Log Monitor V2

Event Channel

The channel property for event log entries, for classic Event logs they match the % nteventlogtype% property, for new event logs, they match the "Event Channel". If enabled, the event must have the configured event type or the rule will not match. 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

Event Raw Source

This contains the full internal name of the event source for new event logs, for classic event logs it contains the same value as in %sourceproc%. 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

Event SeverityID

This is the internal ID of the event log level as number. This is a integer 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 is to be used which might not properly reflect the actual value.

This filter is of type number.

Event Level

This is a textual representation of the eventlog level (which is stored as number in % severityid%). This property is automatically localized by the system. If enabled, the event must have the configured level or the rule will not match. 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

Event Keyword

This is a textual representation of the event keyword. This property is automatically localized by the system. If enabled, the event must have the configured event keyword or the rule will not match. 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

Event KeywordID

This is the internal keyword ID as string. If enabled, the event must have the configured event keyword ID or the rule will not match. 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 is to be used which might not properly reflect the actual value.

This filter is of type string.

4.6.11 Custom Property

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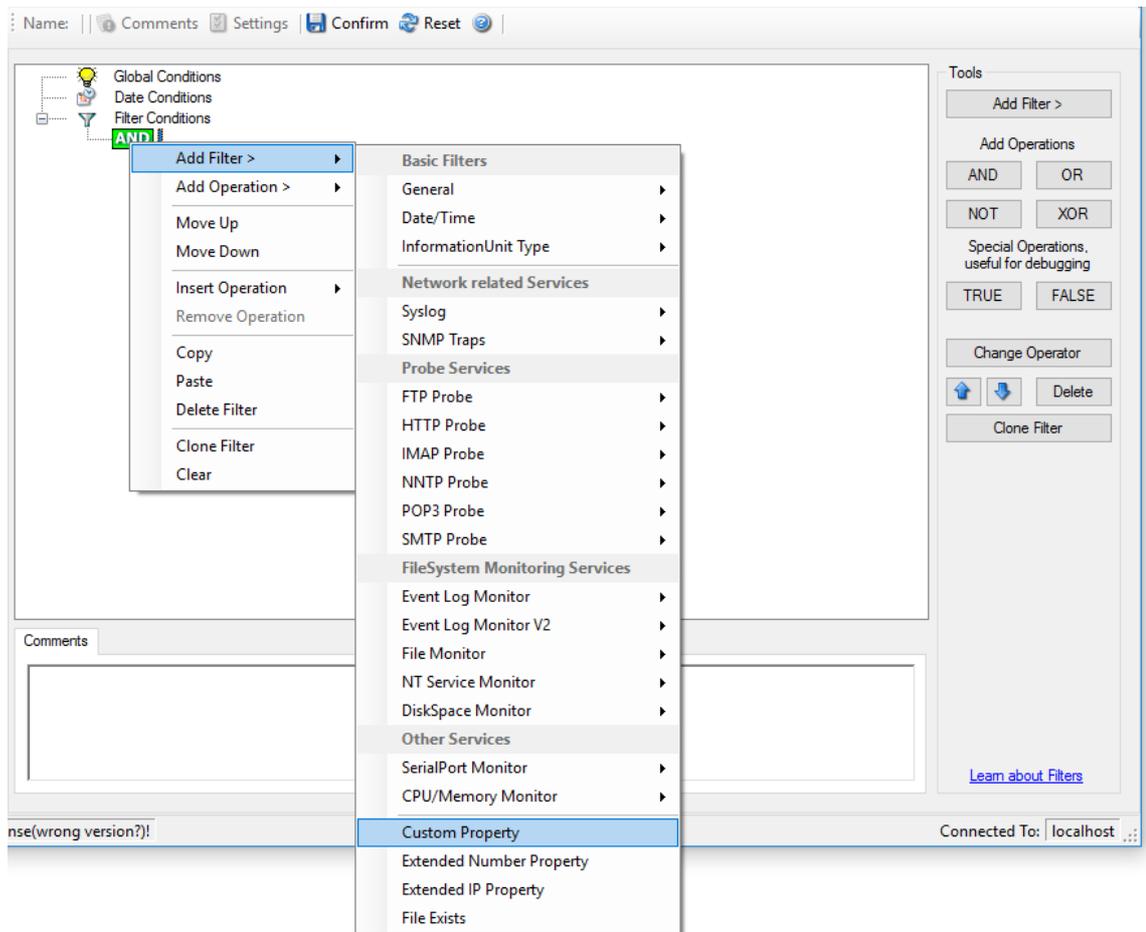
Filter Conditions - Custom Property

Custom Property

As the name suggests it is a "Custom Property". Internally in MonitorWare Agent all values are stored in properties. For example the main message is stored in a property called "msg". By using this dialog you can access properties which are dynamic (Like those from SNMP Trap Monitor when using V2 protocol).

This filter is of type string.

4.6.12 File Exists

Filter setting by string
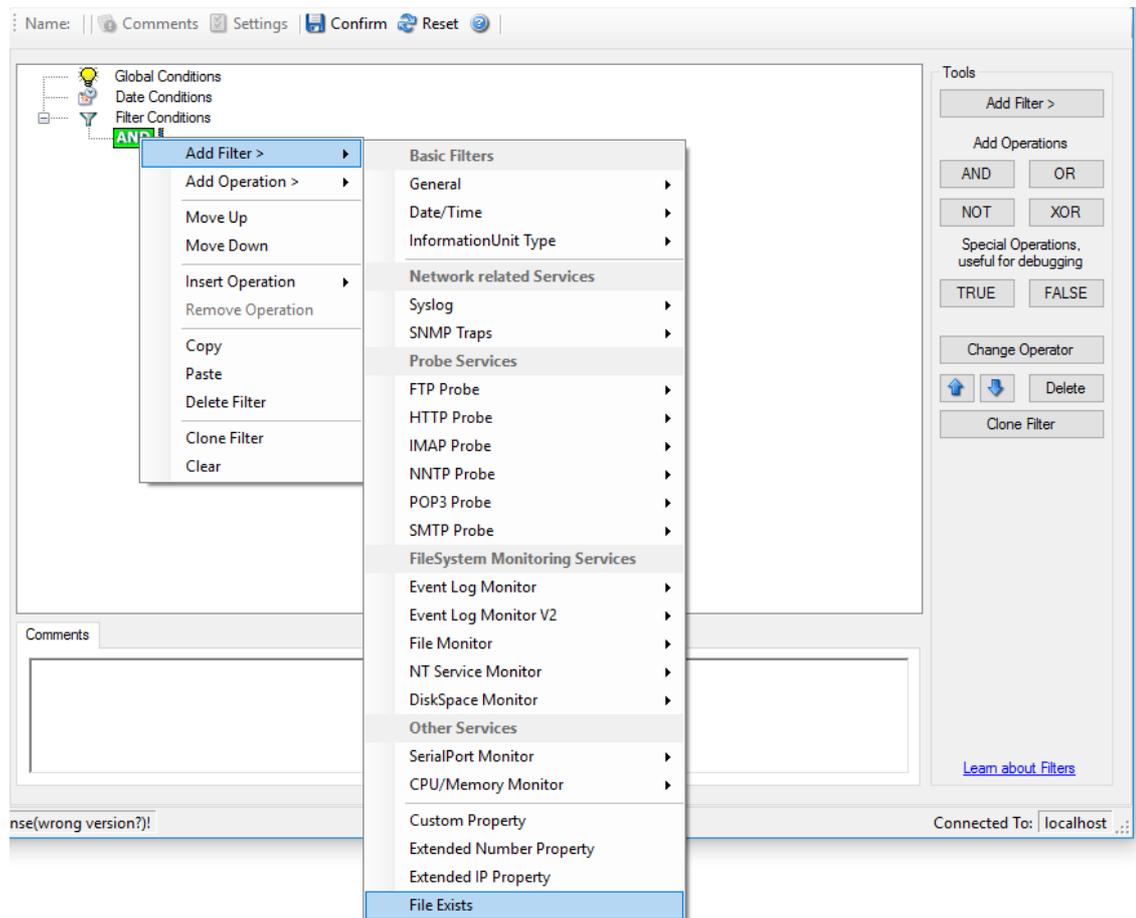
Filter Conditions - File Exists

File Exists

With this Filter you can simply check if a file exists or not. You can directly enter the file and its location or you can use the browse-button to find it.

4.6.13 Extended IP Property

Extended IP Property filter settings
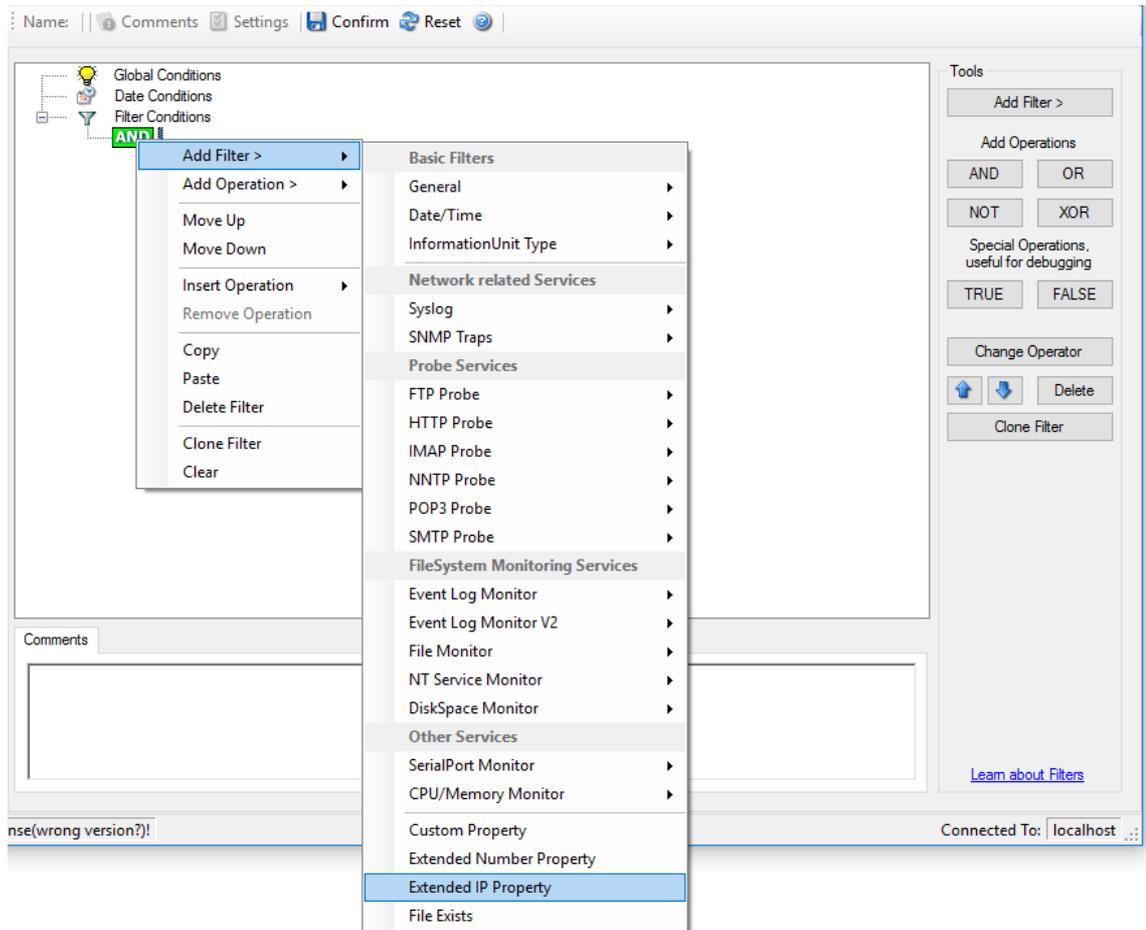
Filter Condition - Extended IP Property

The IP Filter can basically work on any property, but we recommend to only use it on the %source% property, as we usually can be sure that this contains a valid IP Address or hostname. The IP Filter can filter against hostnames and IP Addresses, hostnames are automatically resolved using the internal DNSCache (for obvious performance reasons). If you are going to use a different or custom property, please make sure, that the data in the property is a valid IP Address.

Available compare operations for the IP Filter Type are:
Equal (=): The IP Address must match the one you configured in the Property Value field.
Not Equal (!=): The IP Address must not match the one you configured in the Property Value field.
Higher (>): The IP Address must be higher than the one you configured in the Property Value field.
You can use IP Address Formats like 192.168.0.10, 192.168.0, 192.168 or even 192. It depends on what IP Ranges you are going to filter for.
Lower (<): The IP Address must be lower than the one you configured in the Property Value field.
You can use IP Address Formats like 192.168.0.10, 192.168.0, 192.168 or even 192. It depends on what IP Ranges you are going to filter for.

If you want to filter for IP Ranges, I recommend to use two filters to define the range, one filter with the "Higher (>)" compare operation, and one with the "Lower (<)" compare operation. This could look like the following:

Filter Condition - Filtering for an IP Range

The filter you can see here will accept all IPs which lie between 172.16.0.110 AND 172.16.0.130. That means, that for every IP that matches these two conditions, the whole filter will evaluate to true and therefore the message will be processed. If the filter does not evaluate to true, the rule will be aborted and the message is sent to the next rule.

4.6.14 Store Filter Results

How to store Filter Results is described here.

Filter Conditions - Store Filter Results Property

Store Filter Results

If a filter matches, you can now store the result of the match into a custom property. This custom property can be used in Actions later.

4.7 Actions

4.7.1 Understanding Actions

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

There can be multiple actions for each rule.
Actions are processed in the order they are configured.
However you can change the order of the actions by moving them Up or Down.

4.7.2 Resolve Hostname Action

Many Customers asked for resolve hostname options in different services. This feature has now been implemented as an action. An action can be used with every service, and it doesn't delay the work of a service. See the Screenshot and Descriptions below on how to configure it correctly:

Figure1: Resolve Hostname Action

Select Source Property from which the name will be resolved:

File Configuration fields:	szSourcePropertyName

Description:	Click on the Insert menu link on the right side of the textfield to customize the source property from which the name will be resolved.

Destination Property in which the resolved name will be saved to:

File Configuration fields:	szDestinationPropertyName
Description:	Same as above, please click on the Insert menu link on the right side of the textfield to customize the destination property in which the resolved name will be saved to.

Also resolve name if the source property is already a name.

File Configuration fields:	nResolveIfName
Description:	Activates the feature that the name will also be resolved if there is already a source property with that name.

Cache resolved host entry

File Configuration fields:	nCacheNameEntry
Description:	If activated this will, as it says, cache the resolved host entry.

4.7.3 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 Windows 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>

Parameters in the brackets can be configured via dialog shown below:

File Logging Options

Enable Property replacements in Filename

File Configuration fields:	nEnablePropertyFileName
Description:	By activating this option, you can use properties within the file or pathname like %Source% and all the others. For example: File Path Name can be F:\syslogs\%source% File Base Name can be IIS-%source% If your source is 10.0.0.1, that writes the following file: F:\syslogs\10.0.0.1\IIS-10.0.0.1.log Please note that the path f:\syslogs\10.0.0.1 was generated because the source poperty was used inside the path. Note: You can use ANY property inside the path and base name. Event properties are described in the property replacer section.

Timeout until unused filehandles are closed

File Configuration fields:	nCleanFileHandlesTimeout
Description:	When dynamic filenames are used, filehandles are cached internally to avoid massive amount of File open/close operations. This timeout specifies after which time handles should be finally closed if not used anymore. Each write to a file will reset the timeout counter for the current filehandle.

File Path Name

File Configuration fields:	szFilePath
Description:	The base path (directory) of the file. Please see above for exact placement. Default is "c:\temp". The Insert Menu entry allows you to create "Dynamic Directories". For example: File Path Name can be **F:\syslogs\%source%** Event properties are described in the property replacer section.

File Base Name

File Configuration fields:	szFileBaseName
Description:	The base name of the file. Please see above for exact placement. Default is "MonitorWare". The Insert Menu entry allows you to recreate "Dynamic Base Filenames". For example: File Base Name can be **IIS-%source%**

File Extension

File Configuration fields:	szFileExtension
Description:	The extension to be used when writing the file. Please see above for exact placement. Default is ".log".

Create unique Filenames

File Configuration fields:	nUniqueFileName
Description:	If checked, a unique file name is created 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 is a single file with consistent file name. Some customers that have custom scripts to look at the file name use this.

Include Source in Filename

File Configuration fields:	nIncludeSourceInFilename
Description:	If checked, the file name generation explained above is modified. The source of the Syslog message is 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 straight forward with this single checkbox. If it is checked, the messages are automatically written to separate files and the file name includes the originating device information.

Use UTC in Filename

File Configuration fields:	nUseUTCInFileName
Description:	This works together with the "Create unique Filenames" setting. If unique names are to be created then select the "Use UTC in Filename" option, in this case the file name is generated on the basis of universal co-ordinated 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 7 pm New York time. If it were unchecked, the rollover would occur exactly at midnight New York time (5 am UTC). Using UTC for file name creation can be helpful if log files are written among different time zones and later consolidated. Using UTC ensures a consistent time notation across all log files. **Please note that this setting does affect the file name creation only. A different setting controls the dates recorded inside the file.**

Segment files when the following file size is reached (KB)

File Configuration fields:	nSegmentFileEnable
Description:	Files are segmented when the defined file size is reached. The file name will have a sequence number appended (_1 to _n).

Event properties are described in the property replacer section.

File Logging Options #2

Use Circular Logging

File Configuration fields:	nCircularLogging
Description:	When enabled log files are created and over written in a cycle.

Number of Log files

File Configuration fields:	nNumberOfLogfiles
Description:	Once the last logfile is reached, circular logging begins and over write the first log file again.

Maximum File size

File Configuration fields:	nMaxFileSize
Description:	Max filesize of a log file, once this size is reached a new logfile is created.

Clear logfile instead of deleting (File will be reused)

File Configuration fields:	nReUseFile
Description:	This option causes the File Action to truncate the logfile instead of deleting and recreating it.

File Format

File Configuration fields:	nFileFormat 0 = WinSyslog 1 = Raw Syslog message 2 = WebTrends Syslog compatible
Description:	This controls the format that the log file is written in. The default is "Adiscon", which offers most options. Other formats are available to increase log file compatibility to third party applications. The "Raw Syslog message" format writes raw Syslog format to the log file. That is, each line contains the Syslog message as of RFC 3164. No specific field processing or information adding is done. Some third party applications require that format. The "WebTrends Syslog compatible" mimics the format that WebTrends applications expect. Please note that we only mimic the log file format. It is still the job of the reporting device (most notable firewall) to generate the correct WebTrends WELF format. The "WebTrends" format is supported because many customers would like to use MonitorWare Agent 3.0 enhanced features while still having the ability to work with WebTrends. The "Custom" format allows you to customize formats to increase log file compatibility for third party applications. When you choose this option then Custom line format is enabled. **Please note that any other format besides "Adiscon Default" is a fixed format. As such, if it is selected, all other formatting options do not apply and consequently are turned off.**

General file options

Under this group box, you can see two options discussed as under:

Use XML to Report

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

Use UTC for Timestamps

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

Include <Fieldname>

File Configuration fields:	nFileDateTime nFileDateTimeReported nIncludeMessage nIncludeRAWMessage nFileSource nIncludeSourceInFilename nFileFacility nFilePriority
Description:	The various "include" settings controls at the bottom are used to specify the fields which are to be written to the log file. All fields except the message part itself are optional. If a field is checked, it is 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 when MonitorWare Agent 3.0 received the message. Therefore, it is always a consistent value. In contrast, the "Date and Time Reported by Device" is a timestamp taken from the actual message. As such, it is dependent on the reporting device clock, which might be off. In addition, in the case of Syslog messages, there is no time zone information within the device reported timestamp. As such, if devices from multiple time zones are reporting, the timestamp information is not consistent. This is due to Syslog design as of RFC 3164. The Syslog server can be configured to ignore the RFC in this case and provide a consistent time stamp. However, from the view of the log file writer, the "Date and Time Reported by Device" might not be as trustworthy as the "Date and Time" field. Nevertheless, it might also be more useful than the former one. This is the reason both timestamps are present and can individually be selected. The "Include Message" and "Include RAW Message" fields allow customizing the message part that is being written. The raw message is the message as – totally unmodified, was received. 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 are written. Similarly, if none is selected no message is written at all. Please note that we support these configurations, too – there might be a legitimate need for them.

Custom Line Format

File Configuration fields:	szLineFormat
Description:	Custime Line Format enables you to fully customize the output for the log file. The Insert Menu entry provides further options and they only work in custom line format. Default value is "%msg%%$CRLF%".

4.7.4 Database Options

Use database logging to store messages into a database.

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

Once stored inside the database, different message viewers as well as custom applications can easily browse them. The defaults for the write database action are suitable Adiscon Loganalyzer (web interface).

The database format can be fine-tuned. This is most useful if you intend to run some additional analysis on the database. Also, in high volume environments, tuning the database action to exactly those fields need helps getting best performance out of the database.

The main feature of the "Write To Database" property sheet is the field list. The default reflects the typical assignment of event properties to database columns. However, you can modify this assignment in any way you like. You only need to keep in mind that Adiscon analysis products need the database contents as specified. As such, malfunctions may occur if you modify the database assignments and then use these tools.

Connection Options

Database Logging Options

Configure DSN

If you click on this button, it starts the ODBC administrator of the operating system where you can add, edit or remove a data source(s).
Please Note: The DSN must be a System DSN.

Verify Database

The configuration client will attempt to establish a database connection to your configured ODBC System DSN.

Create Database

If you click on this button, it will create the default tables for SystemEvents and SystemEventsProperties into the database specified in the DSN.

DSN

File Configuration fields:	szODBCDsn
Description:	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). Press the "Data Sources (ODBC)" button to start the operating system ODBC administrator where data sources can be added, edited and removed. **Important**:The DSN must be a system DSN, not a user or file DSN. The DSN must be configured to have the correct connection parameters (for example database type and name, server name, authentication mode etc.).

User-ID

File Configuration fields:	szODBCUid
Description:	The User-ID used to connect to the database. It is dependant on the database system used if it is to be specified (e.g. Microsoft Access does not need one, while Microsoft SQL Server can force you to use one). If in doubt, please see your database administrator.

Password

File Configuration fields:	szODBCPwd
Description:	The password used to connect to the database. It must match the "User-ID". Like the User ID, it is dependent on the database system if a password is needed. Passwords can be stored either encrypted or unencrypted. We highly recommend storing them encrypted.

Enable Encryption

File Configuration fields:	nODBCEnCryption
Description:	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. Even when stored encrypted, we recommend using limited privileges accounts. We are not applying strong cryptography here.

SQL Connection Timeout

File Configuration fields:	nSQLConnectionTimeOut
Description:	Defines the Timeout for the connection.

Table Name

File Configuration fields:	szTableName
Description:	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".

SQL Statement Type

File Configuration fields:	nSQLStatementType
Description:	You can select between a INSERT and Call Statement, which is Microsoft specific for Stored Procedures. This means also this type of SQL Statement will only work if

| | MSSQL is used as database. If you select MSSQL Call Statement, the tablename field will automatically be used as stored procedure name. |

Output Encoding

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

Enable Detail Property Logging

File Configuration fields:	nPropertiesTable
Description:	This option logs event properties other than the standard properties to the SystemEventProperties table. A single event can potentially have multiple properties, so selecting this option can result in multiple writes. With Syslog data, however, there are seldom any additional properties. They most often occur when you use the "Post Process" action to define your own properties. Additional properties are typically found in SETP received data originating from an event log monitor, file monitor or database monitor (plus other monitors, but these are the most prominent ones).
	For example, with Event Log data received via SETP, these properties contain the actually Windows event properties and the event data. Please note that this does not apply to event log messages received via Syslog, because they are no native events but rather Syslog data.
	Please make sure you actually need this before activating it. As a side note, some of the MonitorWare Console reports may need detail logging.

Insert NULL Value if string is empty

File Configuration fields:	nSQLConnectionTimeOut
Description:	This option inserts a NULL value, if a property is empty.

Datafields

The provided fieldnames are those that Adiscon's schema uses - you can add your own if you have a need for this.

You can edit the field list by selecting a row and then modifying the text fields above the table. You can insert and delete rows by selecting the respective button. If you press delete, the currently selected row is deleted.

For string data types, you can use the property replacer. This can be helpful if you would like to store a substring. For example, if you intend to store only the first 200 characters of each message, you can use "%msg:1:200%".

Fieldname

File Configuration fields:	szFieldName_[n]
Description:	The **Fieldname** is the database column name. It can be any field inside the table.

Fieldtype

File Configuration fields:	nFieldType_[n] 1 = varchar 2 = int 3 = text 4 = DateTime
Description:	**Fieldtype** is the data type of the database column. It must reflect the column type selected in the database. It must also be consistent in type with the actual property that must be stored. For example, an integer type property like the syslogpriority can be stored in a varchar column. A string data type like the syslogtag can - for obvious reasons - not be stored in an integer column.

Fieldcontent

File Configuration fields:	szFieldContent_[n]
Description:	Finally, the **Fieldcontent** is the event property. For a complete list of supported properties, see Event properties.

4.7.5   OLEDB Database Action

Due the changes to x64, it became more important to also support the newer database layer from Microsoft called OLEDB. The OLEDB Action works similar to the ODBC Action from configuration point of few. The MS SQL OLEDB Provider and JET4.0 OLEDB Provider have been successfully tested in the Win32 environment. Unfortunately, the JET4.0 Provider has not been ported to the x64 platform yet. In our internal performance tests, there was an enhancement of up to 30% compared to ODBC. So this action may also be interesting for people with a huge amount of incoming data.

This Action allows writing incoming events directly to any OLEDB - compliant database. Once stored inside the database, different message viewers as well as custom applications can easily browse them. The defaults for the write database action are suitable for Adiscon Loganalyzer (web interface).

The database format can be fine-tuned. This is most useful if you intend to run some additional analysis on the database. Also, in high volume environments, tuning the database action to exactly those fields need helps getting best performance out of the database.

The main feature of the "OLEDB Database Action" property sheet is the field list. The default reflects the typical assignment of event properties to database columns. However, you can modify this assignment in any way you like.

Connection Options

OLEDB Database Action Options

Configure OLEDB Connection

If you click on this button, it starts an OLEDB configuration wizard that will help you configuring your OLEDB datasource.

Verify Database

The configuration client will attempt to establish a database connection to your configured OLEDB Connection.

Create Database

If you click on this button, the configuration client will create the default tables for SystemEvents and SystemEventsProperties into your configured OLEDB database.

SQL Connection Timeout

File Configuration fields:	nSQLConnectionTimeOut
Description:	Defines the Timeout for the connection

Provider

File Configuration fields:	szProvider
Description:	OleDB Provider like SQL Server Client (SQLNCLI11.1). Should be filled automatically with **Configure OLEDB Connection** button.

Data Source

File Configuration fields:	szDataSource
Description:	Datasource is most often the servername or ip address like SERVERNAME\SQLEXPRESS for example. Should be filled automatically with **Configure OLEDB Connection** button.

Location

File Configuration fields:	szLocation
Description:	OLEDB Location. Should be filled automatically with **Configure OLEDB Connection** button.

Data Catalog

File Configuration fields:	szDataCatalog
Description:	Is the database name in most cases. Should be filled automatically with **Configure OLEDB Connection** button.

Username

File Configuration fields:	szUsername
Description:	Username used for authentication. Should be filled automatically with **Configure OLEDB Connection** button.

Password

File Configuration fields:	szPassword
Description:	Password used for authentication. Should be filled automatically with **Configure OLEDB Connection** button.

Encrypt password

File Configuration fields:	szPassword
Description:	Password used for authentication. Should be filled automatically with **Configure OLEDB Connection** button.

Table Name

File Configuration fields:	szTableName
Description:	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.**

Statement Type

File Configuration fields:	nSQLStatementType
Description:	You can select between a INSERT and Call Statement, which is Microsoft specific for Stored Procedures. This means also this type of SQL Statement will only work if MSSQL is used as database. If you select MSSQL Call Statement, the tablename field will automatically be used as stored procedure name.

Output Encoding

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

Enable Detail Property Logging

File Configuration	nPropertiesTable

fields:	
Description:	This option logs event properties other than the standard properties to the SystemEventProperties table. A single event can potentially have multiple properties, so selecting this option can result in multiple writes. With Syslog data, however, there are seldom any additional properties. They most often occur when you use the "Post Process" action to define your own properties. Additional properties are typically found in SETP received data originating from an event log monitor, file monitor or database monitor (plus other monitors, but these are the most prominent ones). For example, with Event Log data received via SETP, these properties contain the actually Windows event properties and the event data. Please note that this does not apply to event log messages received via Syslog, because they are no native events but rather Syslog data. Please make sure you actually need this before activating it. As a side note, some of the MonitorWare Console reports may need detail logging.

Maximum value length (Bytes)

File Configuration fields:	nMaxValueLength
Description:	Maximum length in bytes for values stored in Detaildata table.

Datafields

The povided names are those that Adiscon's schema uses - you can add your own if you have a need for this.

You can edit the field list by selecting a row and then modifying the text fields above the table. You can insert and delete rows by selecting the respective button. If you press delete, the currently selected row is deleted.

For string data types, you can use the property replacer. This can be helpful if you would like to store a substring. For example, if you intend to store only the first 200 characters of each message, you can use "%msg:1:200%".

The rest of this section describes the labelled paramters.

Fieldname

File Configuration fields:	szFieldName_[n]
Description:	The **Fieldname** is the database column name. It can be any field inside the table.

Fieldtype

File Configuration fields:	nFieldType_[n] 1 = varchar 2 = int 3 = text 4 = DateTime
Description:	**Fieldtype** is the data type of the database column. It must reflect the column type selected in the database. It must also be consistent in type with the actual property that must be stored. For example, an integer type property like the syslogpriority can be stored in a varchar column. A string data type like the syslogtag can - for obvious reasons - not be stored in an integer column.

Fieldcontent

| File Configuration fields:	szFieldContent_[n]
Description:	Finally, the **Fieldcontent** is the event property. For a complete list of supported properties, see Event properties.

4.7.6   Event Log options

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

Event Logging Options

Use logsource from service

File Configuration fields:	bUseCustomEventLog = 0
Description:	Takes the service name as logsource for the log entry. This option is enabled by default.

Replace Event Log Source

File Configuration fields:	bUseCustomEventLog = 1
Description:	If checked, a special mapping mechanism is activated. In this mode, the Windows event source is set to the IP address of the system sending the Syslog message. In addition, the ID is set to Syslog facility. This mode helps to quickly gather information about the system state in Windows event viewer.

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

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

Custom Event Source

File Configuration fields:	szCustomSource
Description:	EventSource is now fully configurable with all posibilities the property engine gives you. **Please note that content of this field can be configured. Event properties**

	are described in the property replacer section.

Use Custom Eventlog Type

File Configuration fields:	nEventType 0 = EVENTLOG_SUCCESS (Information event) 1 = EVENTLOG_ERROR_TYPE (Error event) 2 = EVENTLOG_WARNING_TYPE (Warning event) 4 = EVENTLOG_INFORMATION_TYPE (Information event) 8 = EVENTLOG_AUDIT_SUCCESS (Success Audit event) 16 = EVENTLOG_AUDIT_FAILURE (Failure Audit event)
Description:	The type – or severity – this log entry is written with. Select from the available Windows system values.

EventID

File Configuration fields:	nEventID
Description:	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 should be used for all custom messages. IDs below 10,000 should not be used as they might potentially interfere with events generated by MonitorWare Agent 3.0 itself.

Message to Log

File Configuration fields:	szMessagecontent
Description:	It is the message which will be logged into the Windows event log. It is fully configurable what is logged into the Eventlog. Please note that Insert Menu entry allows you to add replacement characters e.g. %msg% - you can write the actual message of an event into the Windows event log. **Please note that The message content of the message field can be configured. Event properties are described in the property replacer section.**

4.7.7   Mail Options

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

Mail Server Options

Forward Email Properties - Server Options

Mailserver

File Configuration fields:	szSMTPServer
Description:	This is the Name or IP address of the mail server to be used for forwarding messages. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address. 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.

Mailserver Port

File Configuration fields:	nSMTPPort
Description:	Port the mail server is to be contacted at. Usually, this is 25. It might, however, be changed 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.

Enable Backup Server, used if first Mailserver fails

File Configuration fields:	nEnableBackupServer
Description:	When enabled, you can configure a second Mailserver that will be used if the regular Mailserver is not available/accessible.

Backup Mailserver

File Configuration fields:	szSMTPServerBackup
Description:	In case that the connection to the main configured mail server can not be established, the backup mail server is tried. Note that an error is only generated, if the connection to the backup server fails as well.

Backup Mailserver Port

File Configuration fields:	nSMTPPortBackup
Description:	Port the mail server is to be contacted at. Usually, this is 25. It might, however, be changed 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.

Use SMTP Authentication

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

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

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

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

Session Timeout

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

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

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

The session timeout is user configurable between 1 and 2147483647 milliseconds (32bit integer) or different pre-set values. Larger values are not supported as they probably affect the SMTP server performance and can lead to unpredictable results.

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

Use a secure connection (SSL) to the mail server

File Configuration fields:	nUseSSL
Description:	This option enables SSL-secured traffic to the mail server. Please note, that this only works, if the receiving mail server supports SSL-secured transmission of emails.

Use STARTTLS SMTP Extension

File Configuration fields:	nUseStartTLSMethod
Description:	This extension is required for SMTP Servers which can optionally enable encryption during communication.

Use UTC time in Date-Header

File Configuration fields:	nUseUTCTimeStamp
Description:	Some Email Readers do not support UTC time in date-headers. Therefore here is a switch to turn the UTC time on or off.

Mail Format Options

Forward Email Properties - Format Options

Sender Emailaddress

File Configuration fields:	szSMTPSender
Description:	Some Email Readers do not support UTC time in date-headers. Therefore here is a switch to turn the UTC time on or off.

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

Recipient Emailaddress

File Configuration fields:	szSMTPRecipient
Description:	The recipient emails are addressed to. To send a message to multiple recpients, enter all recipient's email addresses in this field. Separate addressed by spaces, semicolons or commas (e.g. "receiver1@example.com, receiver2@example.com"). Alternatively, you can use a single email address and define a distribution list in your mail software. The distribution list approach is best if the recipients frequently change or there is a large number of them. Multiple recipients are also supported. They can be delimited by space, comma or semicolon.

Use legacy subject line processing

File Configuration fields:	nUseLegacySubjectProcessing
Description:	This checkbox specifies which type of subject line processing will be done. If it is checked, the old-style processing using single character replacement sequences is applied. If it is left unchecked, the far more powerful event property based method is used. In legacy mode, the following replacement characters are recognized inside the subject line:

%s

IP address or name (depending on the "resolve hostnames" setting) of the source system that sent the message.

%f

Numeric facility code of the received message

%p

Numeric priority code of the received message

%m

the message itself. Please note: this is the complete message text and can be rather lengthy. As such, it is most probably subject to truncation. If that occurs, all other information after the %m replacement character is also truncated. As such, we strongly recommend using the %m replacement at the end of the subject line only.

%%

It represents a single % sign.

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

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

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

Subject

File Configuration fields:	szSMTPSubject
Description:	Subject line to be used for outgoing emails and it is used for each message sent. It can contain replacement characters or "Event Properties" to customize it with event details. This is especially useful when sending email to cellular phones or pagers, which often display only the subject line and not the actual message body. The subject line – after expansion of the any replacement sequences – can hold a maximum of 255 characters. Characters beyond this will be truncated. Please note that many email systems impose a more strict limit and truncation may occur before the 255-character limit. It is advisable to limit the subject line length to 80 characters or less. The mail body will also include full event information, including the source system, facility, priority and actual message text as well as any other information that came with this event. As there is no size limitation for message bodies, the body always contains the full message received (except otherwise configured – see below).

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

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

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

Mail Priority

| File Configuration fields: | nMailPriority

0 = low
1 = Default
2 = High |
|---|---|
| Description: | Here you can adjust the priority with which the mail will be sent. You can choose between "low", "normal" and "high" priority. With this you can give your setup some complexity, being able to send some events as "important" and others with less importance. |

Mail Message Format

File Configuration fields:	szSMTPBody
Description:	This is the format of the message body. Properties from the event can be included by using the Property Replacer. Please note that the message body is only sent if "Include Message/Event in Email Body" is checked.

Output Encoding

File Configuration fields:	nOutputEncoding
Description:	Determines the character encoding mode.

Include message / event in email body

File Configuration fields:	nIncludeMessage
Description:	This checkbox controls whether the Syslog message will be included in the message body or not. If left unchecked, it will not be included in the body. If checked, it will be sent.

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

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

This option is must useful together with a well-formatted subject line in non-legacy mode.

Use XML to Report

File Configuration fields:	nUseXMLtoReport
Description:	If checked, the received event will be included in XML format in the mail. If so, the event will include all information, like the original timestamp, the facility, priority etc. XML format is especially useful if the mail is sent to an automated system, which will then parse the message. If unchecked, just the plain text message will be included in the mail. This format is more readable for a human reader.

4.7.8 Forward Syslog Options

This dialog controls Syslog forwarding options.

Protocol Type

There are various ways to transmit syslog messages. In general, they can be sent via UDP, TCP or RFC 3195 RAW. Typically, syslog messages are received via UDP protocol, which is the default. UDP is understood by almost all servers, but doesn't guarantee transport. In plain words, this means that syslog messages sent via UDP can get lost if there is a network error, the network is congested or a device (like a router or switch) is out of buffer space. Typically, UDP works quite well. However, it should not be used if the loss of a limited number of messages is not acceptable.

TCP and RFC 3195 based syslog messages offer much greater reliabilty. RFC 3195 is a special standardized transfer mode. However, it has not receive any importance in practice. Servers are hard to find. As one of the very few, Adiscon products support RFC 3195 also in the server implementations. Due to limited deployment, however, RFC 3195 is very little prooven in practice. Thus we advise against using RFC 3195 mode if not strictly necessary (e.g. part of your requirement sheet).

TCP mode comes in three flavours. This stems back to the fact that transmission of syslog messages via plain TCP is not yet officially standardized (and it is doubtful if it ever will be). However, it is the most relevant and most widely implemented reliable transmission mode for syslog. It is a kind of unwritten industry standard. We support three different transmission modes offering the greatest compatibility with all existing implementations. The mode "TCP (one message per connection)" is a compatibility mode for Adiscon servers that are older than roughly June 2006. It may also be required for some other vendors. We recommend not to use this setting, except when needed. "TCP (persistent connection)" sends multiple messages over a single connection, which is

held open for an extended period of time. This mode is compatible with almost all implementations and offers good performance. Some issues may occur if control characters are present in the syslog message, which typically should not happen. The mode "TCP (octet-count based framing)" implements algorithms of an upcoming (but not yet finalized) IETF standard. It also uses a persistent connection. This mode is reliable and also deals with embedded control characters very well. However, there is only a limited set of receivers known to support it. As of this writing (January 2007), there were no non-Adiscon receivers supporting that mode. We expect progress once the IETF standard is officially out.

As a rule of thumb, we recommend to use "TCP (octet-count based framing)" if you are dealing only with (newer) Adiscon products. Otherwise, "TCP (persistent connection)" is probably the best choice. If you select one of these options, you can also select a timeout. The connection is torn down if that timeout expires without a message beeing sent. We recommend to use the default of 30 minutes, which should be more than efficient. If an installation only occasionally sends messages, it could be useful to use a lower timeout value. This will free up connection slots on the server machine.

Syslog Target Options

Forward Syslog Properties

Syslog Send mode

File Configuration fields:	nSendMode
Description:	The Sendmode has been added since 2018 into all products supporting the forward syslog action. There are two options are available. **Use single syslog server with optional backup server** This is the classic syslog send mode which uses a primary syslog server and a secondary backup syslog server if configured. **Use round robin (multiple syslog servers)** This new method allows you to configure multiple targets that will be used one by one after a configured amount of messages has been send to each target.

Syslog Server (Syslog Send mode)

File Configuration fields:	szSyslogSendServer
Description:	This is the name or IP address of the system to which Syslog messages should be sent to. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Syslog Port (Syslog Send mode)

File Configuration fields:	nSyslogSendPort
Description:	The remote port on the Syslog server to report to. If in doubt, please leave it at the default value of 514, which is typically the Syslog port. Different values are only required for special setups, for example in security sensitive areas. Set the port to 0 to use the system-supplied default value (which defaults to 514 on allmost all systems). Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions.

Use this backup syslog server if first one fails

File Configuration fields:	nEnableBackupServer
Description:	The backup server is automatically used if the connection to the primary server fails. The primary server is automatically retried when the next Syslog session is opened. This option is only available when using TCP syslog.

Amount of messages send to each syslog server before load balancing

File Configuration fields:	nRoundRobinMsgCount

Description:	When using round robin mode, this is the amount of messages to be send to each configured syslog server.

Syslog Servers

Syslog Server (Round robin mode)

File Configuration fields:	szSyslogServer_[n]
Description:	This is the name or IP address of the system to which Syslog messages should be sent to. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Syslog Port (Round robin mode)

File Configuration fields:	nSyslogPort_[n]
Description:	The remote port on the Syslog server to report to. If in doubt, please leave it at the default value of 514, which is typically the Syslog port. Different values are only required for special setups, for example in security sensitive areas. Set the port to 0 to use the system-supplied default value (which defaults to 514 on allmost all systems). Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions.

Syslog Message Options

Syslog Message Options

Syslog processing

File Configuration fields:	bProcessDuringRelay 0 = Disable processing 1 = RFC3164 Header 2 = RFC5424 Header 3 = Custom Syslog Header
Description:	With this settings you can assign how your syslog messages will be processed. For processing syslog you can choose out of four different options. You can use RFC3164 or RFC5424 (recommended) which is the current syslog standard, you are able to customize the syslog header or you do not process your syslog and forwards it as it is.

Custom Header Format

File Configuration fields:	szCustomSyslogHeader
Description:	In this field you can specify the contents of your syslog header. This option is only available when you choose "Use Custom Syslog Header" in the Syslog Processing menu. The contents can be either a fixed message part which you can write into the

	field yourself or you use properties as dynamic content. By default the Header field is filled with the content of the RFC 5424 header. **Please note that the header content of the Header field can be configured. Event properties are described in the property replacer section.**

Output Encoding

File Configuration fields:	nOutputEncoding
Description:	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.

Include UTF8 BOM in message

File Configuration fields:	nProtocolType
Description:	If enabled (default), the UTF8 BOM code will be prepended to the output message if you are using UTF8 Output encoding. If the syslog receiver cannot handle and remove the UTF8 BOM you can disabled this option.

Use XML to Report

File Configuration fields:	bReportInXML
Description:	If this option is checked, the forwarded Syslog message is a complete XML-formatted information record. It includes additional information like timestamps or originating system in an easy to parse format. The XML formatted message is especially useful if the receiving system is capable of parsing XML data. However, it might also be useful to a human reader as it includes additional information that cannot be transferred otherwise.

Forward as MW Agent XML Representation Code

File Configuration fields:	nForwardIUT
Description:	MonitorWare supports a specific XML-Representation of the event. If it is checked, that XML representation is used. It provides additional information (like informationunit type, original source system, reception time & many more) but is harder to read by a human. At the same time, it is obviously easier to parse. Please note that this option is only "experimental" and is not an official standard. Use CEE enhanced Syslog Format If enabled, the new CEE enhanced Syslog format will be used (work in progress). All useful properties will be included in a JSON Stream. The message itself can be

included as well, see the "Include message property in CEE Format" option. Here is a sample how the format looks like for a security Eventlog message:

@cee: {"source": "machine.local", "nteventlogtype": "Security", "sourceproc": "Microsoft-Windows-Security-Auditing", "id": "4648", "categoryid": "12544", "category": "12544", "keywordid": "0x8020000000000000", "user": "N\\A", "SubjectUserSid": "S-1-5-11-222222222-333333333-4444444444-5555", "SubjectUserName": "User", "SubjectDomainName": "DOMAIN", "SubjectLogonId": "0x5efdd", "LogonGuid": "{00000000-0000-0000-0000-000000000000}", "TargetUserName": "Administrator", "TargetDomainName": " DOMAIN ", "TargetLogonGuid": "{00000000-0000-0000-0000-000000000000}", "TargetServerName": "servername", "TargetInfo": " servername ", "ProcessId": "0x76c", "ProcessName": "C:\\Windows\\System32\\spoolsv.exe", "IpAddress": "-", "IpPort": "-", "catname": "Logon", "keyword": "Audit Success", "level": "Information", }

Additionaly to this format you can set Include message property in CEE Format

If enabled, the message itself will be included in the JSON Stream as property. Disable this option if you do not want the message itself in the CEE Format.

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

Use CEE enhanced Syslog Format

File Configuration fields:	nReportInJSON
Description:	If enabled, the Syslog message will be converted into a valid CEE JSON String.

Message Format

File Configuration fields:	szMessageFormat
Description:	The custom format lets you decide how the content of a syslog message looks like. You can use properties to insert content dynamically or have fixed messages that appear in every message. Event properties are described in the property replacer section.

Add Syslog Source when forwarding to other Syslog servers

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

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

Use zLib Compression to compress the data

File Configuration fields:	nUseCompression
Description:	With this option you can set the grade of compression for your syslog messages. For more information please read the note at the bottom of this page.

Compression Level

| File Configuration fields: | nCompressionLevel

1 = Best Speed
3 = Low Compression
6 = Normal Compression
9 = Best Compression |
|---|---|
| Description: | With this option you can set the grade of compression for your syslog messages. For more information please read the note at the bottom of this page.

Note on Using Syslog Compression

Compressing syslog messages is an experimental feature. There is only a very limited set of receivers who is able to understand that format. Turning on compression can save valuable bandwidth in low-bandwidth environments. Depending on the message, the saving can be anything from no saving at all to about a reduction in half. The best savings ratios have been seen with Windows event log records in XML format. In this case, 50% or even a bit more can be saved. Very small messages do not compress at all. Typical syslog traffic in non-xml format is expected to compress around 10 to 25%.

Please note that compressesion over TCP connections requires a special transfer mode. This mode bases on an upcoming IETF standard (syslog-transport-tls) that is not yet finalized. That transfer mode is highly experimental in itself. As a result, future releases of our product might not be able to work with the current implementation. So there is a chance that you need to exchange all parts of the syslog/TCP system in future releases. Backwards compatibility can not be guaranteed.

Besides the fact that the mechanisms behind compression are experimental, the feature itself is solid. |

Overwrite Syslog Properties

Syslog Facility

File Configuration fields:	nSyslogFacility

Description:	When configured, will overwrite the Syslog Facility with the configured value.

Syslog Priority

File Configuration fields:	nSyslogPriority
Description:	When configured, will overwrite the Syslog Priority with the configured value.

SSL/TLS related Options

SSL/TLS related Options

Enable SSL / TLS Encryption

File Configuration fields:	nUseSSL
Description:	If this option is enabled, the action will not be able to talk to a NON-SSL secured server. The method used for encryption is compatible to RFC5425 (Transport Layer Security (TLS) Transport Mapping for Syslog).

TLS Mode

File Configuration fields:	nTLSMode
Description:	**Anonymous Authentication** Default option. This means that a default certificate will be used. **Use Certificate** If this option is enable, you can specify your own certificate. For further authentication solutions, you will need to create your own certificates using OpenSSL Tools for example.

Select common CA PEM

File Configuration fields:	szTLSCAFile
Description:	Select the certificate from the common Certificate Authority (CA). The syslog receiver should use the same CA.

Select Certificate PEM

File Configuration fields:	szTLSCertFile
Description:	Select the client certificate (PEM Format).

Select Key PEM

File Configuration fields:	szTLSKeyFile
Description:	Select the keyfile for the client certificate (PEM Format).

Allow SSL v3

File Configuration fields:	nTLSAllowSSLv3
Description:	This option enables insecure protocol method SSLv3. We recommend NOT enabling this option as SSLv3 is considered broken.

Allow SSL v1.0

File Configuration fields:	nTLSAllowTLS10
Description:	This option enables insecure protocol method TLSv1. We recommend NOT enabling this option as TLSv1 is considered broken.

Allow SSL v1.1

File Configuration	nTLSAllowTLS11

fields:	
Description:	This option enables protocol method TLS1.1 which is enabled by default.

Allow SSL v1.2

File Configuration fields:	nTLSAllowTLS12
Description:	This option enables protocol method TLS1.2 which is enabled by default.

Use OpenSSL configuration commands

File Configuration fields:	nTLSUseConfigurationCommands
Description:	By enabling this option, you can set OpenSSL configuration commands directly. For more information's on available configuration parameters for each command type, visit this page: https://www.openssl.org/docs/man1.0.2/ssl/SSL_CONF_cmd.html We allow to the set the following OpenSSL configuration commands in the configuration commands list: • CipherString: Sets the allowed/dissallowed used Ciphers. Setting this value will **OVERWRITE** the internal default ciphers. • SignatureAlgorithms: This sets the supported signature algorithms for TLS v1.2. • Curves: This sets the supported elliptic curves. • Protocol: Sets the supported versions of the SSL or TLS protocol. This will **OVERWRITE** the Allow SSL options from above! • Options: The value argument is a comma separated list of various flags to set. When setting advanced configuration commands, we highly recommend to enable debug logging and review it after changes have been made. An error will be logged in the debug logfile if a configuration command cannot be processed successfully.

TCP related Options

TCP related Options

When using TCP-based syslog forwarding, you have the additional option to use the diskqueue. Whenever a connection to a remote syslog server fails, the action starts caching the syslog messages into temporary files. The folder for these files can be configured. The filenames are generated using a unique GUID which is automatically generated for each Action, thus enabling you to use this feature in multiple Actions. Once the syslog server becomes available again, the cached messages are being sent automatically. If you restart the Service while the Syslog Cache was active, it cannot be checked during service startup if the syslog server is available now. Once the action is called again, the check is done and if the syslog server is available, the messages are being sent. The size of this cache is only limited by the disk size. Files are splitted by 10MB by default, but this can also be configured. The maximum supported file size is 2GB.

Please Note: This option is not available for UDP or RFC3195.

Session Timeout

File Configuration fields:	nTimeoutValue
Description:	Timeout value for TCP persistent and octet-count based framing connections.

Use Diskqueue if connection to Syslog Server fails

File Configuration fields:	nUseDiscQueue
Description:	Enable diskqueuing syslog messages after unexpected connection loss.

Split files if this size is reached

File Configuration fields:	nDiskQueueMaxFileSize

Description:	Files will be split until they reach the configured size in bytes. The maximum support file size is 2147483648 bytes (2GB).

Diskqueue Directory

File Configuration fields:	szDiskQueueDirectory
Description:	The directory where the queue files will be generated in. The queuefiles will be generated with a dynamic UUID bound to the action configuration.

Waittime between connection tries

File Configuration fields:	nDiskCacheWait
Description:	The minimum waittime until the Syslog Action retries to establish a connection to the syslog server after failure.

Overrun Prevention Delay (ms)

File Configuration fields:	nPreventOverrunDelay
Description:	When the Action is processing syslog cache files, an overrun prevention delay can be added to avoid flooding the target syslog server.

Double wait time after each retry

File Configuration fields:	bCacheWaittimeDoubling
Description:	If enabled, the configured waittime is doubled after each try.

Limit wait time doubling to

File Configuration fields:	nCacheWaittimeDoublingTimes
Description:	How often the waittime is doubled after a failed connection try.

Enable random wait time delay

File Configuration fields:	bCacheRandomDelay
Description:	If enabled, a some random time will be added into the waittime delay. When using many syslog senders, this can avoid that all senders start sending cached syslog data to the syslog server at the same time.

Maximum random delay

File Configuration fields:	nCacheRandomDelayTime
Description:	maximum random delay time that will be added to the configured waittime if **Enable random wait time delay** is enabled.

UDP related Options

UDP related Options

Enable IP Spoofing for the UDP Protocol

File Configuration fields:	nSpoofIPAddress
Description:	This option enables you to spoof the IP Address when sending Syslog messages over UDP. Some notes regarding the support of IP Spoofing. It is only supported the UDP Protocol and IPv4. IPv6 is not possible yet. Due system limitations introduced by Microsoft, **IP Spoofing is only possible on Windows Server 2003, 2008 or higher. It is NOT possible in Windows XP, VISTA, 7 or higher.** For more information see the Microsoft explanation. Also please note that most routers and gateways may drop network packages with spoofed IP Addresses, so it may only work in local networks.

Fixed IP or single property

File Configuration fields:	szSpoofedIPAddress
Description:	You can either use a static IP Address or a property. When using a property, the IP Address is tried to be resolved from the content of the property. For example by default the %source% property is used. If the name in this property cannot be resolved to an IP Address, the default local IP Address will be used.

4.7.9   Forward SETP Options

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

Send SETP Dialog

Servername

File Configuration fields:	szServer
Description:	The MonitorWare Agent sends SETP to the server / listener under this name. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Default SETP Port

File Configuration fields:	nMIAPSendPort
Description:	The Send SETP sends outgoing requests on this port. The default value is 5432. Set the port to 0 to use the system-supplied default value (which defaults to 5432 if not modified by a system administrator). Instead of the port number, a service name can be used. If so, that name is looked up via the socket service database functions. The lookup is for protocol TCP. **Please note: The SETP port configured here must match the port configured at the listener side (i.e. MonitorWare Agent 3.0 or WinSyslog Enterprise edition). If they do not match, a Send SETP session cannot be initiated. The rule engine will log this to the Windows Event Log.**

Enable SSL / TLS Encryption.

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

Use zLib Compression to compress the data

File Configuration	nZlibComp

fields:	
Description:	It enables zLib compression support. Note that the SETP receiver must have zLib Compression support and enabled, otherwise it does not work.

Compression Level

File Configuration fields:	nCompLevel 1 = Best Speed 3 = Low Compression 6 = Normal Compression 9 = Best Compression
Description:	Higher level results in better compression but slower performance.

Session Timeout

File Configuration fields:	nTimeOutSession
Description:	The maximum time a session to a SETP server is to be kept open.

Connection Timeout

File Configuration fields:	nConnectTimeOut
Description:	Maximum time a connection can take to connect or disconnect.

Send / Receive Timeout

File Configuration fields:	nSendRecvTimeOut
Description:	When sending or receiving data, this timeout applies. **Please note:** If this option is enabled, this action is not be able to connect to NON-SSL SETP servers.

4.7.10 Send MSQueue

In order to use this Action, the "Microsoft Message Queue (MSMQ) Server" needs to be installed. This Action can be used to send a message into the Microsoft Message Queue.

Send MSQueue Properties

Server Computername/IP

File Configuration fields:	szComputerName
Description:	Sets the computername or IP which contains the MSQueue you want to query. You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Queuename

File Configuration fields:	szQueueName
Description:	Specify the Queuename into which you want to write.

Queue Priority

File Configuration fields:	nMessagePriority
Description:	Configure or set the priority property here.

Queue Message Label

File Configuration fields:	szQueueLable
Description:	Sets the Label text of a queue item.

Queue Message Body

File Configuration fields:	szQueueBody
Description:	The text here will be set to the body of a queue item.

4.7.11 Net Send

This dialog controls the net send options.

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

Net Send Dialog

Target

File Configuration fields:	szTarget
Description:	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). You can either use an IPv4, an IPv6 Address or a Hostname that resolves to an IPv4 or IPv6 Address.

Message to Send

File Configuration fields:	szMessage
Description:	This is the message that is sent to the intended target. **Please note that the message content of the Message to send field can now be configured. Event properties are described in the property replacer section.**

4.7.12 Start Program

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

Start Process Dialog

Command to execute

File Configuration fields:	szCommand
Description:	This is the path of actual program file to be executed. This can be the path of any valid executable file. A relative file name can be specified if it can be found via the operating system default search path.

Use legacy parameter processing

File Configuration fields:	nUseLegacyProcessing
Description:	When enabled, old style parameter processing is used. Otherwise all properties can be used.

Parameters

File Configuration fields:	szParameters
Description:	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!

Sync Timeout

File Configuration fields:	nSyncTimeOut
Description:	Time Out option is under Sync. Processing. When a program is executed, the service waits for it to finish before it carries on further actions. This is needed in order to ensure that all actions are carried out in the correct sequence.
	The external program should only run for a limited amount of time. If it would block for some reason, the agent would be prevented from carrying out any further processing. As such, a timeout value must be specified. If the program still runs after the configured timeout, the rule engine cancels it, flags the action as unsuccessful and then carries on with processing.
	Important: Even though the timeout value can be as high as 30 seconds, we strongly recommend limiting the run time of external program to below 5 seconds. Otherwise, they could affect the overall performance too much. If the average run time is 5 seconds, the default timeout of 10 seconds ensures that the program can finish even when there is high system activity.
	For performance reasons, we also strongly recommend to use the "Start Program" action only for rules that apply relatively seldom.

4.7.13 Play Sound

This action allows you to play a sound file. Since Windows VISTA/2008/7, Microsoft has disabled any interaction between a system service and the user desktop. This includes playing sounds as well. So if you want to use the Play Sound Action on any of this Windows Version, you will need to run the service in console mode (From command prompt with the -r option).

Name: Play Sound ⊘ Enabled ▾ | ⊕ Comments ☑ Settings | 💾 Confirm 🔄 Reset ❓ |

Filename of the soundfile		Browse
Playcount (How often the file is played)	1	▾
Delay between the sound plays (ms)	2000	▾ milliseconds

Play Sound Dialog

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

However there is a work around if you want to use Play Sound Action for a second sound card!

Filename of the Soundfile

File Configuration fields:	szFilename
Description:	Please enter the name of the sound file to play. **This must be a .WAV file**, other formats (like MP3) are **not** supported. While in theory it is possible that the sound file resides on a different machine, we highly recommend using files on the local machine only. Using remote files is officially not supported (but currently doable if you are prepared for some extra effort in getting this going). If the file can either not be found or is not in a valid format, a system beep is emitted instead (this should - by API definition - be possible on any system).

Playcount

File Configuration fields:	nCount
Description:	This specifies how many times the file is played. It can be re-played up to a hundered times. Please note: Playing sounds is performance intense and MonitorWarthe Service will block all other actions while sounds are being played. As such, we recommend to limit the duration and repeat count of sounds played.

Delay between Plays

File Configuration fields:	nDelay
Description:	If multiple repeats are specified, this is the amount of time that is to be waited for between each individual play.

4.7.14 Send to Communications Port

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

Send to Communications Port Options

Timeout Limit

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

Send message to this communication port

File Configuration fields:	szPortName
Description:	Specify the port to which your device is being attached. Typically, this should be one of the COMx: ports. The listbox shows all ports that can be found on your local machine. You may need to adjust this to a different value, if you are configuring a remote machine. 1. MSFAX 2. COM1 3. COM2 4. COM3 5. COM4 6. FILE 7. LPT1 8. LPT2 9. LPT3

10.	AVMISDN1
11.	AVMISDN2
12.	AVMISDN3
13.	AVMISDN4
14.	AVMISDN5
15.	AVMISDN6
16.	AVMISDN7
17.	AVMISDN8
18.	AVMISDN9

Port Settings

File Configuration fields:	szPortSettings
Description:	Use those settings that your device expects. Please consult your device manual if in doubt.

Bits per Seconds

File Configuration fields:	nBps
Description:	Bits per second can be 110 and go up to 256000, by default 57600 is selected.

Databits

File Configuration fields:	nDatabits
Description:	Databits defines that how many bits you want to send and receive to the communication port.

Parity

File Configuration fields:	nParity
Description:	With Parity you can configure the Parity scheme to be used. This can be one of the following values: 1. Even 2. Mark 3. No parity 4. Odd 5. Space

Stop bits

File Configuration fields:	nStopbits

Description:	You can configure the number of stop bits to be used. This can be one of the following values:
	1.　　　　　　　1 stop bit 2.　　　　　　　1.5 stop bits 3.　　　　　　　2 stop bits

DTR Control Flow

File Configuration fields:	nDtsControl
Description:	DTR (data-terminal-ready) flow control. This member can be one of the following values: 1.　　　　　　DTR_CONTROL_DISABLE - Disables the DTR line when the device is opened and leaves it disabled. 2.　　　　　　DTR_CONTROL_ENABLE - Enables the DTR line when the device is opened and leaves it on. 3.　　　DTR_CONTROL_HANDSHAKE - Enables DTR handshaking.

RTS Control Flow

File Configuration fields:	nRtsControl
Description:	RTS (request-to-send) flow control. This member can be one of the following values: 1. RTS_CONTROL_DISABLE - Disables the RTS line when the device is opened and leaves it disabled. 2. RTS_CONTROL_ENABLE - Enables the RTS line when the device is opened and leaves it on. 3. RTS_CONTROL_HANDSHAKE - Enables RTS handshaking. The driver raises the RTS line when the "type-ahead" (input) buffer is less than one-half full and lowers the RTS line when the buffer is more than three-quarters full. 4. RTS_CONTROL_TOGGLE - Specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be low.

Message to Send

File Configuration fields:	szMessage
Description:	This is the message that is to be send to the device. You can enter text plainly and you can also include all properties from the current event. For example, if you have a serial audit printer and you would just plainly like to log arrived messages to that printer, you could use the string "%msg%%$CRLF%" to write the actual message arrived plus a CRLF (line feed) sequence to the printer. **Please note that the message content of the Message field can now be configured. Event properties are described in the property replacer section.**

4.7.15 Set Status

This dialog controls the set status options.

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

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

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

Set Status Dialog

Status Variable Name

File Configuration fields:	szPropertyName
Description:	Enter the Property name. That name will from now on be used inside the rule base. More precisely, it will be used in the filter conditions and actions. **Please note that the field content can be configured with Event properties are described in the property replacer section.**

Status Variable Value

File Configuration fields:	szPropertyValue
Description:	The value to be assigned to the property. Any valid property type value can be entered. **Please note that the field content can be configured with Event properties are described in the property replacer section.**

4.7.16 Set Property

You can set every property and custom properties using this action.

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

Set Property Dialog

Select Property Type

File Configuration fields:	szPropertyValue
Description:	Select the property type to be changed. The list box contains all properties that can be changed. By default it is set to nothing. **Please note that the field content can be configured with Event properties are described in the property replacer section.**

Set Property Value

File Configuration fields:	szPropertyType
Description:	The new value to be assigned to the property. Any valid property value can be entered. Please use the "Insert Button". 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. **Please note that the field content can be configured with Event properties are described in the property replacer section.**

4.7.17 Call RuleSet

The dialog shown below controls the Call RuleSet options.

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

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

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

Call Ruleset Dialog

Ruleset to Call

File Configuration fields:	szRuleSet
Description:	Select the Ruleset to be called. **Note: Call RuleSet stays disabled until you have more then "One" RuleSet!**

4.7.18 Discard

A Discard Action immediately destroys the current Information Unit and any action of any rule that has been defined after the Discard action execution. When this action is been selected then no dialog appears as nothing needs to be configured for this.

5   Getting Help

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

Please note that all options (except priority support) are also open to evaluating customers. So do not hesitate to try them. Help is available in English and German language. Our local resellers may provide local language support. Please check with them.

Frequently asked Questions

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

Support Forum

Share questions and answers with your peers! The forum is also monitored by Adiscon support staff. To access the forum, point your browser at http://forum.adiscon.com/forum,3.html.

Customer Support System

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

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

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

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

In this case, please remember that notification emails may also get lost, so it is a good idea to check your ticket for status updates from time to time.

EventReporter website

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

Email

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

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

Online Seminars

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

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

Phone

Phone support is limited to those who purchased support incidents. If you are interested in doing so, please email info@adiscon.com **for further details.**

Software Maintenance

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

Click here to learn more about UpgradeInsurance.

Non-Technical Questions

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

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

Product Updates

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

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

6   Purchasing EventReporter

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

The License

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

Pricing & Ordering

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

If you would like to receive assistance with your order or need a quote, please contact info@adiscon.com.

7   Reference

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

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

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

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

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

Properties Available	MonitorWare Agent	WinSyslog	EventReporter
Standard Property	Yes	Yes	Yes
Windows Event Log Properties	Yes		Yes
MonitorWare Echo Request			Yes
Syslog Message Properties	Yes	Yes	
Disk Space Monitor	Yes		
File Monitor	Yes		
Windows Service Monitor	Yes		Yes
Ping Probe	Yes		
Port Probe	Yes		
Database Monitor	Yes		
Serial Port Monitor	Yes		
MonitorWare Echo Request	Yes		
System Properties	Yes	Yes	Yes
Custom Properties	Yes	Yes	Yes
NNTP Probe	Yes		
HTTP Probe	Yes		
FTP Probe	Yes		
SMTP Probe	Yes		
POP3 Probe	Yes		

7.2 Event Properties

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

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

7.2.1 Acessing Properties

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

The property replacer provides very powerful ways to acces the properties: they can not only be accessed as one full property. They can also be acceessed as substrings and even be reformatted. As such, the property replacer provides a specifc syntax to access properties:

%property:fromPos:toPos:options%

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

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

Within the properties, all time is based on UTC regardless if your preferred time is UTC or localtime. So if you want to display localtime instead of UTC, you have to use the following syntax: %variable:::localtime%

7.2.1.1 Property

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

A property is either an event property, a custom property, a dynamic property or a system property.

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

7.2.1.2 FromPos

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

Fixed Character position

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

Search Pattern

A search pattern is specified as follows:

/<search-pattern>/<options>

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

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

Please note that a slash inside the search pattern will terminate the search field. So pure slashes can not be used. However, they can be escaped by prefixing them with a backslash (\). The same applies to the '?' character. For example, if you intend to search for "http://" inside a search pattern, you must use the following search string: "/http:\/\//".

Default Value

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

7.2.1.3   ToPos

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

Absolute Position

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

Relative Position

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

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

Search Pattern

Search pattern support is similar to search pattern support in **FromPos**.

A search pattern is specified as follows:

> /<search-pattern>/<options>

If a search pattern is specified, the property value is examined and the first occurence of <search-pattern> is detected. The search is only carried out in the string that follows FromPos. If the string is not found, nothing is returned. If it is found, the position where the pattern is found is the ending postion or, if the option "$" is specified, the position immediately after the pattern.

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

Please note that a slash inside the search pattern will terminate the search field. So pure slashes can not be used. However, they can be escaped by prefixing them with a backslash (\). The same

applies to the '?' character. For example, if you intend to search for "http://" inside a search pattern, you must use the following search string: "/http:\//".

Search Example

A common use case is to combine searches in **ToPos** and **FromPos** to extract a substring that is delimited by two other strings. To do so, use search patterns in both fields. An example is as follows: assume a device might generate message in the form "... error XXX occurred..." where "..." represents additional message text and XXX the actual error cause. You would like to extract the phrase "error XXX occurred". To do so, use the following property replacer syntax:

%msg:/error/:/occured/$/%

Please note that the FromPos is used without the $-option, while in ToPos it is used. If it hadn't been used in ToPos, only the part "error XXX " would have been extracted, as the ToPos would point to the last character before the search string.

Similarily, if only " XXX " should be extracted, the following syntax might be used:

%msg:/error/$:/occured/%

If you would also like to remove the spaces (resulting in just "XXX"), you must include them into the search strings:

%msg:/error /:/ occured/$/%

Default

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

7.2.1.4 Options

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

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

lowercase	All characters in the resulting property extract will be converted to lower case.
uppercase	All characters in the resulting property extract will be converted to upper case.
uxTimeStamp	This is a special switch for date conversions. It only works if the extracted property value is an ISO-like timestamp (YYYY-MM-DD HH:MM:SS). If so, it will be converted to a Unix-like ctime() timestamp. If the extracted property value is not an ISO-like timestamp, no conversion happens.
uxLocalTimeStamp	This is the same as uxTimeStamp, but with local time instead of GMT.
date-rfc3339	This option is for replacing the normal date format with the date format from RFC3339.

date-rfc3164	This option is for replacing the normal date format with the date format from RFC3164.
escapecc	Control characters* in property are replaced by the sequence ##hex-val##, where hex-val is the hexadecimal value of the control character (at least two digits, may be more).
spacecc	Control characters* in the property are replaced by spaces. This option is most useful when a message contains control characters (e.g. a Windows Event Log Message) and should be written to a log file.
compressspace	Compresses multiple consecutive space characters into a single one. The result is a string where all words are separated by just single spaces. To also compress control characters, use the compressspace and spacecc options together (e.g. '%msg:::spacecc,compressspace%').
	Please note that space compression happens on the final substring. So if you use the FromPos and ToPos capabilities, the substring is extracted first and then the space compression applied. For example, you may have the msg string "1 2". There are two space between 1 and 2. Thus, the property replacer expression
	%msg:1:3:compressspace%
	will lead to "1 " ('1' followed by two spaces). If you intend to receive "1 2" ('1' followed by one space, followed by '2'), you need to use
	%msg:1:4:compressspace%
	or
	%msg:1:/2/$:compressspace%
	In the second case, the exact length of the uncompressed string is not known, thus a search is used in "ToPos" to obtain it. The result is then space-compressed.
compsp	Exactly the same as **compressspace**, just an abbreviated form for those that like it brief.
csv	For example %variable:::csv%. This option will create a valid CSV string, for example a string like this this is a "test"! becomes this "this is a ""test""!" where quotes are replaced with double quotes.
cef	Convert string content into valid McAfee CEF Format. This means that = will be replaced with \= and \ will be replaced with \\.
convgermuml	Converts German Umlaut characters to their official replacement sequence (e.g. "ö" --> "oe")
localtime	Now you can print the Time with localtime format by using %variable:::localtime%
nomatchblank	If this is used, the Property Replacer will return an empty string if the FromPos or ToPos is not found.
replacepercent	This option replaces all % occurrences with a double %%, which is needed for the property replacer engine in case that a string is reprocessed. This is needed because the percent sign is a special character for the property replacer. Once the property is processed, the double %% become automatically one %.
toipv4address	Property string will be converted into IPv4 Address format if possible.
toipv6address	Property string will be converted into IPv6 Address format if possible.

* = control characters like e.g. carriage return, line feed, tab, ...

Important: All option values are case-sensitive. So "uxTimeStamp" works while "uxtimestamp" is an invalid option!

7.2.1.5 Examples

Simple Examples

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

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

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

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

More complex Examples

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

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

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

If you would like to have everything between the string "FROM" and "TO" including NONE of the both searchstrings:
%msg:/FROM/$:/TO/%

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

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

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

Please see also the focussed sample in the ToPos description.

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

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

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

7.2.2   System Properties

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

$CRLF	A Windows newline sequence consisting in the characters CR and LF. If you just need CR, you can use %$CRLF:1:1% and if you need use LF you can use %$CRLF:2:2%
$TAB	An US-ASCII horizontal tab (HT, 0x09) character
$HT	same as $TAB
$CR	A single US-ASCII CR character (shortcut for %$CRLF:1:1%)
$LF	A single US-ASCII LF character (shortcut for %$CRLF:2:2%)
$xNN	A single character, whoms value (in hexadecimal) is given by NN. NN **must** be two hexadecimal digits - a leading zero must be used if a value below 16 is to be represented. The value 0 (%x00) is invalid and - if specified - replaced by the "?" character. As an example, $CR could also be expressed as %$x0d%. Please note that only **one** character can be represented. If you need to specify multiple characters, you need multiple $xNN sequences. An example may be $CRLF which could also be specified as %$x0d%%$x0a% (but **not** as %$x0d0a%).
$NOW	Contains the current date and time in the format: YYYY-MM-DD HH.MM.SS Please note that the time parts are delimited by '.' instead iof ':'. This makes the generated name directly suitable for file name generation. If you need just parts of the timestamp, please use the property replacer's substring functionality to obtain the desired part. Use **%$NOW:1:4%** to get the **year**, **%$NOW:6:7%** to get the **month**, ... **%$NOW:1:10%** to get the **full datestamp**, **%$NOW:12:20%** to get the **full timestamp**
$NEWUUID	Creates a new UUID (Universally Unique Identifiers), a unique 128-bit integer represented as a 32 digit hexadecimal number.

7.2.3 Custom Properties

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

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

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

7.2.4 Event-Specific Properties

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

Each event has a set of properties which in turn have values. For example, there is a property named "source" and it will always contain an indication of which system the event orginated on. Obviously, not every event source does support all properties. For example, a syslog message does not contain a Windows Windows Event ID - simply because there is no such thing as an event ID in syslog. So, depending on the type of event, it may contain different properties.

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

7.2.4.1 Standard Properties

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

msgPropertyDescribed	A human-readable representation of the message text. While this is genereally available, the exact contents largely depends on the source of the information. For example, for a file monitor it contains the file line and for a syslog message it contains the parsed part of the syslog message.
source	The source system the message orginated from. This can be in various representations (e.g. IP address or DNS name) depending on

	configuration settings.
resource	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
CustomerID	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
SystemID	A user-assgined numerical value. Does not have any specific meaning. Primarily intended for quick filtering.
timereported	The time the originator tells us when this message was reported. For example, for syslog this is the timestamp from the syslog message (if not configured otherwise). Please note that timereported eventually is incorrect or inconsistent with local system time - as it depends on external devices, which may not be properly synchronized. For Windows Event Log events, timereported contains the timestamp from the event log record.
timegenerated	The time the event was recorded by the service. If messages are forwarded via SETP, this timestamp remains intact.
importance	Reserved for future use.
iut	Indicates the type of the event. Possible values are: 1- syslog message 2- hearbeat 3- Windows Event Log Entry 4- SNMP trap message 5- file monitor 8- ping probe 9- port probe 10- Windows service monitor 11- disk space momitor 12- database monitor 13- serial device monitor
iuvers	Version of the event record (info unit). This is a monitorware interal version identifier.

7.2.4.2 Windows Event Log Properties

id	Windows Event ID
severity	severity as indicated in the event log. This is represented in string form. Possible values are: [INF] - informational [AUS] - Audit Success [AUF] - Audit failure [WRN] - Warning [ERR] - Error [NON] - Success (called "NON" for historical reasons)
severityid	The serverity encoded as a numerical entity (like in Windows API)
sourceproc	The process that wrote the event record (called "source" in Windows event viewer).
category	The category ID from the Windows event log record. This is a numerical value. The actual value is depending on the event source.

catname	The category name from the Windows event log record. This is a string value. The actual value is depending on the event source. This value is a textual representation from the Category ID. This property could contain line feeds, which can be removed by activating the option "Remove Control Characters from String Parameters" in the advanced options of the EventLog Monitor Service.
user	The user name that was recorded in the Windows event log. This is "N\A" if no user was recorded.
NTEventLogType	The name of the Windows event log this event is from (for example "System" or "Security").
bdata	Windows event log records sometimes contain binary data. The event log monitor service can be set to include this binary data into the event, if it is present. If it is configured to do so, the binary data is put into the "bdata" property. Every byte of binary data is represented by two hexadecimal characters. Please note that it is likely for bdata **not** to be present. This is because the binary data is seldomly used and very performance-intense.

7.2.4.3 Windows Event Log V2 Properties

id	Windows Event ID
severity	severity as indicated in the event log. This is represented in string form. Possible values are: [INF] - informational [AUS] - Audit Success [AUF] - Audit failure [WRN] - Warning [ERR] - Error [NON] - Success (called "NON" for historical reasons)
severityid	The serverity encoded as a numerical entity (like in Windows API)
sourceproc	The process that wrote the event record (called "source" in Windows event viewer).
category	The category ID from the Windows event log record. This is a numerical value. The actual value is depending on the event source.
catname	The category name from the Windows event log record. This is a string value. The actual value is depending on the event source. This value is a textual representation from the Category ID. This property could contain line feeds, which can be removed by activating the option "Remove Control Characters from String Parameters" in the advanced options of the EventLog Monitor Service.
user	The user name that was recorded in the Windows event log. This is "N\A" if no user was recorded.
nteventlogtype	The name of the Windows event log this event is from (for example "System" or "Security").
channel	The channel property for event log entries, for classic Event logs they match the %nteventlogtype% property, for new event logs, they match the "Event Channel".
sourceraw	This contains the full internal name of the event source for new event logs, for classic event logs it contains the same value as in %

	sourceproc%.
level	Textual representation of the eventlog level (which is stored as number in %severityid%). This property is automatically localized by the system.
categoryid	Internal category id as number.
keyword	Textual representation of the event keyword. This property is automatically localized by the system.
user_sid	If available, contains the raw SID of the username (%user%) property.
recordnum	Contains the internal event record number. Please note that if the event log has been truncated before, it may not start with 0 or 1 but a higher number.

7.2.4.4   Syslog Message Properties

rawsyslogmsg	The message as it was received from the wire (unparsed).
syslogfacility	The facility of a syslog message. For non-syslog messages, the value is provided based on configuration. In essence, this is simply an integer value that can be used for quick filtering inside your rules.
syslogfacility_text	The facility of a syslog message. This property is automatically created by using the syslogfacility properly and set to these values: "Kernel", "User", "Mail", "Daemons", "Auth", "Syslog", "Lpr", "News", "UUCP", "Cron", "System0", "System1", "System2", "System3", "System4", "System5", "Local0", "Local1", "Local2", "Local3", "Local4", "Local5", "Local6", "Local7"
syslogpriority	The severity of a syslog message. For non-syslog messages, this should be a close approximation to what a syslog severity code means.
syslogpriority_text	The severity of a syslog message. This property is automatically created by using the syslogpriority properly and set to these values: "Emergency", "Alert", "Critical", "Error", "Warning", "Notice", "Informational", "Debug"
syslogtag	The syslog tag value, a short string. For non-syslog messages, this is provided based on configuration. In most cases, this is used for filtering.
syslogver	Contains the syslog version number which will be one or higher if a RFC 5424 valid message has been received, or 0 otherwise
syslogappname	Contains the appname header field, only available if the Syslog message was in RFC 5424 format. Otherwise, this field will be emulated by the %syslogtag% property
syslogprocid	Contains the procid header field, only set if the Syslog message was in RFC 5424 format.
syslogmsgid	Contains the msgid header field, only set if the Syslog message was in RFC 5424 format.
syslogstructdata	Contains the structdata header field (in raw format), only set if the Syslog message was in RFC 5424 format.
syslogprifac	Contains combined syslog facility and priority useful to build your own custom syslog headers

7.2.4.5   Disk Space Monitor

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

7.2.4.6   CPU/Memory Monitor

wmi_type	This variable is a string and can be one of the following variables: cpu_usage, mem_virtual_usage, mem_physical_usage, mem_total_usage
cpu_number	Number of the current checked CPU
cpu_load	The workload of the CPU as number, can be 0 to 100
mem_virtual_load	How much virtual memory is used (MB)
mem_virtual_max	How much virtual memory is max available (MB)
mem_virtual_free	How much virtual memory is free (MB)
mem_physical_load	How much physical memory is used (MB)
mem_physical_max	How much physical memory is max available (MB)
mem_physical_free	How much physical memory is free (MB)
mem_total_load	How much total(Virtual+Physical) memory is used (MB)
mem_total_max	How much total(Virtual+Physical) memory is max available (MB)
mem_total_free	How much total(Virtual+Physical) memory is free (MB)

7.2.4.7   File Monitor

genericfilename	The configured generic name of the file being reported.
generatedbasefilename	Contains the generated file name without the full path.

Special IIS LogFile Properties

The Logfile Fields in IIS Logfiles are customizeable, so there is no hardcoded command for their use.
The property-name depends on its name in the logfile. For example we take this Logfile:

#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2005-10-27 14:15:25
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status cs(User-Agent)
2005-10-27 14:15:16 127.0.0.1 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.1 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.2 - 192.168.0.1 443 POST /eCommerce/asdf.php
2005-10-27 14:15:16 127.0.0.2 - 192.168.0.1 443 POST /eCommerce/asdf.php

As you can see, in our sample the fields are named: date, time, c-ip, cs-username, s-ip, ... and so on.
To use them as a Property inside our MonitorWareProducts, just use the names from your Logfile and add a "p-" before it:

p-date	The Date on which the Event occurs

p-time	The Time on which the Event occurs
p-c-ip	The IP Adress of the User which accessed
p-cs-username	The Username of the User which accessed
p-s-ip	The Server IP
p-s-port	The Server Port
p-cs-method	The Client-Server Method (POST,GET)
p-cs-uri-stem	The accessed File including its path

7.2.4.8   Windows Service Monitor

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

7.2.4.9   Ping Probe

echostatus	Status returend for the echo request
	The status value can be one of the following:
	0 = IP_SUCCESS
	11002 = IP_DEST_NET_UNREACHABLE
	11003 = IP_DEST_HOST_UNREACHABLE
	11010 = IP_REQ_TIMED_OUT
	11013 = IP_TTL_EXPIRED_TRANSIT
	11016 = IP_SOURCE_QUENCH
	11018 = IP_BAD_DESTINATION
roundtriptime	Round trip time for the ping packet (if successful)

7.2.4.10   Port Probe

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

7.2.4.11   Database Monitor

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

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

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

select name, street, zip, city from addresses

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

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

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

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

7.2.4.12 Serial Monitor

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

7.2.4.13 MonitorWare Echo Request

responsestatus	The status of the echo request. Possible values: 0 - request failed (probed system not alive) 1 - request succeeded If the request failed, additional information can be found in the *msg* standard property.

7.2.4.14 FTP Probe

ftpstatus	The status of the connection.
ftprespmsg	The response of the connection.

7.2.4.15 IMAP Probe

imapstatus	The status of the connection.
imaprespmsg	The response of the connection.

7.2.4.16 NNTP Probe

nntpstatus	The status of the connection.
nntprespmsg	The response of the connection.

7.2.4.17 SMTP Probe

smtpstatus	The status of the connection.
smtprespmsg	The response of the connection.

7.2.4.18 POP3 Probe

pop3status	The status of the connection.
pop3respmsg	The response of the connection.

7.2.4.19 HTTP Probe

httpstatus	The status of the connection.
httprespmsg	The response of the connection.

7.3 Complex Filter Conditions

The rule engine uses complex filter conditions.

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

Example 1

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

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

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

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

In the filter dialog, this pseudo code looks as follows:
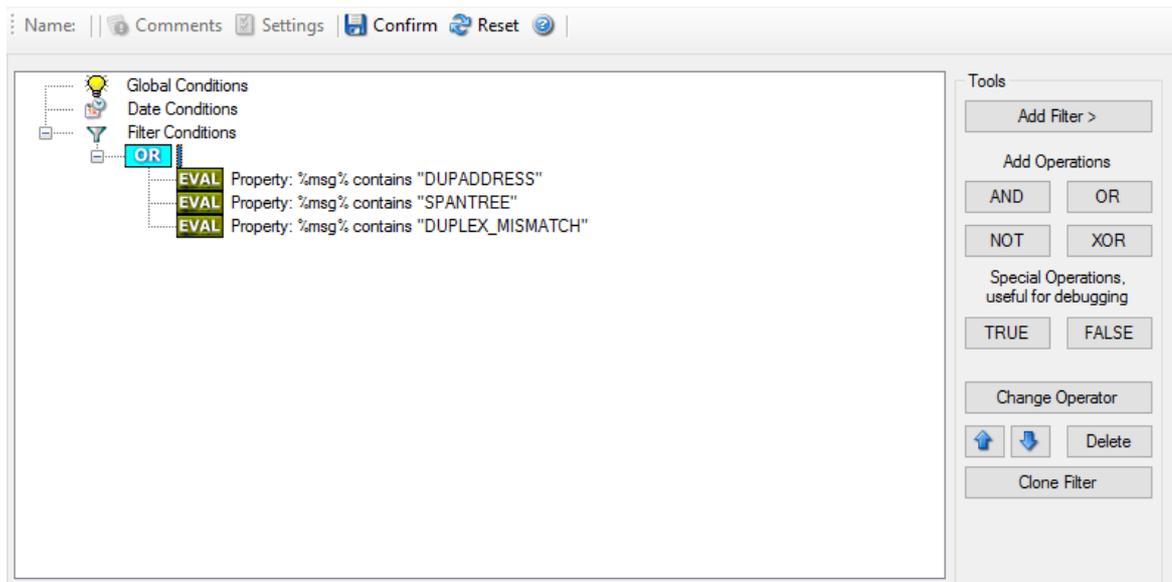
Figure 1 - Example 1

Example 2

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

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

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

In the filter dialog, this pseudo code looks as follows:
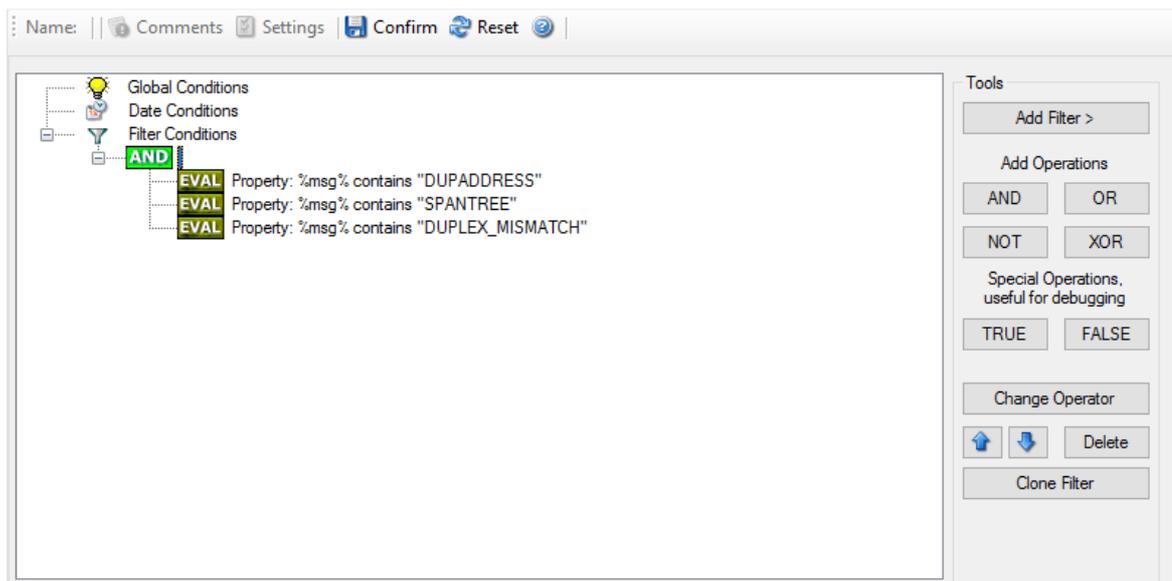
Figure 2 - Example 2

Example 3

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

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

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

```
If ((sourceSys = "192.0.2")
   And
    ((msg  = "DUPADDRES") Or (msg = "SPANTREE") Or (msg = "DUPLEX_MISMATCH))
   ) then
   execute action(s)
end if
```

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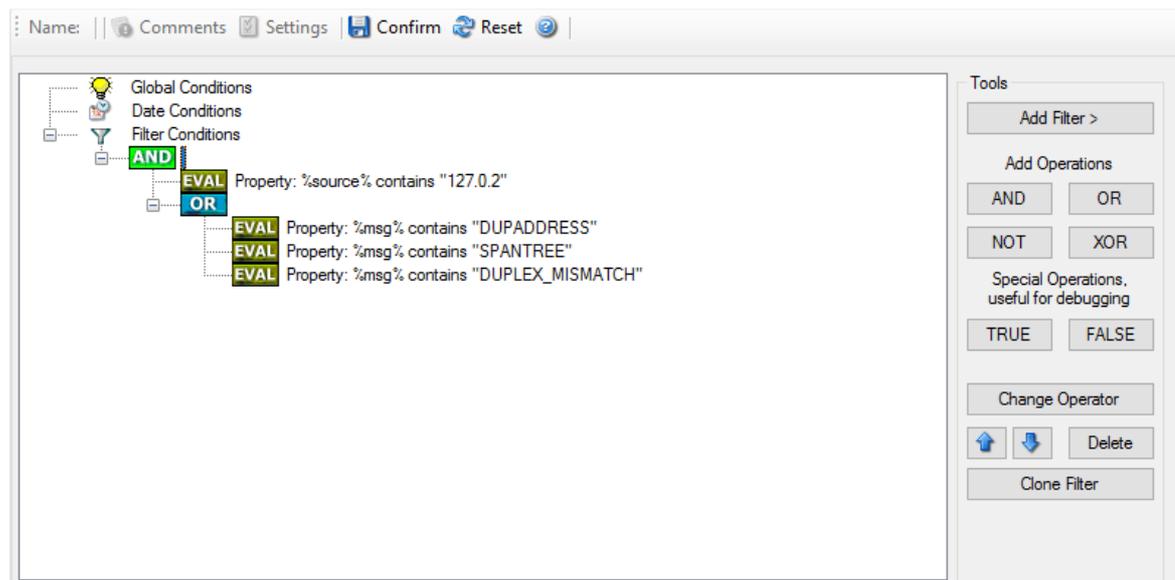
Figure 3 - Example 3

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

Real-World Examples

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

- Detecting Password Attacks under Windows

Example 4

In this example, the report is to be filtered in such a way that it shows information only in the case, if the time is greater then certain time with certain event source and one of two event ID's.

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

If (DeviceReportedTime is greater than {9:16:27} AND EventSource is equal to {Print} AND [EventID is equal to {10} OR EventID is equal to {18}]]

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

7.4   EventReporter Shortcut Keys

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

Press	To
CTRL+S	Save

```
CTRL+X                              Cut
CTRL+C                              Copy
CTRL+V                              Paste
CTRL+Z                              Undo
```

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

7.5 Command Line Switches

There are several command line switches available for using the agent via the command line.

-h	Show command line help
-v	Show version information
-i	Install service
-u	Remove (uninstall) service
-i "CustomServiceName"	Install service with a custom servicename
-u "CustomServiceName"	Uninstall a service with a custom servicename
-r	Run as console application
-r -o	Run ONCE as console application

If you install the service, you can start and stop the service with the "net start" and "net stop" commands. By using the "-r" switch, you run it only on the command line. When you close the command line, the program will stop working.

The "-v" switch gives you information about the version of the service.

You can import Adiscon Config Format (cfg) configuration files via the commandline as well. The syntax is quite easy. Simply execute the configuration client and append the name of the configuration file. This could look like this:

mwclient.exe example.cfg	Sample for MonitorWare Agent
CFGEvntSLog.exe example.cfg	Sample for EventReporter
WINSyslogClient.exe example.cfg	Sample for WinSyslog
RSyslogConfigClient.exe example.cfg	Sample for RSyslog Windows Agent

or

mwclient.exe "example.cfg"	Sample for MonitorWare Agent
CFGEvntSLog.exe "example.cfg"	Sample for EventReporter
WINSyslogClient.exe "example.cfg"	Sample for WinSyslog
RSyslogConfigClient.exe "example.cfg"	Sample for RSyslog Windows Agent

After this is executed, you will se the splash screen of the configuration client and then the import dialogue, which you have to confirm manually.

For doing a silent import, the "/f" (without the quotes) parameter has to be appended. This will look like this:

mwclient.exe "example.cfg" /f

In this case, the filename of the configuration has to be used with the quotes.

7.6 Version Comparison

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

7.7 Connect to Computer
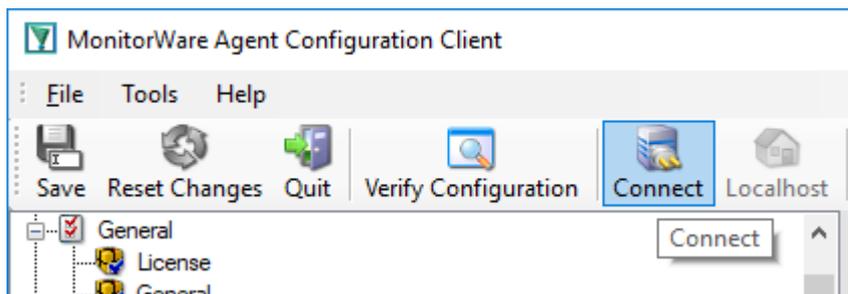
Click the Connect button in order to access another machine remotely. A window will open up.
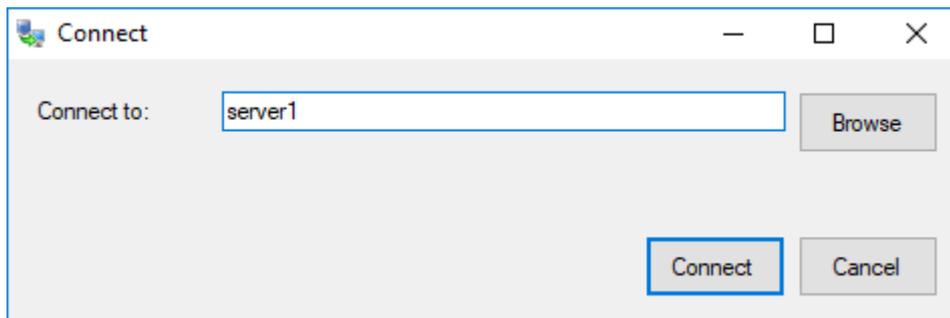
Here you can enter the name of the machine you want to configure remotely. You can either directly enter the name into the textfield or you use the Browse button to see a list of available machines in the network. The click on the Connect button, the configuration client will verify access to the remote machine. If the verification is successful, you will be able to proceed with the remote access. Otherwise an error message will be shown.

Please Note: For remote configurations, you must ensure, that the remote machine is accessible by network and has access rights for the current logged on local user.

8 Copyrights

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

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

Openssl-1.0.2k: http://www.adiscon.org/3rdparty/openssl-1.0.2k.tar.gz
Net-SNMP-5.7.3:   http://www.adiscon.org/3rdparty/net-snmp-5.7.3.tar.gz
Liblogging: http://www.adiscon.org/3rdparty/liblogging.zip
Librelp-1.2.11:   http://download.rsyslog.com/librelp/librelp-1.2.11.tar.gz
Libfastjson-0.99.8:   https://github.com/rsyslog/libfastjson/archive/v0.99.8.zip
Liblognorm-2.0.4:   https://github.com/rsyslog/liblognorm/archive/v2.0.4.zip

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

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

9 Glossary of Terms

The Glossary of Terms is also available on the Web:

http://www.monitorware.com/Common/en/glossary/

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

9.1 EventReporter

EventReporter is Adiscon's solution to forward Windows 2000/2003/2008/2012/2016/XP/Vista/7/8/10 event log entries to a central system. These central systems can be either WinSyslog's, other Syslog daemons (e.g. on UNIX) or MonitorWare Agents. EventReporter is part of Adiscon's MonitorWare line of products.

Click here for more Information about EventReporter.

9.2 Millisecond

A millisecond is a thousand of a second. It is abbreviated as "ms". As such, 500ms mean half a second.
Inside the MonitorWare line of products, many timers are expressed in milliseconds as a fine control over the services and actions is provided to the administrator.

9.3   Monitor Ware Line of Products

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

- AliveMon (www.alivemon.com)
- EventReporter (www.eventreporter.com)
- MonitorWare Agent (www.monitorware.com)
- WinSyslog (www.winsyslog.com)
- Rsyslog Windows Agent (www.rsyslog.com)

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

- Liblogging (www.liblogging.org)

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

9.4   Resource ID

The Resource ID is an identifier used by the MonitorWare line of products. It is a simple, administrator assigned string value. It can be used to correlate different events - even from different source - to a specific resource.
For example, on a Windows server running Microsoft Exchange, all Exchange events could be assigned to a resource id of "Exchange Server".

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

Later releases of the MonitorWare Line of Products will much broader support the Resource ID.

9.5   RELP

RELP is the "Reliable Event Logging Protocol". It assures that no message is lost in transit, not even when connections breaks and a peer becomes unavailable. The current version of the RELP protocol has a minimal window of opportunity for message duplication after a session has been broken due to network problems. In this case, a few messages may be duplicated (a problem that also exists with plain tcp syslog).

RELP addresses many shortcomings of the traditional plain tcp syslog protocol. For some insight into that, please have a look at http://blog.gerhards.net/2008/04/on-unreliability-of-plain-tcp-

syslog.html. Please note that RELP is currently a proprietary protocol. So the number of interoperable implementations is limited.

Note that for reliable operation where messages should be preserved over a service shutdown, queue cache mode must be activated.

9.6 SETP

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

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

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

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

9.7 SMTP

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

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

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

9.8 Syslog Facility

Syslog Facility is one information field associated with a syslog message. It is defined by the Syslog protocol. It is meant to provide a very rough clue from what part of a system the message originated from. Tradidionally, under UNIX, there are facilities like KERN (the OS kernel itself), LPD (the line printer daemon) and so on. There are also the LOCAL_0 to LOCAL_7 facilities, which were traditionally reserved for administrator and application use.

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

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

9.9 TCP

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

9.10 UDP

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

Click here for more Information about UDP.

9.11 Upgrade Insurance

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

Click here for more Information about Upgrade Insurance.

9.12 IPv6

Adiscon Products officially support IPv6. The IPv6 support was introduced with the following versions:

MonitorWare Agent 8.0
WinSyslog 11.0
EventReporter 12.0

Support for IPv6 is available in all network related facilities of the engine. All network related actions will automatically detect IPv6 and IPv4 target addresses if configured. You can also use DNS resolution to resolve valid IPv6 addresses. Network related Services can either use IPv4 or IPv6 as internet protocol. In order to support both protocols, you will need to create two services. The only exception is the RELP Listener, which uses IPv4 and IPv6 automatically if available.

9.13 UTC

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

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

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

Index

- I -

- M -

- S -

Endnotes 2... (after index)

Back Cover